



Customer Relationship  
Management

## **CRM.Web Administrator's Guide**

**Version 14.1**

# Notices

---

Copyright © 2004–2022. Aurea Software, Inc. (“Aurea”). All Rights Reserved. These materials and all Aurea products are copyrighted and all rights are reserved by Aurea.

This document is proprietary and confidential to Aurea and is available only under a valid non-disclosure agreement. No part of this document may be disclosed in any manner to a third party without the prior written consent of Aurea. The information in these materials is for informational purposes only and Aurea assumes no responsibility for any errors that may appear therein. Aurea reserves the right to revise this information and to make changes from time to time to the content hereof without obligation of Aurea to notify any person of such revisions or changes.

You are hereby placed on notice that the software, its related technology and services may be covered by one or more United States (“US”) and non-US patents. A listing that associates patented and patent-pending products included in the software, software updates, their related technology and services with one or more patent numbers is available for you and the general public’s access at <https://markings.ip-dynamics.ai/esw/> (the “Patent Notice”) without charge. The association of products-to-patent numbers at the Patent Notice may not be an exclusive listing of associations, and other unlisted patents or pending patents may also be associated with the products. Likewise, the patents or pending patents may also be associated with unlisted products. You agree to regularly review the products-to-patent number(s) association at the Patent Notice to check for updates.

Aurea and Aurea Software are registered trademarks of Aurea Software, Inc. in the United States and/or other countries. Additional Aurea trademarks, including registered trademarks, are available at: <https://www.aurea.com/legal/trademarks/>. Jive is a registered trademark of Jive Software, Inc. in the United States and/or other countries. Additional Jive trademarks, including registered trademarks, are available at: <https://www.jivesoftware.com/legal/>.

---

---

# Table of Contents

<b>Preface</b> .....	<b>9</b>
About this documentation.....	9
Notation conventions.....	9
Aurea global support.....	10
<b>Chapter 1: Overview</b> .....	<b>11</b>
CRM.Web Introduction.....	11
Support.....	11
CRM.Web Package.....	12
Customizing Aurea CRM.Web.....	13
General Information.....	14
Access Rights.....	14
CRM.Server.....	15
Customizing Styles.....	15
Field Names and IDs.....	17
Invalidate Cache.....	18
Info Area IDs.....	19
Link Fields and Reference Texts.....	19
Link IDs.....	19
Logging.....	22
Plug-ins.....	24
Syntax.....	25
User Settings.....	25
<b>Chapter 2: Configuring CRM</b> .....	<b>27</b>
Aurea CRM.Web Administration.....	27
Application Settings.....	28
System Configuration.....	29
SDK.....	30
Reps.....	31
Logins.....	31
Station.....	32
Catalog Maintenance.....	33
Mass Data Operations.....	39
Import.....	43
Jobs & Logs.....	47
Field Help.....	50
Texts.....	50

---

---

Language.....	52
Managing Checked Out Documents.....	52
Variables.....	53
CRM.Designer Maintenance.....	56
Data Model.....	57
CRM.Designer.....	60
Integration of CRM.Designer & Aurea CRM.Web.....	60
Using CRM.Designer.....	63
Logging Changes of CRM.Designer Items.....	69
Providing the Enterprise Serial Key to CRM.Designer.....	72
Users and Configurations.....	74
Configurations.....	74
Users   Configurations.....	78
Web Configuration.....	87
Setting Values of Web Configuration Parameters.....	88
Parameter Types.....	89
Customizing Web Configuration Parameters.....	90
Configuration Layouts.....	93
Web Configuration Parameters.....	93
Languages and Texts.....	172
Languages.....	173
Texts.....	176
Uploading and Downloading Texts.....	185
Info Area Customizations.....	186
Field Groups.....	188
Table Captions.....	265
Alternate Field Labels.....	270
Custom Field Labels and Inputs.....	271
Configuring Line Breaks and Truncating Texts.....	271
Custom Links.....	274
Link Names.....	274
Info Area Labels.....	275
Virtual Info Areas.....	275
Navigation Customizations.....	287
Defining an Action.....	287
Default Action.....	289
Menus.....	290
Buttons.....	304
Headers.....	321
Quick Search.....	346
<b>Chapter 3: Using CRM.....</b>	<b>350</b>
Action Templates.....	350
Expand.....	351

---

---

Search.....	364
Tree.....	381
JavaScript.....	391
ShowForm.....	393
AnalysisEdit.....	394
AnalysisRun.....	395
AnalysisSelect / QuerySelect.....	398
BuyingCenter.....	399
Calendar.....	402
CircleOfInfluence.....	403
CreateRelation.....	408
DayWeekPlanning.....	410
DefaultValue.....	415
DispatchingDashboard.....	416
ExecuteReport.....	423
HierarchicalCatalogTreeView.....	424
HierarchicalCircleOfInfluence.....	426
HierarchicalSerialEntry.....	427
MailMerge.....	430
MarketingCalendar.....	432
Message.....	434
NewSiblingRecord.....	434
QueryReport.....	436
QueryRun.....	441
RedirectPage.....	444
RelationTreeView.....	446
ResourcesView.....	448
SearchPhoneEmail.....	453
SerialEntry.....	454
Timeline.....	475
Trigger.....	475
URL.....	479
VisitPlanning.....	479
Workflow.....	485
Filter   Query   Analysis.....	487
Filters.....	488
Format Types for Queries and Analyses.....	492
Queries.....	493
Advanced Filter Criteria.....	504
Dynamic Filters.....	509
Analyses.....	513
Timelines.....	538
Key Performance Indicators (Aurea Enterprise).....	543
Chart Types.....	545
Processes.....	547

---

---

Process Elements.....	548
Designing a Process.....	572
Calling a Process.....	575
Cancelling a Process.....	575
Action Templates within Processes.....	576
Process Ask Configuration.....	582
Standard Process Steps.....	590
Examples.....	591
Forms.....	621
Form Designer.....	622
Defining a Header/Title for a Form.....	630
Displaying the Total Number of Result Records in a List View.....	631
Testing Your Forms.....	631
Displaying Forms in Aurea CRM.Web.....	632
Dashboards.....	632
LiveTile Widget.....	635
Showing/Hiding the Default Text of DateTime Widget.....	635
Enterprise Features.....	636
Aurea CRM Add-in for Microsoft Word.....	637
Cloud Backup Service.....	642
Change Notifications Service.....	654
Google Analytics Integration.....	658
Aurea Campaign Manager (formerly Lyris HQ) Integration.....	669
Launching Fat Clients Using Aurea CRM.....	687
Channels Output Cache Framework.....	691
KPI Definition (Enterprise CRM).....	701
Packages.....	711
Defining a Package.....	712
1-Click Package Management.....	714
Transferring Packages.....	715
Administration.....	717
Images.....	717
Catalogs.....	720
Roles & CRM Processes.....	727
Delete Units.....	738
Consistency Check.....	742
Configuration Validator.....	744
Compare Field Controls.....	746
Authentication.....	748
Settings.....	750
<b>Chapter 4: Other Programs and Functions.....</b>	<b>752</b>
Affiliation View.....	752
Configuration Units.....	754

---

---

Alternative Condition for Affiliation View.....	757
Campaign Graph.....	758
Creating/Updating Records From email Signatures/Websites.....	758
Configuration Units.....	759
CRM.bulkloader.....	760
Downloading Configuration Data from the CRM.Designer Database.....	761
Uploading Configuration Data to the CRM.Designer Database.....	764
Maintenance.....	768
Data Model, CRM Processes, Roles and Catalogs.....	772
Data Model References.....	779
Aurea CRM Data Model Upgrade Implications on Synchronized Catalogs.....	781
CRM.bulkloader Command Line Parameters.....	782
Bulkloader Database configuration settings.....	791
CRM.launcher(Enterprise).....	793
Installing CRM.launcher.....	793
Configuring the CRM.launcher.....	808
CRM.launcher Logging and Debugging Reports.....	816
Troubleshooting CRM.launcher issues.....	817
Communication with Aurea CRM.Web.....	820
CRM.launcher communication with CRM.phone plugin.....	822
CRM.launcher communication with Document plugin.....	822
CRM.launcher Languages.....	823
CRM.Phone.....	824
Displaying a Record's History.....	824
Displaying Bing Maps in Related Areas and Forms.....	826
Configuring a Map Panel.....	828
Documents.....	829
Access Rights.....	829
Configuring MIME Types.....	830
Uploading Documents.....	830
Document fields.....	831
Document Check-Out/-In.....	832
Context Menus and Menu Actions.....	833
emails.....	835
Send As email.....	835
Notify Rep.....	837
email Reply.....	839
External Address Check.....	840
Favorites.....	842
Letters & Mail Merge.....	843
Mail Merge Configuration.....	844
Quick Campaigns.....	845
Mail Merge Letters Based on Marketing Activities.....	849
File Name Variables.....	850
Document Caching & Performance Considerations.....	852

---

---

Merging Companies and Persons.....	852
Merging Properties (BTB only).....	854
Refreshing Data after C(R)UD Events.....	854
Support for Database Transactions.....	855
Tuning the Performance of the Notification Engine.....	856
XML Exports & Reporting.....	856
XML Output.....	857
Export Options.....	858
XSL Transformations.....	862
Creating Aurea CRM win Reports in Aurea CRM.Web.....	863
Customer Admin Mode.....	865
Creating SelfAdmin Configurations.....	868
Defining and Assigning a Customer Admin Role.....	870
EditDetailsControl.....	871
Base Customizing.....	873
Extending Field/Header Groups Defined in Base.....	875
Virtual Info Areas & Core Customizing.....	876

**Appendix A: Data Types in Aurea CRM.....877**

**Appendix B: Effectiveness of Rights and Configuration Changes...878**

**Appendix C: AQL – Aurea Query Language.....881**

AQL Syntax.....	882
Example.....	884
Filtering by Catalog Values.....	885

**Appendix D: URL Parameters.....887**

---



# Preface

---

For details, see the following topics:

- [About this documentation](#)
- [Notation conventions](#)
- [Aurea global support](#)

## About this documentation

This guide is part of the documentation set for Aurea CRM.

## Notation conventions

This document uses the following notation conventions:

Convention	Meaning
Fixed-width	Fixed-width font indicates code, path names, file names, environment variable names, parameter names, command names, machine names, URLs.
<b>Bold Fixed-width</b>	<b>Bold Fixed-width</b> font is used to indicate user input or to emphasize certain lines of code.
<i>Italic Fixed-width</i>	<i>Italic Fixed-width</i> font indicates a placeholder for which you must supply a value.
<b>Bold Sans serif</b>	<b>Bold sans serif</b> typeface indicates the names of graphic user interface elements such as dialog boxes, buttons, and fields.
<i>Italic serif</i>	In text, <i>italic serif</i> typeface indicates the first use of an important term. The term is defined in the glossary.
Underlined	Underlined text in command lines and parameter descriptions indicate that you only have to enter the underlined part of the command or parameter name. For example, if you use the <u>-LOGFILE</u> parameter in a command, you only need to enter <code>-LOGF</code> .
[ ]	Brackets enclose optional arguments.
{ a   b   c }	Braces enclose two or more items. You can specify only one of the enclosed items. Vertical bars represent OR separators. For example, you can specify a or b or c.

---

Convention	Meaning
...	Three consecutive periods indicate that you can repeat the immediately previous item. In code examples, they can be horizontal or vertical to indicate omissions.
<b>Menu &gt; Choice</b>	An angle bracket between two menu items indicates that you should choose an item from a menu. For example, the notation <b>File &gt; &gt; Exit</b> means: "Open the <b>File</b> menu and choose <b>Exit</b> ."
>>	Links to related information in other chapters or documents are indicated using the >> symbol.

## Aurea global support

If you encounter a problem while using an Aurea product or require assistance with downloading the software or upgrading a product release, please open a ticket on [Aurea Support Central](#). Preferably, search the articles on the [Aurea Knowledge Base](#) for solutions to your issues before opening a ticket.

Information about the support organization is available on Support Central. The product documentation is available at <https://help.aurea.com/crm/#>.

For information about purchasing an upgrade or professional services, contact your account executive. If you do not know who your account executive is, or for other queries, [contact us](#) through our [website](#).

# 1

## Overview

---

Customer Relationship Management provides you with a comprehensive CRM experience from the initial contact with the client to offering optimal support and targeted services.

### CRM.Web Introduction

Aurea Customer Relationship Management is a CRM solution with end-to-end capabilities for sales and marketing.

Aurea CRM creates a one-stop-shop that delivers a compact overview of team activities, sales pipelines, and key performance indicators so you can set priorities and make informed decisions.

CRM.Web is the administration platform that allows users to design, manage, and control the CRM solution from a single interface.

Our business solutions cater to the demands of businesses of all sizes, providing services to both small departments and multinational corporations. All marketing, sales, and service processes are depicted in the system, helping you provide your clients with a comprehensive CRM experience from the initial contact with a client to offering optimal support and targeted services.

Aurea CRM can be deployed without any programming knowledge and configured to meet your internal processes.

For more information about Aurea CRM, see the [product documentation](#) in Aurea Support Central. While planning for an installation or upgrade of Aurea CRM, refer to the system requirements.

For any upgrade or purchase inquiries, please consult your account executive.

### Support

Contact the Aurea Support Center by opening a call on our web site if you encounter any problems with Aurea CRM products that you cannot solve or want to report a potential bug.

Feature requests and consulting services are not part of the services provided by the Support Center; ideas for new features can be submitted at <http://support.aurea.com/>, consulting requests is forwarded to the respective department for processing, and is invoiced accordingly.

## Support Center

email: [email:support@aurea.com](mailto:email:support@aurea.com)

Web site: <https://support.aurea.com>

When opening a call, ensure to include the following information:

- A precise description of the problem (including screenshots, log files or other information that helps understand and reproduce the problem).
- Information on the version of Aurea CRM you are using and any patches you have applied, as well as information on any add-ons.
- Information on your Aurea CRM version can be found in the About Box or in the System information (**Patch Info** link). This information is also stored in the respective installation directories in the file `<product>_Patch_History.txt`.

A more detailed description of what information should be provided when opening a call can be found directly in the Call Management area of the web site.

## CRM.Web Package

The CRM suite brings you add-ons and other extensions to enhance its capabilities. In this topic, there is a list of all available components.

Your Aurea CRM.Web package contains:

- Aurea CRM.Web
- SQL scripts to create or upgrade a CRM.Designer database
- CRM.Designer
- The out-of-the-box configuration of Aurea CRM.Web consisting of XML files that contain the default designer configuration ("UPDATE\_DEFAULT") and CRM.Bulkloader used to upload the contents of these files to the CRM.Designer database.

This manual covers the following:

- Administrator functions in Aurea CRM.Web
- Description of CRM.Designer functionality and usage
- Customization of Aurea CRM.Web using CRM.Designer

Additional information is provided in the following manuals:

- For details on installing Aurea CRM.Web and CRM.Designer, see [Aurea CRM.Web Installation Guide](#).
- For information on the user interface and underlying functionality of Aurea CRM.Web, see [Aurea CRM.Interface Administrator Guide](#).
- For details on the Aurea CRM business logic, see [Aurea CRM.Web Business Logic User Guide](#) for your vertical.

- For information on CRM.Phone, see [CRM.Phone Administrator Guide](#).
- The administration and configuration of core functions are described in the [CRM.Core Administrator Guide](#).
- For details on the meaning and functionality of specific database fields, refer to the field help.
- For information about client and server-side programming concepts.

Information on the structure of the Aurea CRM database (tables, fields, indices, etc.) can be found in the **Data Model** (in the Aurea CRM.Win **Maintenance** module). You can also output the database structure as text or XML files. For further information see **mmba.exe Parameters** in the [CRM.Core Administrator Guide](#).

All available documentation can be found on the Aurea CRM support page at <https://support.aurea.com>

If not mentioned otherwise, all examples in this manual apply to the BTB vertical.

## Customizing Aurea CRM.Web

Aurea CRM.Web allows administrators to customize and extend the functionalities of CRM.Web to adapt to the differences in uses and processes of your particular organization.

There are different ways to implement customization:

- Customize the application using CRM.Designer. For further information, see [CRM.Designer](#) on page 60.
- Extend the client functionality by adding JavaScript code.
- Extend the server-side framework of Aurea CRM.Web.
- Customize Aurea CRM's core functionality by editing or extending the Aurea CRM data model, defining access rights and defining triggers and background workflows. For further information, consult the [CRM.Core Administrator Guide](#).

Using triggers and workflows has the advantage that less add-on programming is required and that triggers and workflows are automatically migrated from one version to the next.

The order of these methods reflects the complexity of using them. While it is rather simple to use CRM.Designer to customize the application, there may be advanced features that cannot be implemented with this tool. To implement such features you may have to use JavaScript code.

For information on more advanced customizations, e.g., adding .ASPX pages with server-side code, extending the underlying data model by adding complex virtual links or custom fields, or transferring data from the server to the client in a way that is not supported out-of-the-box, see the [Aurea CRM.Web Developer Guide](#). In addition, you can extend the server's business logic via add-on programming in C# or any other .NET programming language.

## General Information

The following chapters contain information that you should keep in mind while working with Aurea CRM.Web and CRM.Designer.

## Access Rights

Use access rights to control permissions for users.

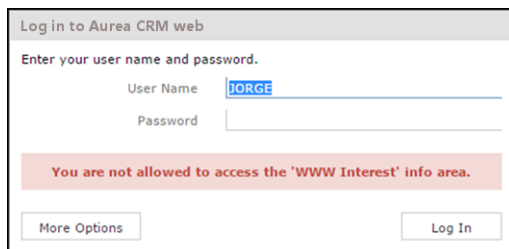
CRM.Web offers administrator a detailed control over which parameters and configurations user have the right to modify or not. Access right help to define different permissions to different groups inside your organization such as managers, administrator, supervisors, sales representatives, etc. The same permissions can be given to external business partners to whom you may grant access to CRM.Web.

Changes to access rights defined in the Aurea CRM.Win **Rights** module are only applied after refreshing the application using the **Settings > Invalidate Cache** . For further information, consult [Effectiveness of Rights and Configuration Changes](#) on page 878.

Access rights defined in the Aurea CRM.Win **Rights** module do not apply to CRM.Designer.

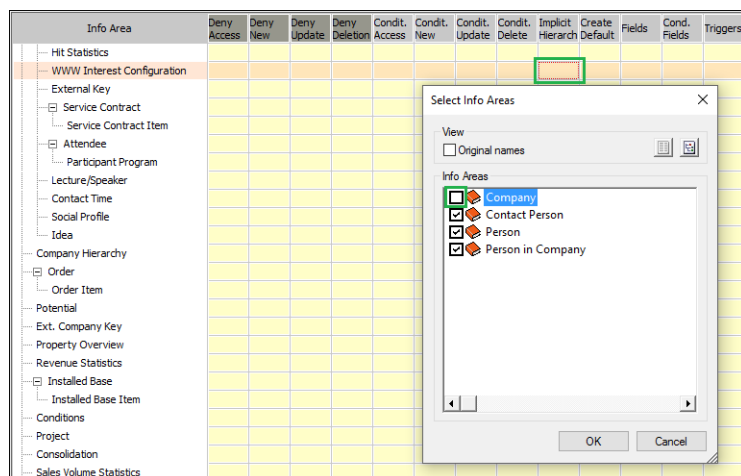
The I9 info area is used to save the user's roaming settings and is required by Aurea CRM.Web. Do not lock I9 or set it to read-only. Denying access to the Company FI info area also locks the I9 info area. For further information, see [User Settings](#) on page 25 section.

When the I9 info area is locked, a user cannot login into Aurea CRM.Web as illustrated in the screenshot below:



The screenshot shows a login form titled "Log in to Aurea CRM web". It includes a prompt "Enter your user name and password." and two input fields: "User Name" with the value "JORGE" and "Password". Below the fields is a red error message: "You are not allowed to access the 'WWW Interest' info area." At the bottom, there are two buttons: "More Options" and "Log In".

To provide login access to CRM.Web for a user, while denying access to the FI in-foarea, you should detach **Company** info area from the Implicit Rights settings of **WWW Interest Configuration (II)** and **WWW Interest (I9)** info areas. See the screenshot below.



For more information on access rights, see the chapter Defining Access Rights in the [CRM.Core Administrator Guide](#).

## CRM.Server

Aurea CRM.Web uses CRM.Server for background operations like calculations, executing marketing activities, mass data operations, stock-taking, and for asynchronously processing background queries.

For CRM.Server to work for Aurea CRM.Web, you need to specify the rep ID of the CRM.Server user in the `UpdateServer.RepID` Web Configuration parameter. For further information see [UpdateServer.RepID](#). CRM.Web then creates To-dos tasks for background processing for this rep.

For details on setting up CRM.Server, see the **CRM.Server** section in the [CRM.Core Administrator Guide](#).

## Customizing Styles

You can customize the look of Aurea CRM.Web such as color scheme, fonts etc. by creating new CSS files stylesheet.

The out-of-the-box style is called `update` and consists of the following files:

`..\styles\update.theme\*.css` containing constants like color scheme, fonts etc.

`..\styles\update\*.css`

`..\images\update\*.png`

The CSS files for pages and widgets are imported into the main CSS file using the `@import` rule:

```
@import url("../update/update.css ");
```

**Note:** Do not modify the CSS files delivered by Aurea software since these files may be replaced by newer versions when installing e.g. a hotfix.

The CSS files are automatically compiled and compressed whenever a CSS file changes. You can force the update by deleting the contents of the folder `..\styles\compiled`.

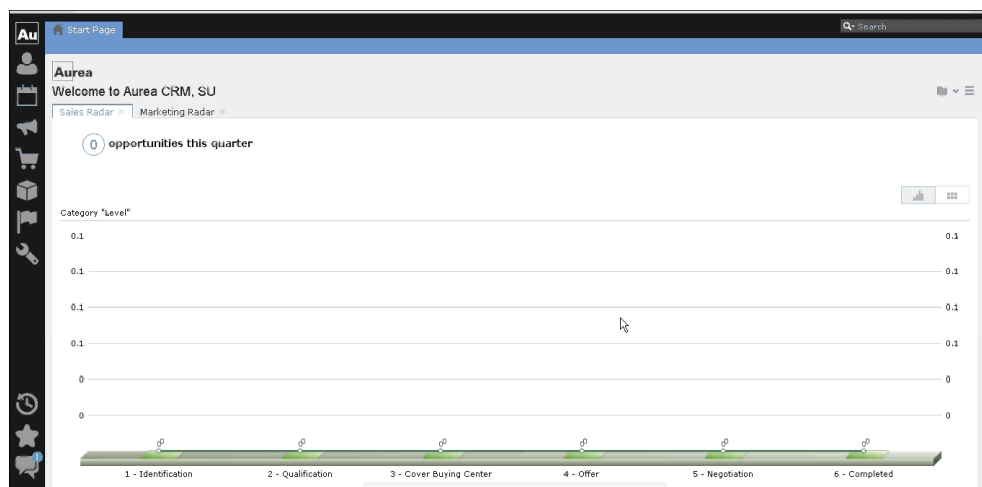
To apply your custom style to Aurea CRM.Web, set the `style` Web Configuration parameter. For further information see [Style](#).

You can set the color of the CRM.Web page header by setting the CSS property `u8-color-brand` to the required value. If you are using custom CSS setup, you can use this property in your CSS file to override the default page header color. For example, to get a blue color in the page header, you can set the following values:

```
!u8-color-accent = #6C97CE;
```

```
!u8-color-brand = !u8-color-accent
```

The resulting page header would look like this:



For details on creating custom styles and themes, CSS sprites, and how to change the color coding of calendar items contact the Aurea Support Portal at <https://support.aurea.com>.

## Aurea CRM.Web on Tablet Devices

An additional derived theme `update.touchable.theme` allows Aurea CRM.Web to be used on tablets devices. It contains larger user interface elements, more spacing, etc.

If you already use your custom theme, use the following naming convention to create a custom touchable theme from it. Create a derived theme in the folder `..\styles` using the naming convention `<standard theme>.touchable.theme`. For example, `mycustom.touchable.theme`.



The touchable theme does not necessarily need to contain a complete set of CSS files. Instead, create a new `constants.css` containing different constants for margins and spacing, etc. Refer to `..\styles\update.touchable.theme` for examples.

The touchable theme is automatically used on tablet devices, i.e., you do not need to set it via the `Style Web Configuration` parameter. If no `<standard theme>.touchable` style is available, `<standard theme>` is used as a fallback.

For devices supporting both mouse and touch gestures, users can choose which theme to use via the dynamic menu item `D_ToggleTouchableTheme` (**Use touch layout**) in `$System`. If no touchable style is found, `D_ToggleTouchableTheme` is not displayed. The user's selection is stored in the local settings. For further information see [User Settings](#) on page 25.

Alternatively you can set the theme via URL parameter:

```
http://<my web address>/crm/start?theme=update.touchable
```

## Field Names and IDs

Set Fields Names and IDs to identify fields in the CRM database.

The fields in the Aurea CRM database are identified by the following numbers/IDs:

	XML Output	Description	Usage
Field number	<code>fnr</code>	Sequential field numbering in the data model	None
Field ID	<code>fid</code>	<b>vertical-specific unique field ID</b>	Used to reference fields in CRM.Designer: All Aurea CRM.Web and CRM.Designer widgets containing a field selection display the vertical-specific unique field IDs.
Base field ID	<code>cid</code>	Base field ID	Used for template development (base template)

When specifying fields, as in a parameter definition, you can either enter the **unique field ID** or the **XML name**. The XML name is **vertical-independent**.

---

**Note:** Aurea recommends using XML names.

---

Wherever JSON or JavaScript is used, you need to enclose strings in single or double quotes, e.g., `{property1: "value"}`. In JSON, property names need to be enclosed in quotes as well, e.g. `{"property1": "value"}`. Therefore when referencing XML names in JavaScript expressions, you should always enclose them in double quotes.

To output field IDs and XML names:

- In CRM.Designer the Field Control Editor displays both XML name and ID for the selected field. For additional information, see [Field Attributes](#) on page 221.
- Start `mmba.exe` with the parameters `--writedbstr --xmlfieldnames`. For further information, check the **mmba.exe parameters** in the [CRM.Core Administrator Guide](#).

The `mm_dbstr.txt` file in the `..\system\db` folder lists all info areas and their fields. For example:

```
2 ; Last Name; FName; character; 80; ; ; ; ; LastName
```

- The XML name of the **Last name** field in KP is "LastName".
  - The field ID of this field is "2".
- Alternatively, look up this information in the data model. Open Aurea CRM.Win and go to the **Maintenance** module). To display XML names, select the desired info area and click in the **Text** column. Select **Display resource texts** from the context menu on the **Texts** page.

---

**Note:** When adding fields or creating new info areas, make sure you enter XML names for your fields. For further information, see the **Data Model** in the [CRM.Core Administrator Guide](#).

---

## Invalidate Cache

Invalidating cache ensures that all configuration units cached on the server are removed. The next time a client requests a configuration unit, the most recent version is available.

Modifications of configuration units are not immediately visible to web clients. For further information see [Integration of CRM.Designer & Aurea CRM.Web](#) on page 60.

To ensure that all clients get the current content:

1. Go to the the **Settings** menu (`ID_NET_SERVICES`).
2. Select the menu item **Invalidate Cache** (`ID_NET_INVALIDATECACHE`).

---

**Note:** Since action parameters are saved in the URL, you need to re-execute the action after invalidating the cache to make sure all your changes are applied.

---

**Note:** Because **Invalidate Cache** is a very expensive operation on the server, access to it has been restricted via the CRM Process Configuration (in the Aurea CRM win **Rights** module): `ID_NET_INVALIDATECACHE` is available for the SU and users who have access to the CRM process 1.5.1.101. "Web Administration" (CRM process ID 50050040). This CRM process is disabled by default.

## Info Area IDs

Info areas are identified by their 2 or 4 letter ID.

To identify info areas:

- For info areas defined in the Aurea CRM data model e.g. FI, KP, C001 etc. see **Data Model** in the [CRM.Core Administrator Guide](#).
- VXXX for virtual info areas, see the [Virtual Info Areas](#) on page 275.

## Link Fields and Reference Texts

Use the configuration to define custom reference texts.

Wherever a record contains a link to another record, the link field contains the record's reference text. You can define custom reference texts in the **Configuration** info area MC **Data Model** category, **Reference Text** option. For additional information, see **Configuration Info Area** in the [CRM.Core Administrator Guide](#).

## Link IDs

Create relationships between different Info Area with multiple links.

In Aurea CRM's generic data model, it is possible to define multiple links between two info areas. For Example, you can link two different persons to one project, as the project leader and the project assistant.

Numeric IDs identify links. The first link to an info area always has link ID zero: this is the default link. Additional links to the same info area (DUPLICATE), have ID 1, ID 2, etc.

ID	Info Area	Index (Ticket)	Index	
0	1 - N Activity (MA)	(2) KM-StaNo,KM-SerNo	(11) KM_StaNo,KM_SerNo	
1	1 - N Activity (MA)	(2) '1',KM-StaNo,KM-SerNo	(5) Activity Switch,PR_PR-	
2	N - 1 Company (FI)	(1) FI_StaNo,FI_SerNo	(1) FI-StaNo,FI-SerNo	
3	1 N - 1 Company (FI)	(6) FI_1_StaNo,FI_1_SerNo	(1) FI-StaNo,FI-SerNo	DUPLICATE
4	2 N - 1 Company (FI)	(8) FI_2_StaNo,FI_2_SerNo	(1) FI-StaNo,FI-SerNo	DUPLICATE
5	3 N - 1 Company (FI)	(9) FI_3_StaNo,FI_3_SerNo	(1) FI-StaNo,FI-SerNo	DUPLICATE
6	N - 1 Contact Person (CP)	(1) FI_StaNo,FI_SerNo,PE	(1) FI_StaNo,FI_SerNo,PE	
7	N - 1 Item Master (AR)	(0) Item No.	(1) Item No.	
8	1 - N Offer (PR)	(2) KM-StaNo,KM-SerNo	(9) KM_StaNo,KM_SerNo	UPD
9	1 - N Order (AU)	(2) KM-StaNo,KM-SerNo	(7) KM_StaNo,KM_SerNo	UPD

You can find the number ranges for link IDs based on their type and context. See Link IDs in the [CRM.Core Administrator Guide](#).

To identify the correct link when e.g. creating a query in CRM.Designer, you need to specify the link ID. For further information, see [Specifying Link IDs](#) on page 20.

To prevent users from selecting the wrong link (e.g. in the **Modify List Fields** dialog), you can define meaningful **Link Names** to be displayed in Aurea CRM.Web instead of the numeric link IDs. For additional information, see [Link Names](#) on page 274.

## Specifying Link IDs

CRM.Web allows you to specify Link IDs using CRM.Designer.

You may need to specify the link ID in CRM.Designer when configuring the following units:

- Expand configurations (**Parent Relations** and **Breadcrumb Parent**). See [Expand Configuration](#) on page 352
- Field groups (**Add Linked Fields** functionality). See [Add Linked Fields](#) on page 216
- `SearchSubList` and `ExpandChildren` headers (when configuring the sub-info areas). See [Related Data area \(Header Info Areas\)](#) on page 329

---

**Note:** If you configure sub-lists for an info area that is linked to the parent via a generic link. For example, the Document Links (D3), To-Do (TD) or Messages (MD) info areas, you always have to specify the link ID 127.

---

- Query definition (adding info areas). See [Defining Queries](#) on page 496
- Table captions (when using linked fields). See [Configuring a Table Caption](#) on page 266
- Timelines. See [Configuring a Timeline](#) on page 540
- TreeView configurations (parent link selection). See [TreeView Configuration](#) on page 381
- Linked Records in a Process Ask configuration. See [Process Ask Configuration](#) on page 582. (In this case you need to enter the link ID using `LINK`, e.g., `Link[KP:1]`. See [Link Names in JavaScript](#) on page 22.)

---



**Note:** You only need to specify the link ID if you want to use another than the default link.

---

To specify a link ID:

1. Check **LinkID**.
2. Select the desired link from the drop-down list.

If both the source and the destination info area are known, only the allowed Link-IDs between these two info areas are displayed in a drop-down list.

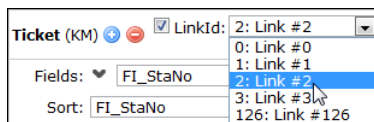
Most pages offer a toggle option to select a link ID or manually enter it: Either **Text/Selection** links or  (text input)/ (catalog input) icons (see examples below).

**Note:** The text and selection fields are not bound to each other. Therefore, make sure you clear the value/selection before switching to the other input method to avoid saving an undesired link ID.

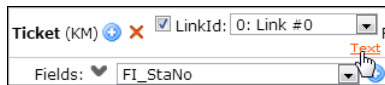
**Note:** If no context is available when defining a configuration unit (e.g., a timeline), you need to manually enter the link ID. See [Link IDs](#) on page 19.

### Examples

- Adding an info area to a query using another than the default link:




Click on the **Text** link to enter a link ID manually:





- Adding a linked field to the table caption  $KM$ :


Table Caption ' $KM$ ' \*

Save Delete







Image Kundenproblem 

Prefix-Text Ticket 

Format-Text {1} - {2} 



Description Enter description here. 


Configuration User\_SUSANNA\_ChildOf\_UPDATE\_DEFAULT

Info Area	Link ID	Field
1. Ticket ( $KM$ ) 		No. (1088) 
2. Ticket ( $KM$ ) 		Subject (1090) 
3. Company (FI) 		(Created on)internal code (4202) 
+ Company (FI)		

Special definitions

When these fields are blank

	No. ( $KM$ :1088)		
			Use this format instead
			{2}

Click  to enter a link ID manually:



## Link Names in JavaScript

When using Javascript, you need specific parameters to identify link relations.

To specify link relations in JavaScript, use the following parameters:

- `$Parent`: Reads the parent record.
- `$Children`: Reads all child records. Specify `$Children[<infoAreaID>` to determine from which info area the child records are read, e.g., `$Children[Y2]` gets the opportunity participants linked to an opportunity.
- `$Link`: Specifies the link to be used, e.g., `$Link[KP:1]` reads all person records linked via the link with ID 1.
- `$RelatedCompany/$RelatedPerson/$RelatedHuman`: Reads the target FI/KP/PE record for a relationship record. Example in `UPDATE_DEFAULT`: The menu action `A_GotoRelatedPerson` in `M_Relationship_TreeView` is used to access a related KP record from a relationship record.
- `$GenericLink`
- `$CustomLink`
- `$CurrentRep`

## Logging

Log files are an essential source of information when troubleshooting issues or auditing the system for specific functionalities.

By default, the server logs are written to the following files:

- CRM.Designer: `..\web.data\log\designer.log`
- Aurea CRM.Web: `..\web.data\log\web.log`

For details on logging on the server, see **Server Log** in the [Aurea CRM.Web Installation Guide](#).

## Client Log

Use the client log to troubleshoot issues with CRM.Web and to create bug reports.

Start Aurea CRM.Web or CRM.Designer with the URL parameter `?debug` to display the client log.

Use CTRL+F to search the log.

## Filtering the Client Log

You can filter entries in the client log by selecting **Log > Filter**.

Click **Apply Filter** to apply your filter. Click **Clear Filter** to reset.

### Syntax:

+<facility name> OR -<facility name> (separate multiple expressions by comma)

You can identify the available facilities via the tooltips that are displayed on mouse-over the time entry at the beginning of each line.

### Examples:

```
+process
```

The client log displays only process-related entries.

```
-process
```

The client log displays everything except process-related entries.

```
+process, +notifications
```

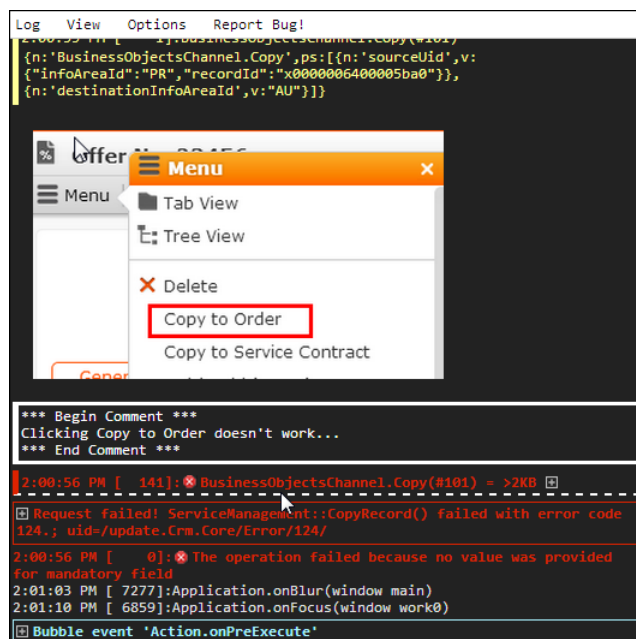
The client log displays process-related entries and notifications.

```
<300
```

Defines the log level. Only errors and warnings are visible in the client log. Use the following numeric values for log levels: error = 100, warning = 200, notice = 250, informational = 300.

## Adding Screenshots and Comments

You can insert screenshots and comments into the client log. Click on the element for which you want to insert a screenshot or comment. A horizontal dashed line is displayed.



To insert screenshots:

1. Create screenshots and save them. Supported formats: PNG, JPG, BMP, etc. - all formats supported by your browser.
2. Drag one or more screenshots onto the client log window.

Alternatively, when using Google Chrome, you can paste an image from the clipboard using Ctrl+V.

To add a comment:

1. Click **View > Comments**.
2. Enter the comment in the **Comment** field.
3. Click **Add Comment**.

The screenshot/comment is inserted at the dashed line. Comments are marked `*** Begin/End Comment ***`. If you did not select any element in the log, the screenshot/comment is inserted at the end of the client log.

To search for comments:

1. Enter `+comment` as filter expression.

To remove screenshots and comments:

1. Click the **✗ (Delete)** icon displayed for the element on mouse-over.

## Creating a Bug Report

To include information from the client log when submitting **support calls**:

1. Click **Report Bug!** to create a bug report containing the log (with your screenshots and comments) as a download.

Attach this file to your support call.

## Plug-ins

CRM.Web uses plug-ins to communicate with third-party applications.

Aurea CRM.Web needs to communicate with applications *outside the browser sandbox* for some of its functions. For example, MS Office and CRM.Phone. Use CRM.Launcher and/or plug-ins to provide functionalities specific to the external application. Some functionalities are listed below:

- Document check-in/-out, editing and mail merge functions for MS Word.
- Using CRM.Phone with Aurea CRM.Web.

---

**Note:** CRM.Launcher handles communications with external applications. If your browser's version is from before 2014, using the NPAPI (Netscape Plugin APIs), you should switch to the latest version of the browser. Aurea CRM no longer supports NPAPI.

More information on the currently used versions of the plug-ins and/or CRM.launcher can be found on the **System Information** page on Aurea CRM.Web. Go to **Settings > System Information** .

---



---

## Installation on the client

Users are offered a download link when they first access a function that requires a plug-in or CRM.Launcher. Clicking on the link downloads the appropriate file from the server and automatically starts the installation.

On the server the files are located at:

```
..\data\plugins\update.web.client.msi  
..\data\plugins\CRM.launcher.exe
```

---

**Note:** The CRM.launcher setup requires .NET framework 4.5. If .NET framework 4.5 is already installed on the client, setup can be executed within the user context. Otherwise, the setup installs .NET framework 4.5. This installation requires administrator rights.

---

## Syntax

This topic contains general information on using JavaScript in CRM.Designer

Wherever JSON or JavaScript is used, you need to put strings in (single or double) quotes, e.g., {property1: "value"}. In JSON property names need to be put into quotes as well, e.g., {"property1": "value"}.

Note that property names specified in JavaScript and JSON (e.g. in a record selector) are **case-sensitive** (searchLinkId, infoAreaId, uid).

Aurea CRM.Web contains built-in functions to convert native JavaScript values into strings or vice-versa, enabling you to deal with different field types:

- `u8.services.catalogs`, see the article “How to work with catalogs” at <https://support.aurea.com>
- `u8.services.typeConverter`, see the article “The Type Converter Service” at <https://support.aurea.com>

For further information on using JavaScript in CRM.Designer, contact Aurea Support Portal at <https://support.aurea.com>

## User Settings

User settings are saved between sessions according to their types.

User settings are automatically saved between sessions:

- **Roaming settings** are saved in the Aurea CRM database (in the I9 info area). These settings are used on all clients whenever a user re-accesses the respective page, e.g.. search settings, sort order of a list, filter sets, etc.
- **Local settings** are saved in the browser's local storage on the client. These settings are client-specific, e.g., the size and position of a popup, the view settings of a dashboard panel, etc.

Additional information for developers is available in the article “How to Deal with User Settings” at <https://support.aurea.com>.

# 2

## Configuring CRM

---

Use CRM.Designer to provide custom configuration to meet and facilitate your internal processes.

### Aurea CRM.Web Administration

Use the setting and administration tab to configure the CRM console.

CRM.Web gives administrators excellent control over the settings and administration of the CRM console. Admins can manage the feature available and configure them to facilitate the day-to-day administrative tasks.

Use the **Settings**  menu to access Aurea CRM.Web's administrative functions.

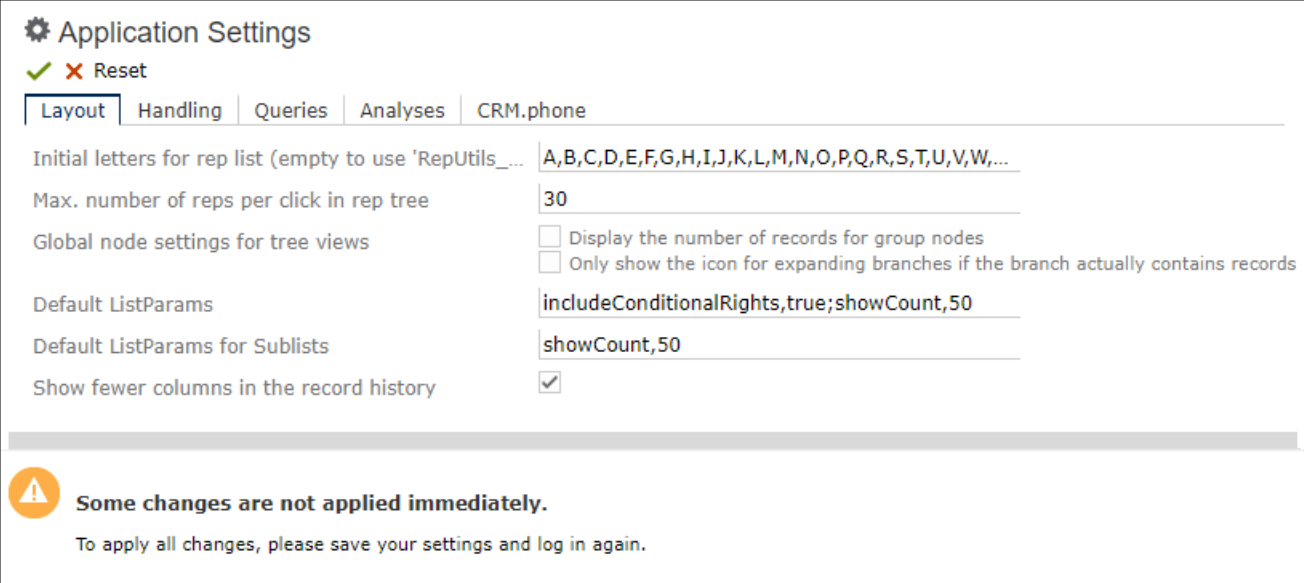
The menu `ID_NET_WEBDESIGNER` is available for the SU (Super User) and users who have access to the CRM process Web Administration `CRM process ID 50050040`. This CRM process is disabled by default.

The following chapters describe the contents and functionality of the **Settings** menu as configured in `UPDATE_DEFAULT`.

For information on **Portfolio**, **Portfolio Configuration**, and **Exchange Rate**, see the [Web Business Logic User Guide](#).

## Application Settings

The Application Settings section controls layout, handling of documents, query parameters, and analyses.



Application Settings

✓ ✗ Reset

Layout Handling Queries Analyses CRM.phone

Initial letters for rep list (empty to use 'RepUtils\_...': A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,...

Max. number of reps per click in rep tree: 30

Global node settings for tree views:

- Display the number of records for group nodes
- Only show the icon for expanding branches if the branch actually contains records

Default ListParams: includeConditionalRights,true;showCount,50

Default ListParams for Sublists: showCount,50

Show fewer columns in the record history:

**Some changes are not applied immediately.**  
To apply all changes, please save your settings and log in again.

To configure the appearance of Aurea CRM.Web and the default behavior of specific actions, select **Settings > Application Settings**.

### Note:

- The **Application Settings** menu item `ID_NET_APPCONFIGURATION` is available for the SU (Super User) and users who have access to the `CRM process 1.5.1.4.19 Application Configuration CRM process ID 50050050`. This CRM process is disabled by default.
- As shown in the screenshot above, some settings only take effect upon restarting Aurea CRM.Web.

For more details on these settings, see [Web Configuration Parameters](#) on page 93. In CRM.Designer you can define which tabs and settings are displayed as **Application Settings**. For further information, see [Configuration Layouts](#) on page 93.

## System Configuration

The System Configuration section holds the fields to configure the parameters for log files, documents, emails, system settings, offline sync, and Data Model.

To configure the system setting such logging, email behavior, etc. select **Settings > System Configuration**.

### Note:

- The **System Settings** menu item `ID_NET_SYSTEMCONFIGURATION` is available for the SU (Super User) and users who have access to the CRM process 1.5.1.4.21 System Configuration CRM process ID 50050080. This CRM process is disabled by default.
- Some settings only take effect upon restarting Aurea CRM.Web.
- Be careful when changing these parameters. If configured in a wrong way, parts of Aurea CRM.Web might not work correctly. Therefore, only the administrator SU should change these parameters during customization.

For more details on these settings, see [Web Configuration Parameters](#) on page 93.

In CRM.Designer you can define which tabs and settings are displayed as **System Configuration** section. For further details, see [Configuration Layouts](#) on page 93.

# SDK

Use the SDK page to explore and test the functionality of the available widgets in CRM.Web

Aurea CRM web SDK - Samples					
<b>A</b>	Actions	Administration	Analysis	Associations	Authentication
<b>B</b>	Benchmark	Binding	BreadCrumbs	BulkOperations	Busy
	Buttons	BuyingCenter			
<b>C</b>	Calendar	CampaignGraph	CatalogMaintenance	Catalogs	Channels
	Charts	CheckBox	CircleOfInfluence	Codeeditor	Collaboration
	ColorPicker_V1	ColorPicker_V2	Colors	ComboBox	ConfigurationGraph
	ConflictGrid	ContactTimes	Coverage-report.utml	CrmProcessesAndRoles	Crud
	Css				
<b>D</b>	DashboardPanel	DataModelEditor	DataQuality	DatePicker	DayWeekPlanning
	DefaultValues	Designer	DispatchingDashboard	Documents	DragAndDrop
<b>E</b>	Email	ExpandAllFields	ExpandView		
<b>F</b>	Favorites	FilterForm	Filtersets	FlexPanel	Form
	Formats	FormDesigner	FormFrame		
<b>G</b>	Geolocation	Gestures	Graph	Grid	GroupPanel
<b>H</b>	Header	Help	History		

The **SDK** section contains samples of all widgets available in Aurea CRM.Web. Use the SDK page to explore and test the functionality of these widgets, without repeatedly having to access CRM.Designer or re-write code.

To access the SDK sections go to **Settings > Maintenance > Reps** .

To reset the examples provided on the SDK page, you need to delete the browser history.

## Reps

The Rep info area allows administrators to emulate their internal company structure, such as employees, groups, and resources.

The screenshot shows the 'Search Reps' interface. At the top, there is a search bar and a 'Search' button. Below this is a search filter section with the following fields:

- Name: [Name, Last Name, First Name] Type: [Add a value...]
- E-mail: [Internal] [?]
- Company: [ID-TenNo] [With me] [Add more ...]

Below the filter section, there is a 'Start Search' button and a 'Max. 15 records per page' dropdown. The search results are displayed in a table with the following columns: Type, Name, First Name, Last Name, Org. Group, Customer/..., ID-TenNo, and Rep/Group. The results list several employee records:

Type	Name	First Name	Last Name	Org. Group	Customer/...	ID-TenNo	Rep/Group
Employee	ADAM					9101	<a href="#">ADAM</a>
Employee	Andi test					9101	<a href="#">Andi test</a>
Employee	Andy Admin	Andy	Admin			9101	<a href="#">Andy Admin</a>
Employee	Angelika Ri...	Angelika	Ring			9101	<a href="#">Angelika Ri ...</a>
Employee	Anna Anna...	Anna	Annalena			9101	<a href="#">Anna Anna ...</a>
Employee	Annabelle ...	Annabelle	Berger				<a href="#">Annabelle ...</a>

Select **Settings > Maintenance > Reps** to access the **Rep** info area ID.

Administrators can use the **Rep** info area to:

- Create new rep records.
- Search for an existing rep.
- Add/change rep data like telephone profile, contact information, etc.
- Access and configure a rep's login. For further details, see [Logins](#) on page 31.
- Assign roles to reps. For further information, see [Role-Based Access](#).
- Define a rep's special skills, e.g., fields of specialization, knowledge of products etc.
- Assign reps to temporary organizational groups via the **Group Allocation GZ** info area.

For detailed information on the **Rep** info area and its sub-info areas, see **Rep** in the [CRM.Core Administrator Guide](#).

**Note:** The **Reps** menu item `ID_NET_REP_SEARCH` is available for the SU and users who have access to the **Rep** info area ID and to the `CRM process 1.5.2.2.104 Rep` `CRM process ID 50000670`. The Rep CRM process is disabled by default.

## Logins

The Logins section allows the administrator to apply module access rights, password settings, and lock policies

Select **Settings > Maintenance > Login** to manage logins for Aurea CRM users.

You can:

- Change login data, such as user name, password, module access rights, password settings, etc.
- Delete existing logins.

To create a new login for a rep, you need to access the **Rep** record and select **Configure Login** from the context menu.

For more information, see **Configuring Logins** in the [CRM.Core Administrator Guide](#).

---

**Note:**

- The **Logins** menu item `ID_NET_LOGIN_SEARCH` is available for the SU (Super User) and users who have access to the **Login File** info area (US).
- By default, users, except the SU, can only create/update/delete their login (US) record. Only the **Password** (US:38) and **Encrypted Password** (US:47) fields can be updated.. For logins of other users, only the following field values are displayed: **No.**, **User name**, **Alias**, and **Added on StNo**.
- Access to **all** US records is controlled via the `CRM process 1.5.2.2.1.6. Manage Logins CRM process ID 70000310`.
- The Manage Logins CRM process is disabled by default.
- Make sure you enable the Manage Logins CRM process for the SPECIAL user's rep record if it is not the SU. Otherwise, Aurea CRM.Web users cannot change their password at login. For further details see **Special User** in the [Aurea CRM.Web Installation Guide](#).

---

## Station

A station in Aurea CRM is the equivalent of a database. Each station is assigned a station number, which is used to track the source of data and coordinate communication between stations.

Select **Settings > Maintenance > Station > Stations** to access the **Station** (AS) infoarea. Use **Settings > Maintenance > Station > Station Configurations** to access the **Station Configuration** (T0) infoarea.



You can:

- Define stations. For further details, see **Defining a Station** in the [CRM.Core Administrator Guide](#).
- Assign languages to stations. For further information, see [Language](#) on page 52 and **Assigning Languages** in the [CRM.Core Administrator Guide](#).
- Define station configuration settings. For further details, see **Station Configuration** in the [CRM.Core Administrator Guide](#).

All available settings are also described in the online field help.

**Note:** The **Station** and **Stations** menu items `ID_NET_Stations` and `ID_NET_Station` are available for the SU and users who have access to the CRM process 1.5.1.2.100. Station CRM process ID 2502000.

The **Station Configuration** menu item `ID_NET_STATIONCONFIGURATION` and sub-list are available for the SU and users who have access to the CRM process 1.5.1.4.6. Station configuration CRM process ID 3117000.

The **Station Language** sub-list is available for the SU and users who have access to the CRM process 1.5.2.3.2. Station Language CRM process ID 3100000.

These CRM processes are enabled by default.

## Catalog Maintenance

Use Catalog Maintenance to edit variable catalogs.

Catalogs are lists of predefined values you can select when entering data.

The catalog maintenance feature is accessible from the **Settings > Maintenance > Catalog Maintenance** menu `ID_NET_CatalogMaintenance`. Besides, it is available as a link **Go to Catalog Maintenance** in the **Find or Add Catalog Value** popup when clicking on + **Add** in a catalog field.

### Note:

- The `ID_NET_CatalogMaintenance` menu item is available for the SU and users who have access to the **Catalog** info area (KA) and to the `CRM process 1.5.2.1.1. Catalog Maintenance CRM process ID 312600?`.
- The **Go to Catalog Maintenance** link is available for the SU and users who have access to the `CRM process 1.5.2.1.1. Catalog Maintenance CRM process ID 312600?`.
- The Catalog Maintenance CRM process is disabled by default.

For more information on catalogs and access rights, see [Catalogs and Access Rights](#) on page 36.

For details on catalogs and catalog maintenance in Aurea CRM.Win, see **Catalogs** in the [CRM.Core Administrator Guide](#).

The Catalog Maintenance feature in Aurea CRM.Web allows users to:

- Modify existing catalog texts and external keys (in all languages except the catalog base language).
- Lock/unlock catalog values.
- Sort catalog values.
- Add new catalog values.
- Add descriptions for catalog values.
- Define the availability of catalog values (valid from/to).

---

**Note:**

- This functionality is only available on the catalog maintenance station **Aut.Cat.Maintenance** checked in the **Station** record (AS). For further details, see [Station](#) on page 32.
  - If the **Catalog Maintenance** page is called from another station than the catalog maintenance station, an error message is displayed.
- 

The following actions can **only** be performed in **Aurea CRM.Win**:

- Delete catalog values.
- Merge catalog values.
- Modify existing catalog texts and external keys in the base language.
- Enter new or modify existing tenant numbers.

## Searching for Catalog Values

The Search feature of the Catalog Maintenance info area helps to locate catalogs based on a number, external key or text.

To search for catalog values:

1. Enter search criteria for the catalog values you want to change:

- **Info Area** and **Catalog**: If no info area is selected, all available catalogs are listed in the **Catalog** field. If an info area is chosen only the catalogs used in the selected info area are listed.
- Search catalog values by:
  - **TenNo**
  - **Ext. Key**
  - **Text**

These fields support wildcard search. The values accepted are star (\*) and interrogation mark (?).

- **Languages**: Select the languages you want to search. The `Languages.Editable` and `Languages.Visible` Web Configuration parameters determine, which languages are editable/visible. For further information, see [Languages.Editable/Languages.Visible](#).

`Languages.Editable` and `Languages.Visible` do not apply to the SU, i.e., the SU can view/edit all languages.

Your selection determines the language columns displayed in the search results. The base language and the active user's login language are checked by default. The base language cannot be unchecked.

---

**Note:** You need to specify either catalog, external key, or text at least.

If the catalog maintenance is accessed from a catalog field, **Info Area** and **Catalog** are automatically entered.

---

By default, the search results are sorted by catalog code.

The search results always display the **TenNo** and **Parent catalog** value columns. The **Catalog** column is displayed only if the resulting catalog values come from different catalogs, i.e., if no catalog was selected as search criterion. **Ext. Key**, **Sort** number, and **Lock** are displayed per language. If a catalog value doesn't exist in a language, the respective fields remain empty.

## Catalogs and Access Rights

Lock catalogs for Aurea CRM.Web users via the rights definition, tenant rights, or the Lock checkboxes in Catalog Maintenance.

In the **Search** mask info areas, catalogs and languages are displayed according to the user's access rights.

On the Catalog Maintenance page, access rights apply as follows:

- Info areas with **Deny Access** in the active user's rights definition are not shown in the **Info Area** drop-down list.
- Catalogs with **Deny Access** are not displayed in the **Catalog** drop-down list.
- Catalogs with **Deny New**: When selecting a catalog with **Deny New**, the catalog is displayed with a wavy red underline. Clicking on **+ New** displays a message.
- Catalogs with **Deny Update**: When selecting a catalog with **Deny Update**, the catalog is displayed with a wavy red underline. Clicking on **Edit** displays a message.

In the **Search Results** list, catalog values that are locked in the active user's rights definition **Deny Access** are displayed read-only and with all **Lock** checkboxes checked. **Deny New** and **Deny Update** rights settings for catalog values lead to corresponding error messages.

---

**Note:** By default, search queries automatically include all associated fields in the array fields used for catalogs. To ensure that the search result displays only the fields specified in the query set the `Query.DisableAutoAddArrayFields` parameter to true. For more information, see [Query.DisableAutoAddArrayFields](#).

---

For more details on access rights for catalogs, see [Defining Access Rights for Variable Catalogs](#) in the [CRM.Core Administrator Guide](#).

## Locking and Unlocking Catalog Values


Locking catalog values hides them from catalog fields in Aurea CRM.Web.

If a catalog value was selected before it was locked, the value is displayed grayed out in the drop-down list and cannot be selected.

To lock or unlock one or more values for all languages:

1. Select the catalog values from the search results list.
2. Select **Lock/Unlock selected** from the context menu.


To lock or unlock catalog values for one language:

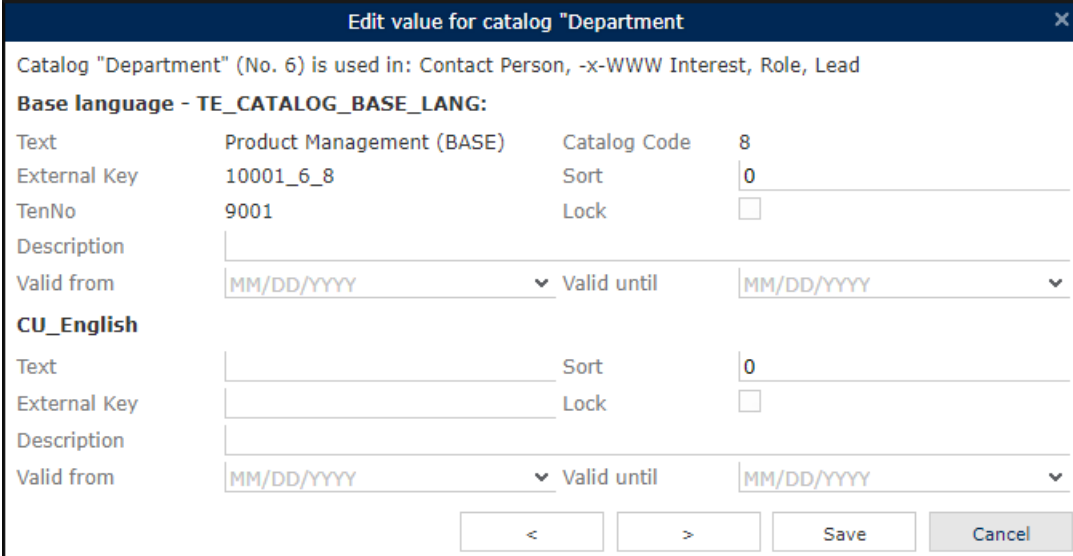
1. Select the catalog value from the search results list.
2. Click  **Edit**.
3. Check or uncheck **Lock** for the desired language.

For information on the behavior of catalog locks within language hierarchies. For details, see [Catalogs](#) in the [CRM.Core Administrator Guide](#).

## Editing Catalog Values

Use the Catalog Maintenance info area to edit catalog values.

To edit catalog values, double-click a record in the search results list or select a record and click  **Edit**. You might need to select the catalog first if you searched by **Text** or **Ext.Key** only.



**Edit value for catalog "Department"**

Catalog "Department" (No. 6) is used in: Contact Person, -x-WWW Interest, Role, Lead

**Base language - TE\_CATALOG\_BASE\_LANG:**

Text	Product Management (BASE)	Catalog Code	8
External Key	10001_6_8	Sort	0
TenNo	9001	Lock	<input type="checkbox"/>
Description			
Valid from	MM/DD/YYYY	Valid until	MM/DD/YYYY

**CU\_English**

Text		Sort	0
External Key		Lock	<input type="checkbox"/>
Description			
Valid from	MM/DD/YYYY	Valid until	MM/DD/YYYY

< > Save Cancel

The **Edit value** dialog lists the info areas where the respective catalog is used, and the parent catalog value, if applicable.

For the catalog value in the base language, only **Sort** number and **Lock** can be changed.

For values in all other languages, these fields can be changed:

- **Text**
- **Ext. Key**
- **Sort** number
- **Lock**

Use the arrow buttons to navigate between records and edit multiple catalog values. Changes are saved automatically before switching to the next record.

Create and uncheck the `Maintenance.CatalogMaintenance.ShowExtendedFieldsWeb` Configuration parameter to hide the **Description** and **Valid to/from** fields. For further details, see [Maintenance.CatalogMaintenance.ShowExtendedFields](#).

## Creating New Catalog Values

Use the Catalog Maintenance info area to create catalog values.

To create a new catalog value, select a **Catalog** in the search mask and click **+** **New**.

**New value for catalog "Department"**

Catalog "Department" (No. 6) is used in: Contact Person, -x-WWW Interest, Role, Lead

**Base language - TE\_CATALOG\_BASE\_LANG:**

Text		Catalog Code	
External Key		Sort	0
TenNo		Lock	<input type="checkbox"/>
Description			
Valid from	MM/DD/YYYY	Valid until	MM/DD/YYYY

**CU\_English**

Text		Sort	0
External Key		Lock	<input type="checkbox"/>
Description			
Valid from	MM/DD/YYYY	Valid until	MM/DD/YYYY

Save Cancel

**TenNo** of the current user and **Catalog Code** are entered automatically.

Enter **Text**, **Ext. Key**, **Sort** number, and **Lock** for the selected languages. If configured in the **Configuration** info area (MC), **Catalog** category, **Automatically enter external keys** option, the external key for the catalog base language is entered automatically.

If you are adding a value to a dependent catalog, make sure you select the desired **Parent catalog value** first.

To allow the maintenance of catalog values in a multiple language environment, providing the catalog value in both the catalog base language and the current login language is a precondition. Therefore, users that are logged on to Aurea CRM.Web in another language than the base language, and are allowed to add catalog values, are compelled to provide new catalog values in both the base language and their current login language.

## Mass Data Operations

Use Mass Data Operations to update or delete a large number of records. For example, when replacing one rep with another in all companies in a specific region.

The **Mass Data Operations** menu item the `ID_NET_BulkOperations` parameter is available when the following conditions are met:

- The Web Configuration parameter `UpdateServer.RepID` is defined. For further details, see [UpdateServer.RepID](#).
- The user has access to the CRM processes 1.5.2.8.1.1 **Update** CRM processes 3200000 and CRM processes 1.5.2.8.1.2 **Delete** CRM processes 3202000 . Both CRM processes are enabled by default.

**Mass Data Operations**

1. Select the info area you want to apply mass data updates or deletions to.
2. Enter your search criteria or define a filter.
3. Check whether the result corresponds to the records you want to update/delete.
4. Perform the desired mass data operation.

Select Infoarea:

---

**Search Companies**

Company  Country

From postal code  To postal code

Rep 1 ID

With me

max. 15 Records per page

---

**Search Results** << 1 - 15 of 16 >>

Company	Lead St...	Country	Pos...	City	Street	Rep 1 ID
ABC AG	Cold	Austria	1040	Wien		<a href="#">Lisa Simpson</a>
AkiComp	Interest	Austria	1040	Wien	Landgasse 7	<a href="#">Lisa Simpson</a>
BemComp	Customer	Austria	1100	Wien	Neustraße 1	<a href="#">Lisa Simpson</a>
Krist OY	Regular C...	Finland	00120	Helsinki	Mannerheimintie 123	<a href="#">Lisa Simpson</a>
Minions Ltd.	Cold					<a href="#">Lisa Simpson</a>

To update or delete records using **Mass Data Operations**:

1. Select **Settings > Maintenance > Mass Data Operations** .
2. Select the info area where you want to change records.

Users can only select info areas for which a Search&List configuration `<infoAreaId>` exists in the current designer configuration hierarchy, and for which they have sufficient access rights. Access rights are taken into account as follows:

- **Deny Access:** The info area is not displayed in the drop-down list.
- **Deny Upd:** The info area is listed, but the **Edit** button, in the results header, is disabled.
- **Deny Deletion:** The info area is listed, but the **Delete** button is disabled.

---

**Note:** The **Do not display the info area used to define filters and queries (web)** flag **Rights definition > Additional Settings** is not taken into account.

---

3. Click **Show Search**.


4. Enter the desired search criteria and/or define a filter for the records you want to change.

The user's search settings are saved between sessions. The search settings for mass data operations are saved separately, i.e., the global search settings are not used or changed. For further details, see [Remember Search Settings](#). The `Search.RememberDynamicFieldsWebConfiguration` parameter is taken into account. For further information see [Search.RememberDynamicField](#).

5. Click **Start Search**.

6. Check the search results and redefine your criteria if necessary.

Once the server returns the total number of result rows, it displays in the result header. Check if the number of results matches the expected number of records to be updated or deleted.

You can change the contents of the list by clicking  **Modify List fields**. Your changes are saved in the roaming settings. For further information, see [User Settings](#) on page 25.

You can display **MiniDetails** and **QuickViews** for the result records. You can **not** edit single records.

7. Once you are sure about the result set, select the desired mass data operation:

- **Edit**
- **Delete**
- **Delete without Communication**

These buttons are configured in the `SearchResults` header of header group `BulkOperations` only **Edit** and **Delete** in **UPDATE\_DEFAULT** configuration.

---

**Note:** All operations apply to the currently executed search, i.e., changing a search criterion without again executing the search is not taken into account.


---



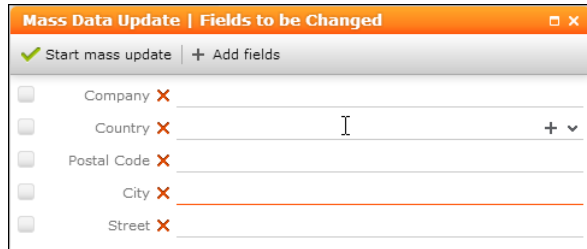
## Edit

Use Mass Data Operations to update a large number of records.

To update the result records:

1. Once you are sure that the search result contains only records you want to edit, click on  **Edit** in the search results header.


The **Mass Data Update** dialog contains the same functionality as the **Edit multiple records** dialog **Bulk Editor**. For further information, see [A\\_MultiEdit](#).



2. By default, all editable fields from the results list are displayed. Click on **Add fields** to change the fields list.

The following fields are not displayed in the **Mass Data Update** dialog:


- Decoded fields (Z-fields).
- Custom fields defined in `customFields.xml`.
- Fields flagged as VirtualCore, i.e., virtual fields with fixed IDs from 4200 to 4230, e.g., **Created by ID**. For further details, see **Fields with Fixed IDs** in the [CRM.Core Administrator Guide](#) in the 2nd table.
- Fields defined as read-only in the data model or via field attribute.
- Fields locked via access rights field rights and conditional field rights.

3. Make the desired changes: Changing a field value automatically checks the checkbox. Unchecking the checkbox resets the field value. Check the checkbox without entering a value to delete a field's content.
4. Click  **Start mass update** to launch the updating process: Once you confirm the message **Start mass data modification?**, a CRM.Server to-do (type 14) is created. For further details, see [CRM.Server To-Dos](#) on page 42.

## Delete and Delete without Communication

Use Mass Data Operations to delete a large number of records.

To delete the result records:

1. Once you are sure that the search result contains only records you want to delete, click on  **Delete** in the search results header.

If you select **Delete without Communication**, which is not configured in `UPDATE_DEFAULT`, the deletion is not communicated to other stations, i.e., the records are only deleted locally. For further details see **Communicating deleted records** in the [Aurea CRM.Win Administrator Guide](#).

---

**Note:** **Delete without Communication** is disregarded for info areas for which the **Mark as Deleted** option is enabled in the rights definition. For records in such info areas the **DelCd.** is activated instead.

---

2. The **Start mass data deletion?** message is displayed. Once you confirm with **OK**, a CRM.Server to-do (type 15 or 16) is created. For further information, see [CRM.Server To-Dos](#) on page 42.

## CRM.Server To-Dos

For each mass data operation CRM.Server creates a To-Do record that is executed according to the settings configured for maintenance jobs.

The text of the To-Do contains the following:

- **Edit:** `14;<infoAreaId>;<optional sequence>` e.g. `14;FI` or `14;MA;2d`
- **Delete:** `15;<infoAreaId>;<optional sequence>` e.g. `15;FI` or `15;NO;3m`
- **Delete without Communication:** `16;<infoAreaId>;<optional sequence>` e.g. `16;FI`

`<optional sequence>`: Use this optional parameter to perform mass data operations periodically. Format: `3d` = every 3 days, `2m` = every 2 months - case-insensitive, see examples above. In order to be able to use this functionality, you have to create a dialog where the To-Do text and the due/end dates can be entered.

Use the Web Configuration parameter `Maintenance.BulkExecutionStart` to schedule mass data operations for a specific time, see [Maintenance.BulkExecutionStart](#). Otherwise, the CRM.Server To-Do is added to the queue. CRM.Server executes polls for new To-Dos once a minute.

The search criteria and the fields and values to be changed for **Edit** are stored in two XML files `query.xml` and `defaultvalues.xml` and linked to the To-Do record.

All records that meet the defined conditions at the moment CRM.Server executes the query are updated or deleted.

---

**Note:** The To-Dos are executed within the user context, i.e., they are performed using the access rights of the user who defined and launched the mass data operation.

---

The update or deletion processes are logged in the Process Tracking (T3) info area for module "server". The **Text** field contains the rep ID of the user that started the mass data operation.

After the update or deletion process has been completed, you receive a message.

Errors are logged in the `massdata_log.txt` file stored in the Aurea CRM database. The log file is linked to the message: For each record that could not be processed, it contains a line `<error number>: <reference text> | <update protocol web URL to record>`.

For a list of error codes and their meaning, see [CRM.Core Administrator Guide](#).

## Import

Use the Import module to exchange data between Aurea CRM and external systems such as databases, spreadsheets, etc.

Select **Settings > Maintenance > Import** to import data into the Aurea CRM database. The import formats must be defined in Aurea CRM.Win. For further details, see **Import/Export** in the [Aurea CRM.Win Administrator Guide](#). Only file-based formats are supported in Aurea CRM.Web.

---

**Note:** The **Import** menu item `SimpleImport` is available for the SU and for users who have access to the `CRM process 1.5.3.1.8. Import CRM process ID 3604000`.

---

You can also configure an `Import` button, e.g., into the `FI SearchResults` header to allow users to import data into specific info areas. The `Import` button is assigned to the "Import" CRM process as well.

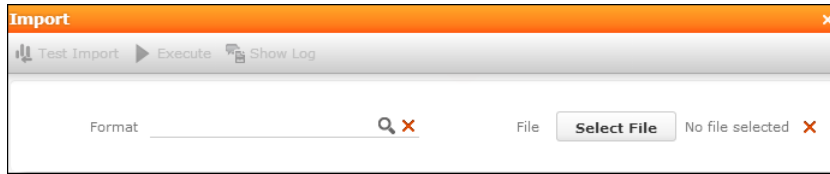
The menu item and button use the `SimpleImport` action template. If the action is executed in the context of an info area e. g., the Company search the import formats are automatically filtered to display only formats containing fields from that info area. If no context can be determined, e.g., in headers of custom forms, you can explicitly specify an info area using the `infoAreaId` input argument.

## Executing the Import

Execute the import job using the Import page on the Maintenance section or the Import button in a specific info area.

To import data:

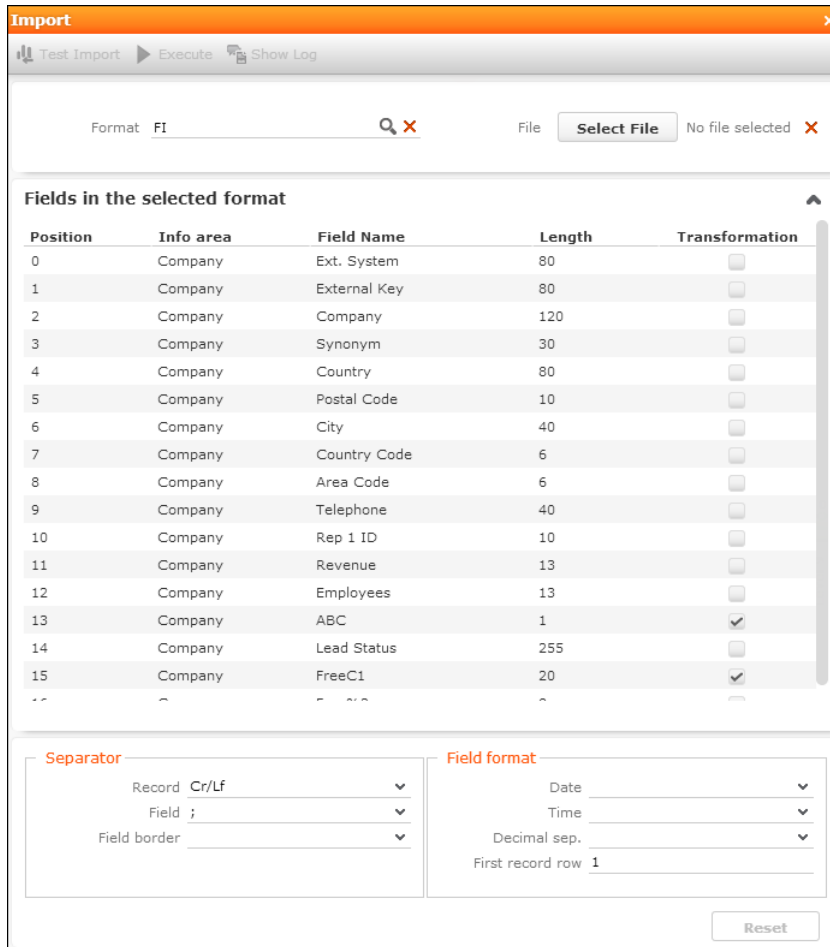
1. Select **Settings > Maintenance > Import** or click on an  **Import** button in a `Search` header, if configured.



2. Select an import **Format**. All file-based import formats are listed.

If the import is started from an info area or with `infoAreaId` defined in the action, only the import formats containing this info area are listed.

The selected format is saved in the current user's roaming settings, see [User Settings](#) on page 25.



The fields and import settings **Separator** and **Field format** areas are displayed, indicating the required structure of the import file. For detailed information on the import settings, for further details, see **File Format** in the [Aurea CRM.Win Administrator Guide](#).

A check mark in the **Transformation** column indicates that a transformation is defined for the field, see **Transforming Field Contents** in the [Aurea CRM.Win Administrator Guide](#).

Fields with **Duplicate previous field** enabled, see **Field Properties** in the [Aurea CRM.Win Administrator Guide](#) are hidden from this list, but handled correctly by the import itself.

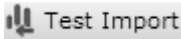
You can adapt the import settings. The changed settings are used for the current import and saved in the user's roaming settings; they are **not** saved in the import format.

Use the **Reset** button to reload the settings from the import format.


3. Browse for the import **File**. As soon as a file is uploaded to the server, the **Test Import** button is enabled.

The max. size of the import file is defined by the `Import.MaximumFileSize` Web Configuration parameter. For further details, see [Import.MaximumFileSize](#).

If the file size exceeds the limit, a message is displayed.


4. Click on  **Test Import**.

The **Import** popup displays a **Preview** of the imported records.

Info area	Field Name	File values	Imported values
Company	Company	Fratelli Pasquini	Fratelli Pasquini
Company	Synonym	Pasquini	Pasquini
Company	Country	1_2_6	1_2_6
Company	Rep 1 ID	000100002	000100002
Company	Revenue	50000	50000
Company	Employees	3	3
Company	ABC	2	B
Company	Lead Status	Interest	Interest
Company	FreeC1	quite a lot of text 01234...	<-- imported: quite a lot...
Company	Free%2	some text	some text 

The preview can display up to 50 records. If the import file contains more than 50 records, the header text displays "Preview of the first 50 records".

The following information is displayed:

- Values that are transformed on import are highlighted in green.
- Values that can not be imported, e.g. because the field type does not match the value, are marked with an . A description of the problem is displayed in the tooltip.
- The full contents of large field values are shown on mouse-over in a tooltip.


5. Once you are sure about the data that is imported, click on  **Execute**. The **Execute** button is disabled until **Test Import** has been clicked.

After confirming the warning message with **OK**, the import is executed.

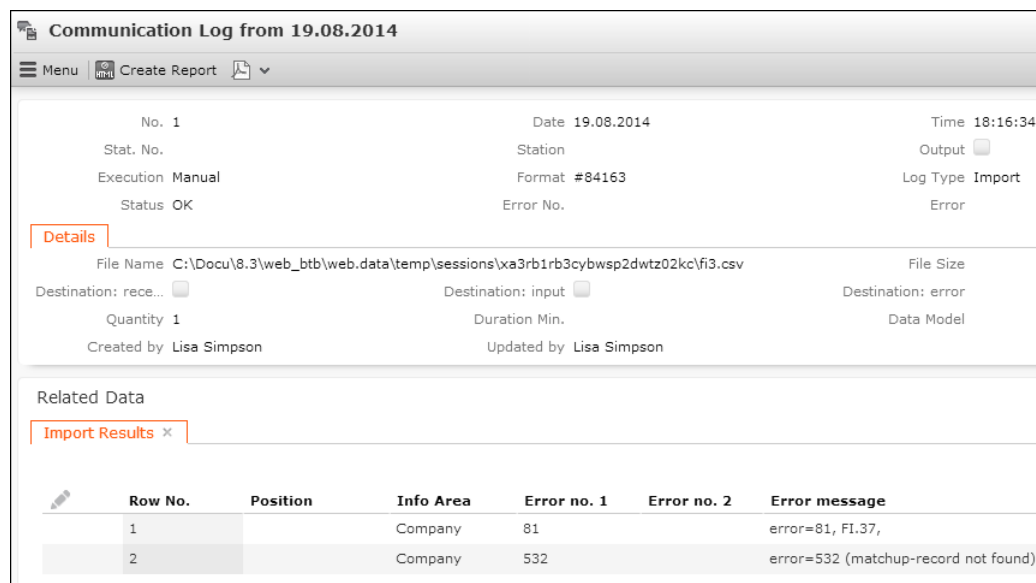
The **Import** popup displays a summary of the processed records:

Companies
New: 0
Updated: 2
Deleted: 0
Matched up: 2
Too similar: 0
Rejected: 1



This information is also written to the **Communication Log** record C3 that is created for each import. For further details, see [Communication Log](#) on page 47.

- Click on  **Show Log** to display the **Communication Log** record. The **Show Log** button is disabled until the import has been executed.

**Note:** The **Show Log** button is assigned to the CRM process 1.5.2.8.2.2. Communication Log CRM process ID 33030000. The CRM process is enabled by default.



**Communication Log from 19.08.2014**

Menu  Create Report 

No. 1	Date 19.08.2014	Time 18:16:34
Stat. No.	Station	Output <input type="checkbox"/>
Execution Manual	Format #84163	Log Type Import
Status OK	Error No.	Error

**Details**

File Name C:\Docu\8.3\web_btb\web.data\temp\sessions\xa3rb1rb3cybwsp2dwtz02kc\fi3.csv	File Size
Destination: rece... <input type="checkbox"/>	Destination: input <input type="checkbox"/>
Quantity 1	Duration Min.
Created by Lisa Simpson	Updated by Lisa Simpson

**Related Data**

**Import Results** x

Row No.	Position	Info Area	Error no. 1	Error no. 2	Error message
1		Company	81		error=81, FI.37,
2		Company	532		error=532 (matchup-record not found)

If errors occur during import, **Import Results** records C5 are created and linked to the corresponding C3 record.

One C5 record per error is created. If no errors occur, no C5 records are created.

For import errors where a record could not be updated, you can select **Open Affected Record** from the context menu to display the record.

You can view the **Communication Log** and **Import Results** records via **Administration > Jobs & Logs > Communication Log** and the corresponding sub-list.

For detailed information on error codes and messages, see **General Error Codes** and **Import/Export Error Codes** in the [CRM.Core Administrator Guide](#).

## Jobs & Logs

This section holds information about communication log, process tracking, and job manager.

### Communication Log

The Communication Log info area C3 contains information on all communication processes between stations as well as on all imports.

Select **Settings > Maintenance > Jobs & Logs > Communication Log** to access the info area.

---

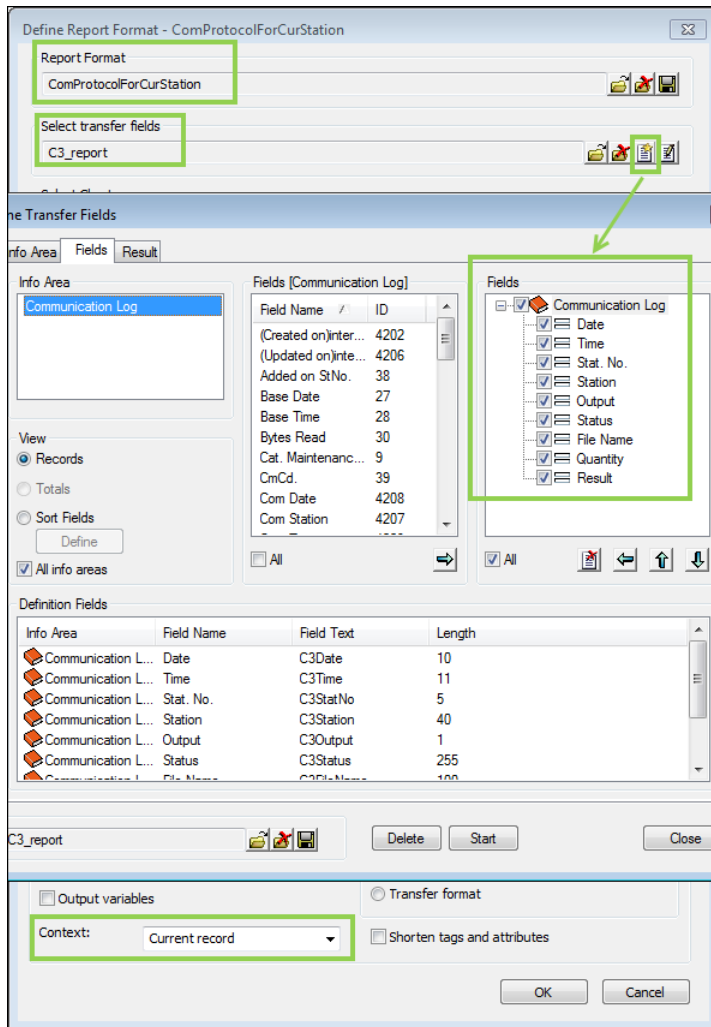
**Note:** The **Communication Log** menu item `ID_NET_CommLog` is available for the SU and users who have access to the `CRM process 1.5.2.8.2.2. Communication Log CRM process ID 33030000`. The CRM process is enabled by default.

---

You can export a communication log as an HTML report by selecting **Create Report** from the context menu `A_ExportComProtocol` in `M_ComProtocol`.

`A_ExportComProtocol` calls an `ExecuteReport` action that uses a report format called **ComProtocolForCurStation** `name` argument, no quotes. For further details, see [ExecuteReport](#) on page 423.

This report format is **not** part of Aurea CRM but must be created in Aurea CRM.Win.



For more details on Aurea CRM.Win reports, see [Creating Aurea CRM win Reports in Aurea CRM.Web](#) on page 863 and **Defining Report Formats** in the [Aurea CRM Win User Manual](#).

For further information about communication log, see **Communication Logs** and **Logging the Import** in the [Aurea CRM Win Administrator Guide](#).

**Note:** The `ExecuteReport` action and the XSL transformation are part of Aurea CRM.Web. Once you define the report format as indicated by the screenshot above, `A_ExportComProtocol` creates the report.

## Process Tracking

The Process Tracking info area is used to track critical processes that affect the Aurea CRM database to prevent multiple users from concurrently starting a process that can only be executed once.

Select **Settings > Maintenance > Jobs & Logs > Process Tracking** `ID_NET_ProcessTracking` to view process tracking (T3) records.



**Search Process Tracking**

Date  Station number   
 Module  Status   
 Active     
 + Add more ...


Max. 15 records per page

**Search Results** << 1 - 15 of 50+ >>

Date	Time	Process	Status	Module	Created by	Duration	Text
10.04.2014	12:20:07:852	(4009) Data Model	Open	Maintenance	Administrator		modify data model
10.04.2014	12:06:16:021	(3000) Rights	Open	Rights	Administrator		FT-5,61,"
10.04.2014	11:12:15:294	(1100) Components Instance	Open	web	WWW	1 Hours, 9 Mi...	
10.04.2014	11:04:44:779	(4003) Catalog Maintenance	Completed	Maintenance	Administrator		<DEL:1>
10.04.2014	11:04:43:742	(4002) Catalog Maintenance	Completed	Maintenance	Administrator	1 Sec	<READ:177><UPD:...

To search processes by type, you can use various predefined filters under **Add more**.

Use the **Active** flag to filter for active processes only. This virtual field can only be used in Search controls; in all other control types this field has no function.

The `SearchResults` header contains a `CleanupActiveProcesses` button  **Repair active processes** that allows you to end invalid processes. For details, see [Repairing Processes](#) in the [CRM.Core Administrator Guide](#).

**Note:** The `ID_NET_ProcessTracking` menu item is only available for the SU and users who have access to the CRM process 1.5.5.1.3. Process Tracking CRM process ID 3120000.

The `CleanupActiveProcesses` button is only available for the SU and users who have access to CRM process 1.5.2.6.2.1. Repair active processes CRM process ID 70000280.

The Repair active processes CRM process is disabled by default.

## Job Manager

Use Job Manager to view currently active server jobs.

Select **Settings > Maintenance > Jobs & Logs > Job Manager** to see active server jobs. The following jobs are displayed:

- Synchronization jobs with Aurea CRM.Web offline clients that are currently being executed.
- Synchronization jobs that are queued for future execution

Select **Remove from queue** from the context menu to remove a pending job.

**Note:** The **Job Manager** menu item `ID_NET_JOBMANAGER` is available for the SU and users who have access to the CRM process 1.5.2.6.5 Job Manager CRM process ID 50022000. This CRM process is disabled by default.

## Field Help

The field help contains information on all fields in the database. Field help is displayed as context-sensitive help tooltip in Aurea CRM.Web when the user hovers the mouse over a field label.

To edit the field help:

1. Select **Settings > Maintenance > Field Help** to access the **Field Help** info area (FH).

**Note:** The **Field Help** menu item `ID_NET_FIELDHELP` is available for SU and users who have access to the FH info area and the CRM process 1.5.1.7.2 Field Help CRM process ID 3136000.

The screenshot shows the 'Search Field Help' interface. At the top, there are search filters for 'Info Area Code' (set to 'as') and 'Field Number'. Below the filters is a 'Start Search' button. The main area displays 'Search Results' as a table with columns: Info Area, Field Name, Field Number, Short Text, and Comment. The table lists several fields, with 'Station Main Office' (Field Number 2) highlighted. Below the table is a 'Related Data' section with three panels: 'Field Help', 'Long Text', and 'Administrator Information'. The 'Field Help' panel shows details for 'Station (AS)/Main Office (2)', including its short text and long text. The 'Long Text' panel shows two entries for German and English. The 'Administrator Information' panel shows two entries for German and English.

Info Area	Field Name	Field Number	Short Text	Comment
Station	Stat.No.	0	Station number	
Station	Directory	1	Directory used to store communication files	
Station	Main Office	2	Determines whether the station is the main office	
Station	Aut.Cat.Maintenance	3	Determines whether the station is the catalog maintenance station.	
Station	SU Pe.No.	4		

For details on editing the field help, see **Field Help** in [CRM.Core Administrator Guide](#).

## Texts

The Text info area (04) is used to manage custom texts in the Aurea CRM database.

Texts that have been defined in the CRM.Designer database cannot be managed in the **Text** info area, see [Texts](#) on page 176.

Search Texts

---

Language No.

Info Area Code

04\_1\_SerNo

Record

Field Help

+ Add more ... v

Q x Text

Field Number

Format Type

Data Model

Max. 5 records per page Start Search

---

**Search Results** << < 1 - 5 of 15 > >>

	Langua...	Lang.(Text)	Text	Data M...	Format
	105	TE_Dutch	Klant potentieel	<input type="checkbox"/>	<input type="checkbox"/>
	105	TE_Dutch	Is de klant een referentie klant?	<input type="checkbox"/>	<input type="checkbox"/>
	105	TE_Dutch	Hoe is de feedback tijdens klant bezoeken?	<input type="checkbox"/>	<input type="checkbox"/>
<span style="font-size: 0.8em;">v</span>	105	TE_Dutch	Behoort de klant tot onze doelgroep sector?	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	105	TE_Dutch	Is de klant internationaal actief?	<input type="checkbox"/>	<input type="checkbox"/>

---

**Related Data**

All languages

<< < 1 - 15 of 22 > >>

	Langua...	Lang.(Text)	Text	Data M...	Format
			Is the customer part of our core sales sector?	<input type="checkbox"/>	<input type="checkbox"/>
	100	TE_German	Ist der Kunde Teil unserer Verkaufs-Kernbranche?	<input type="checkbox"/>	<input type="checkbox"/>
	101	TE_English	Is the customer part of our core sales sector?	<input type="checkbox"/>	<input type="checkbox"/>

Select **Settings > Maintenance > Texts** to access the **Text** info area (04).

**Note:** The **Texts** menu item `ID_NET_TEXTSEARCH` is available for the SU and users who have access to the **Text** info area (04) and to `CRM process 1.5.2.3.3. Text CRM process ID 3211???`.

For the menu action `ID_NET_TEXTSEARCH` the current user's login language is defined as the **Text** info area's default language field no. 4 **Language No.**, parameter `LOGINLANGUAGEinUPDATE_DEFAULTdefaultFieldValues 4:LOGINLANGUAGE`. Therefore the user's current login language is automatically entered in the text search **Language No.** field.

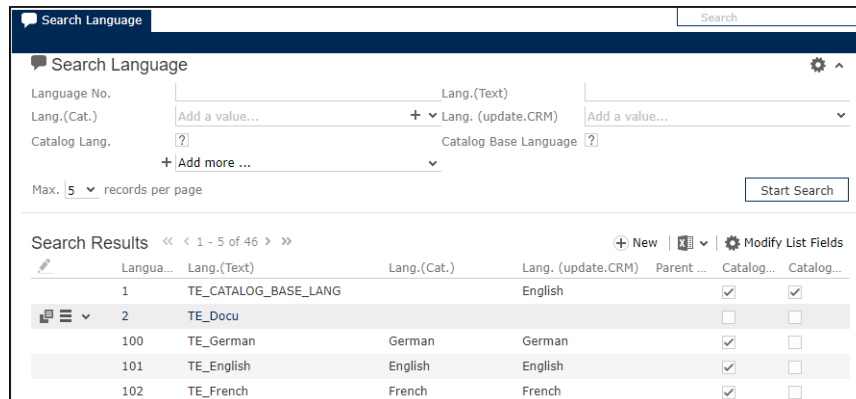
The sub-list displays all translations of the selected text. The translations are linked via the **04\_1\_StaNo** and **04\_1\_SerNo** unique fields key for translated texts.

Use the **New Text in Other Language** `A_NewTextReferenced` menu item to add new translations to an existing text.

**Note:** Use CRM.Translation tool to manage the translation of larger amounts of customized texts, see **CRM.Translation tool** in the [CRM.Core Administrator Guide](#).

## Language

Define all languages used by Aurea CRM in the Language info area and assign all stations the necessary languages.



Select **Settings > Maintenance > Language** to access the **Language** info area (00) where you can define languages and language variants. For further details, see **Languages** in the [CRM.Core Administrator Guide](#).

**Note:** The **Language** menu item `ID_NET_Language` is available for the SU and users who have access to the `CRM process 1.5.2.3.1. Language CRM process ID 3122000`. The CRM process is enabled by default.

You can assign languages to stations via the sub-list of the **Station** info area. For further details, see [Station](#) on page 32.

## Managing Checked Out Documents

The Manage Checked Out Documents info area offers an overview of all documents currently checked out.

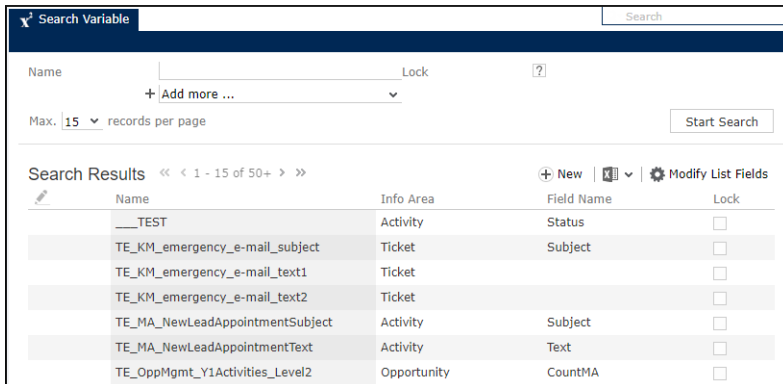
To obtain an overview of all checked out documents, select **Settings > Maintenance > Manage Checked Out Documents** : A list of all documents currently checked out by any rep is displayed.

You can undo the check-out of any document by selecting **Undo Check Out** from the record's context menu.

For information on documents in Aurea CRM.Web, see [Documents](#) on page 829.

## Variables

Use the Variable info area to define variables with values that change according to the current rep's data such as country, tenant, etc.



The screenshot shows the 'Search Variable' interface. At the top, there is a search bar and a 'Start Search' button. Below the search bar, there is a table with the following columns: Name, Info Area, Field Name, and Lock. The table contains several rows of variables, including '\_\_\_TEST', 'TE\_KM\_emergency\_e-mail\_subject', 'TE\_KM\_emergency\_e-mail\_text1', 'TE\_KM\_emergency\_e-mail\_text2', 'TE\_MA\_NewLeadAppointmentSubject', 'TE\_MA\_NewLeadAppointmentText', and 'TE\_OppMgmt\_Y1Activities\_Level2'.

Name	Info Area	Field Name	Lock
___TEST	Activity	Status	<input type="checkbox"/>
TE_KM_emergency_e-mail_subject	Ticket	Subject	<input type="checkbox"/>
TE_KM_emergency_e-mail_text1	Ticket		<input type="checkbox"/>
TE_KM_emergency_e-mail_text2	Ticket		<input type="checkbox"/>
TE_MA_NewLeadAppointmentSubject	Activity	Subject	<input type="checkbox"/>
TE_MA_NewLeadAppointmentText	Activity	Text	<input type="checkbox"/>
TE_OppMgmt_Y1Activities_Level2	Opportunity	CountMA	<input type="checkbox"/>

Use the **Settings > Maintenance > Variable** info area to define variables with values that change according to the current rep's data. For details, see **Variables** in the [CRM.Core Administrator Guide](#).

These variables can be used in workflows, triggers, and reports. In XML reports the `<variables/>` section lists all variable values valid in the current context `EmitVariables=true`. For further information, see [XML Exports & Reporting](#) on page 856.

Variables can also be used in filters and queries defined in CRM.Designer with syntax: `$VAR(<variable name>)`. For further details, see [Using Variable Expressions](#) on page 505. They cannot be used when defining filters and queries in Aurea CRM.Web.

For information on using variable values in processes, see [ReadVariablesDataProvider](#) on page 569.

To access the **Variable** info area:

Select **Settings > Maintenance > Variables** .

**Note:** The **Variables** menu item `ID_NET_VARIABLES` is available for the SU and users who have access to the **Variable** info area (Z5).

Default	Country	Var. Decimal
<input checked="" type="checkbox"/>		55,0000000
<input type="checkbox"/>	Germany	100,0000000
<input type="checkbox"/>	Austria	95,0000000
<input type="checkbox"/>	Italy	65,0000000

In the **Variable** info area (Z5) you enter a name for the variable and the info area and field to which the variable applies.

In the **Variable Value** info area (Z6) you define the values: The upper part of the details mask contains the rep related criteria for which this entry is valid, e.g., language, tenant no, etc.

**Note:** If you want to use **Condition for info area** as a criterion, you need to enter the filter string directly as text.

The lower part is used for entering the actual variable value. Its contents depend on the type of field selected in the parent record, e.g., if a date field is selected in Z5, only a date value can be entered in Z6.

### Date, time, number, Boolean and rep fields

Variable Value for myNumberVariable	
Info Area <b>Company</b>	Field Name <b>Revenue</b>
Stat. No.	Station
Z6-TenNo	Tenant Name
Rep Group ID	Rep ID
Language No.	Language
Country <b>Austria</b>	Login role
Default <input type="checkbox"/>	
<b>Variable Value</b>	
Catalog Text	Var. Rep ID
Var. Date	Var. Time (s)
Var. Decimal	Var. Logical <input type="checkbox"/>
Var. Number <b>100000</b>	Var. Text

## Variable catalog fields

Variable Value for myCatalogVariable	
Info Area <b>Installed Base</b>	Field Name <b>Product</b>
Stat. No.	Station
Z6-TenNo <b>300</b>	Tenant Name
Rep Group ID	Rep ID
Language No.	Language
Country	Login role
Default <input type="checkbox"/>	
<b>Variable Value</b>	
1st Par.Cat.Text <b>Server</b>	Parent Catalog <b>5</b>
Catalog Text <b>HQ Data Server</b>	Var. Number/Catalog <b>1</b>

The **Catalog Text** field accesses the **Catalog** info area (KA) where you can select the desired catalog value. The parent catalog value, if applicable, is automatically entered according to the field selected in Z6. Due to a UI restriction you need to save the record before selecting a catalog value.

**Note:** The Details control `Z6_VarCat` field group needs to contain the fields 41 `CatalogNo` and 15 `V_PCode` for the catalog search to work.

## Fixed catalog fields

Variable Value for myFixedCatalogVariable	
Info Area <b>Activity</b>	Field Name <b>Activity Type</b>
Stat. No.	Station
Z6-TenNo	Tenant Name
Rep Group ID <b>Administration</b>	Rep ID
Language No.	Language
Country	Login role
Default <input type="checkbox"/>	
<b>Variable Value</b>	
Index of Catalog Value <b>5</b>	Catalog Value <b>E-mail</b>

Enter the number of the desired catalog value. The value is entered automatically after saving the record.

## Generic link fields

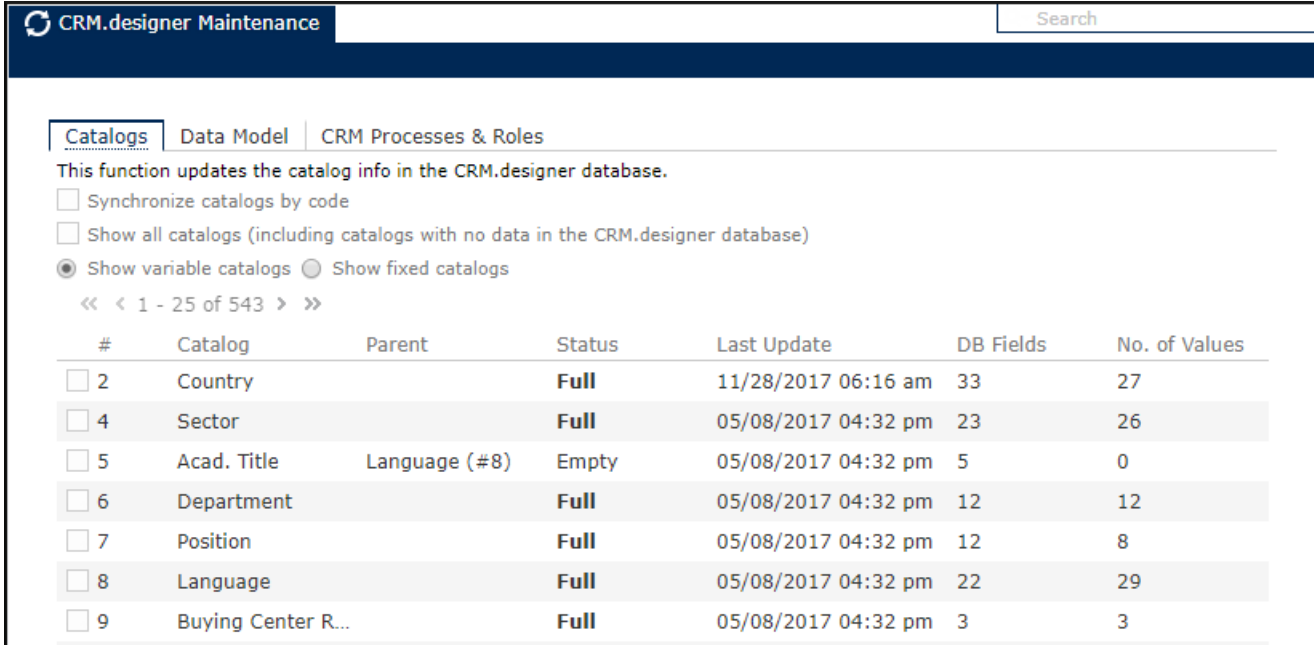
To use the generic link field **Link** as a variable value in Z6, you need to define a `Select-Function` for that field, see [Select-Function](#) on page 240 and [selectRecord for Generic Link Fields](#) on page 253.

## CRM.Designer Maintenance

Use CRM.Designer Maintenance to update the CRM.Designer database with the current catalogs, data model, CRM processes, and roles as defined in the Aurea CRM database.

Select **Settings > Maintenance > CRM.Designer Maintenance** to access the info area.

The functionality of **CRM.Designer Maintenance** is the same as described for CRM.Bulkloader. For details, see [Data Model, CRM Processes, Roles and Catalogs](#) on page 772.



The screenshot shows the CRM.Designer Maintenance interface. At the top, there is a search bar and a refresh icon. Below the search bar, there are three tabs: **Catalogs**, **Data Model**, and **CRM Processes & Roles**. The **Catalogs** tab is selected. Below the tabs, there is a description: "This function updates the catalog info in the CRM.designer database." followed by three checkboxes: "Synchronize catalogs by code", "Show all catalogs (including catalogs with no data in the CRM.designer database)", and "Show variable catalogs" (which is selected). There are also radio buttons for "Show fixed catalogs". Below this, there is a pagination control: "<< < 1 - 25 of 543 > >>". The main content is a table with the following columns: #, Catalog, Parent, Status, Last Update, DB Fields, and No. of Values. The table contains the following data:

#	Catalog	Parent	Status	Last Update	DB Fields	No. of Values
<input type="checkbox"/> 2	Country		Full	11/28/2017 06:16 am	33	27
<input type="checkbox"/> 4	Sector		Full	05/08/2017 04:32 pm	23	26
<input type="checkbox"/> 5	Acad. Title	Language (#8)	Empty	05/08/2017 04:32 pm	5	0
<input type="checkbox"/> 6	Department		Full	05/08/2017 04:32 pm	12	12
<input type="checkbox"/> 7	Position		Full	05/08/2017 04:32 pm	12	8
<input type="checkbox"/> 8	Language		Full	05/08/2017 04:32 pm	22	29
<input type="checkbox"/> 9	Buying Center R...		Full	05/08/2017 04:32 pm	3	3

The page consists of the following tabs, which are assigned to the following CRM processes:

- **Catalogs:** Allows you to update the catalog information in the CRM.Designer database with new and changed catalog values. For further information, see [Updating Catalogs](#) on page 778. The **Catalogs** tab is assigned to CRM process 1.5.2.1.4. Designer: Catalogs CRM process ID 50050090.
- **Data Model:** Allows you to transfer the current Aurea CRM data model to the CRM.Designer database. For details, see [Updating the Data Model](#) on page 775.



The **Data Model** tab is assigned to CRM process 1.5.1.6.32. Designer: Data Model CRM process ID 50050010.

- **Processes & Roles:** Allows you to transfer the currently defined CRM processes and roles to CRM.Designer. For further information, see [Updating CRM Processes and Roles](#) on page 777. The **Processes & Roles** tab is assigned to CRM process 1.5.1.6.39. Designer: Processes and Roles CRM process ID 50050100.

All three CRM processes are disabled by default. Users only see tabs which they may access due to their CRM process configuration.

## Data Model

The Data Model level allows administrators to configure Aurea CRM's data model, such as info areas, fields, and catalogs to meet the requirements of your organization.

Select **Settings > Maintenance > Data Model > Fields** to edit field labels directly in the Aurea CRM data model. For details, see [Editing Field Names](#) on page 57.

Besides, you can configure an **Edit Field Names** button `DataModelFieldNames`, e.g., into the FI `Expand` header.

The **Data Model** menu `ID_NET_DataModel` and **Fields** menu item `A_DataModelFields` are available for the SU and users who have access to the CRM process 1.5.1.6.1. **Data Model** CRM process ID 3210000. The `DataModelFieldNames` button is assigned to CRM process 1.5.1.6.12. **Change Field Names** CRM process ID 3210011.

---

**Note:** Both CRM processes are disabled by default.

---

A single user can edit the data model at a time. If a user tries to access the data model while another user is modifying it, Aurea CRM.Web displays a message, and the page is displayed read-only.

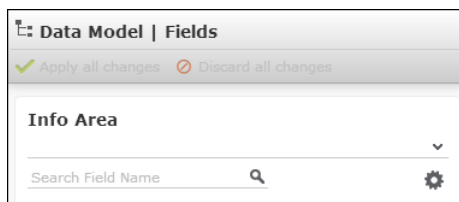
For detailed information on the handling of data model texts, see **Data Model > Info Area Options > Text** in the [CRM.Core Administrator Guide](#).

## Editing Field Names

Use the **Fields** page to edit field labels directly in the Aurea CRM data model.

To edit data model texts in Aurea CRM.Web:


1. Select **Settings > Maintenance > Data Model > Fields** or click on an **Edit Field Names** button, if configured.



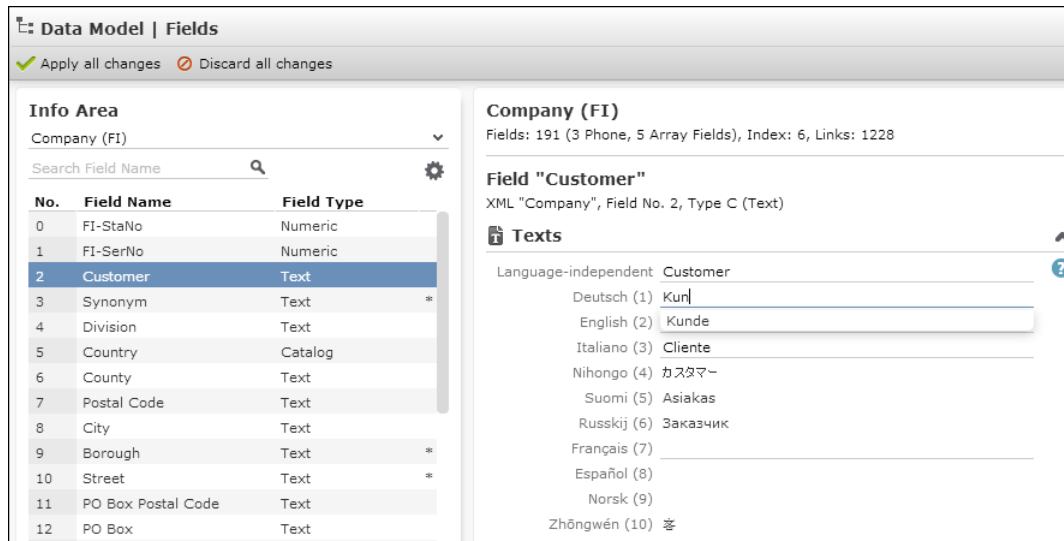
2. Select an **Info Area**.

If the page was accessed from an info area via an **Edit Field Names** button, the current info area is preselected.

All fields of the info area are listed. Fields with **Deny Access** in the current user's access rights definition are not listed.

Click on  to select which columns are displayed: **No.**, **Field Name**, **Field Type**, and **XML Name**.

Search for and select a field.



**Data Model | Fields**

Apply all changes Discard all changes

**Info Area**  
Company (F1)

Search Field Name

No.	Field Name	Field Type
0	FI-StaNo	Numeric
1	FI-SerNo	Numeric
2	Customer	Text
3	Synonym	Text *
4	Division	Text
5	Country	Catalog
6	County	Text
7	Postal Code	Text
8	City	Text
9	Borough	Text *
10	Street	Text *
11	PO Box Postal Code	Text
12	PO Box	Text

**Company (F1)**  
Fields: 191 (3 Phone, 5 Array Fields), Index: 6, Links: 1228

**Field "Customer"**  
XML "Company", Field No. 2, Type C (Text)

**Texts**

Language-independent Customer

- Deutsch (1) Kun|
- English (2) Kunde
- Italiano (3) Cliente
- Nihongo (4) カスタマー
- Suomi (5) Asiakas
- Russkij (6) Заказчик
- Français (7)
- Español (8)
- Norsk (9)
- Zhōngwén (10) 客

The right part of the screen displays the following:

- Data model information concerning the info area like number of fields, indices, and links.

---

**Note:** The number of links displayed here may differ from the number of links displayed in the CRM.Core data model accessed via the Aurea CRM.Win **Maintenance** module. Links marked as hidden are not counted in Aurea CRM.Web. Links to, e.g., rep fields are counted in Aurea CRM.Web but not in Aurea CRM.Win. Aurea CRM.Web reverse links, i.e., links from child to parent are counted for the parent info area as well.

---

- Data model information concerning the field, such as XML name, field no. and type. For further details, see [Data Types in Aurea CRM](#) on page 877.
- A collapsible **Texts** area containing the field labels.

---

**Note:** The **Texts** area is assigned to CRM process 1.5.1.6.12. Change Field Names CRM process ID 3210011.

---

3. Enter the desired texts. Changed fields are marked with a '\*' character in the field list.

The displayed/editable languages are determined by the `Languages.Visible` and `Languages.Editable` Web Configuration parameters. For details, see [Languages.Editable/Languages.Visible](#). (Defaults: display = all, edit = current language.)

`Languages.Editable` and `Languages.Visible` do not apply to the SU, i.e., the SU can view/edit all languages.

The text input provides a full-text "find as you type" search to search for existing texts in the data model. If you select an existing text, all available translations are entered as well.

You can also reuse existing texts by entering placeholders for field names, info area names, catalogs and catalog values:

- `{[FI,2]}` ... text of field 2 in the **Company** info area
- To reference an info area name, use the field number 4240: Enter `{[FI, 4240]}` to get "Company".
- `{[FIX4,0]}` ... text of the fixed catalog **Activity Type**, value 0 "Telephone"
- `{[VAR,2]}` ... text of the variable catalog **Country**
- You can also use a reference within a text, e.g., "My `{[FI,2]}` (Private)".

---

**Note:** To use a reference for all languages, enter it in the **Language-independent** field.

---

4. Click **Apply all changes** to update the data model texts. Or click on **Discard all changes** to clear all changes.
5. The changes are applied as follows:
  - The Aurea CRM Data Model is updated with all texts and translations entered by the user.
  - The CRM.Designer database is updated in the maintenance user's language or if undefined in the current user's language. All field names are updated in this language, not only the changed fields.

---

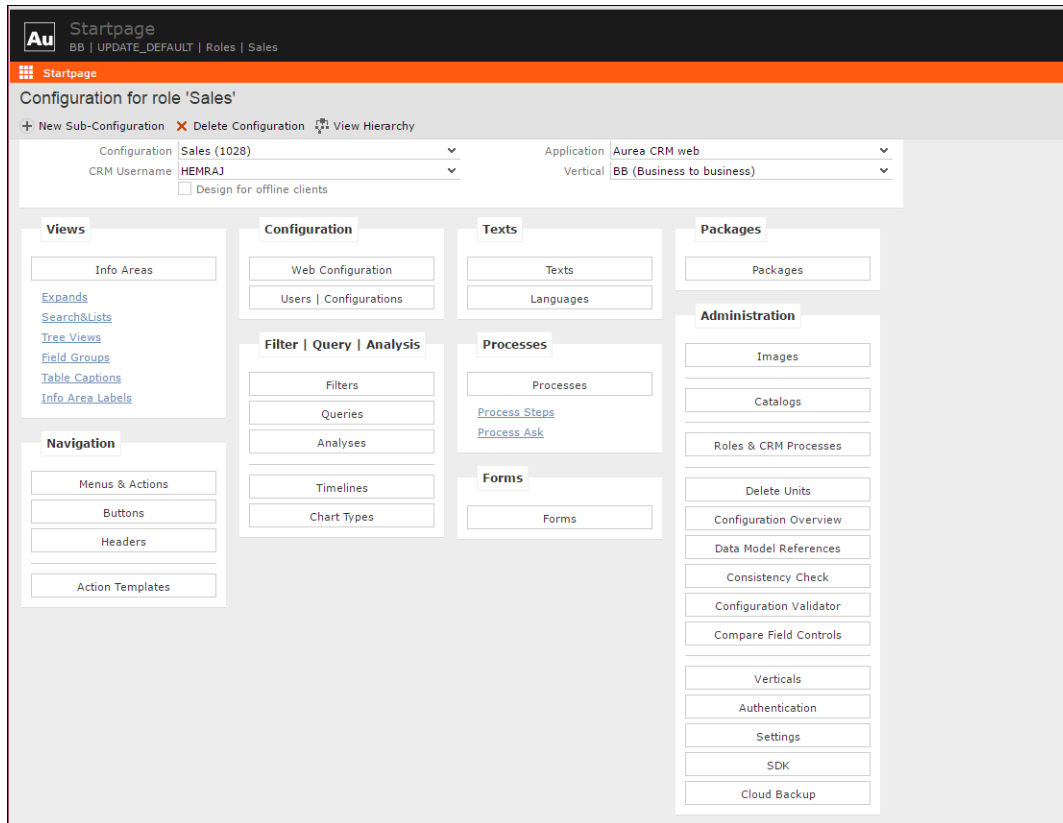
**Note:** CRM.Web highly recommends defining a dedicated maintenance user and enter a language in this user's rep record to avoid updating the CRM.Designer database in the wrong language, see **Maintenance User** in the [Aurea CRM Web Installation Guide](#).

---

# CRM.Designer

This chapter describes how to use CRM.Designer to customize Aurea CRM.Web for the processes and needs of individual customers.

CRM.Designer is a .NET application that allows you to customize configuration units like masks, lists, and context menus using a graphical, web-based user interface. The configuration data is stored in an MS SQL Server database. See [Aurea CRM.Web Installation Guide](#) for details. CRM.Designer connects to its database via ADO.NET. The screenshot below shows the start page of the CRM.Designer.



## Integration of CRM.Designer & Aurea CRM.Web

CRM.Designer holds the configuration of CRM.Web and allows the administrator to add customization to change the look and feel of the application.

Aurea CRM.Web setup consists of the **Aurea CRM.Web application framework** which is composed by a set of DLL's and ASPX pages, and the `UPDATE_DEFAULT` configuration. Together they form the Aurea CRM.Web application. The CRM.Designer database stores the `UPDATE_DEFAULT` configuration, and the Aurea CRM.Web application framework connects and accesses it online.

If you want to change the contents or functionality of info area views, navigation, processes, texts or many other areas of the application, you must define your configuration derived from the default and implement your changes in the database with CRM.Designer.

### How does it work?

Aurea CRM.Web needs **online access** to the CRM.Designer database using ADO.NET for MS SQL. The connection string is read from the `settings.xml` file stored on `update.configuration/OleDbDsn`. Aurea CRM.Web reads the necessary data from the CRM.Designer database and caches it in its application cache.

The `update.Configuration.dll` reads data from the CRM.Designer database and converts it into XML. The XML files are converted into local objects which are used to control Aurea CRM.Web interface.

### Checking XML files

To check the XML files, you need to specify the `XMLFilePath` entry in the `settings.xml` file pointing to a valid directory on your system, with appropriate rights for the ASP.NET user and set the `WriteToFile` entry to `true`.

```
<update.configuration>
[... ]
<WriteToFile>true</WriteToFile>
<XMLFilePath>C:\xml</XMLFilePath>
```

The XML files are then written to this file path.

### Using Debugging Mode

In a debugging mode, you can set the `<ConfigurationFromDB>` entry to `false`. The result is an offline mode where Aurea CRM.Web uses the XML files instead of the CRM.Designer database. All functions that write data to CRM.Designer, e.g., list settings, does not work.

### Using Cache Invalidation

Aurea CRM.Web caches data from the CRM.Designer database. Because of that, modifications of configuration units made in CRM.Designer are not immediately visible to web clients. To ensure that all web clients get the current content, restart the application by recycling the application pool, or restarting the web server using `iisreset`.

Alternatively, you can select menu item **Invalidate Cache** `ID_NET_INVALIDATECACHE` to invalidate all configuration units cached on the server. For further information, see [Invalidate Cache](#) on page 18. The next time a client requests a configuration unit, it receives the most recent version.

By default, invalidating the cache is also reload Aurea CRM.Web in the browser. To prevent it set the parameter `ResetSession` of menu action `ID_NET_INVALIDATECACHE` to 0. However, this does not invalidate the configuration units cached on the client, e.g., Web Configuration parameters.

For some configuration unit types, such as list fields, queries, analyses, and Web Configuration parameters the server-side cache is invalidated automatically.

### Working with Customer Admin Mode

Customers using a SaaS environment can set up a special **Customer Admin Mode** allowing administrative users of Aurea CRM.Web, to perform small customizations themselves. In this scenario, CRM.Designer is accessed from within Aurea CRM.Web. See [Customer Admin Mode](#) on page 865 for further details.

## Installing the Database

Follow the instructions on this topic to install the CRM.Designer database for the first time.

If you do not have a CRM.Designer database yet, and is making a fresh install, this is the **supported way to create a designer database**:

1. Create the database and upload the root configuration `UPDATE_DEFAULT` using CRM.bulkloader. For further details, see [CRM.bulkloader](#) on page 760.
2. Import the data model from the Aurea CRM database.
3. Upload the sub-configurations using CRM.bulkloader.

---

**Note:** If you already have a CRM.Designer database, you should always update the data model information of CRM.Designer **before** you upload sub-configurations.

---

For information on how to install or upgrade the CRM.Designer database, the default configuration, and CRM.Designer, see the **Setting Up the CRM.Designer Database** in the [Aurea CRM Web Installation Guide](#).

## Limitations of CRM.Designer

CRM.Designer is available only in English and should be used only by administrators.

CRM.Designer is a developer tool to edit the contents of the CRM.Designer database. The target group is technical personnel, like administrators. It is not designed for users of Aurea CRM.Web.

CRM.Designer is only available in English.

Aurea CRM Legacy Forms are deprecated and no longer supported by Aurea in version 8 and above. Forms like **LastQueryResult** and related menu items are not supported in versions after U7. You are requested to move to the new [Forms](#) on page 621.

## Using CRM.Designer

CRM.Designer allows administrators to have granular control of configuration and the look and feel of CRM.Web. Use CRM.Designer to customize the application to adapt to your business environment.

Aurea CRM.Web is shipped with standard configuration called `UPDATE_DEFAULT`. This default configuration allows you to revert to the original application configuration at any time. You cannot change or delete the `UPDATE_DEFAULT` configuration. Only derived configurations can be edited. All customizations have to be done in sub-configurations of this default configuration.

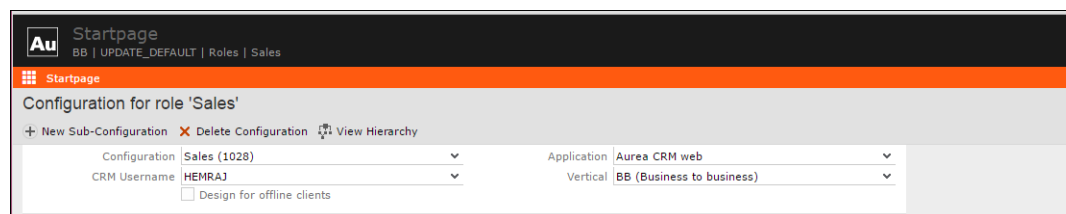
### Creating a New Configuration

You can either directly select the **Configuration** you want to modify, or you set it by selecting the **CRM Username** of the configuration you want to change. See [Users and Configurations](#) on page 74 for further information.

Click **New Sub-Configuration** to create a new configuration derived from an existing one. For more details, see [Creating a New Configuration](#) on page 75.

### Working with Existing Configuration

To customize elements of the Aurea CRM.Web application, you can change existing configuration units. For further details, see [Designing Configuration Units](#) on page 64. Configuration units are identified and referenced by their name and searched in the configuration hierarchy from bottom to top, i.e., from the user's configuration to `UPDATE_DEFAULT`.



The main window of CRM.Designer allows you to select the **Application**, the **Vertical**, the **CRM username**, and the **Configuration** you want to view or edit.

A **Configuration Unit** can be one of the following:

- Field group containing field controls defining which fields should be displayed on a search mask, a list or a details mask for a specific info area
- Text that is used in Aurea CRM.Web
- Menu, sub-menu or menu action of a context menu or the application menu
- Buttons
- Headers
- Action templates
- Processes
- Forms
- Queries

- Filters
- Analysis
- Etc.

### Using AQL Syntax

CRM.Designer allows you to define and edit queries and filters using the AQL syntax. See [AQL Syntax](#) on page 882 for details. You can also edit the AQL syntax in the AQL editor.

Before you change an existing configuration unit, you have to copy the unit to your configuration. By copying a unit, you take control of this unit. Subsequent changes of this unit in a parent configuration are **not** reflected in your Aurea CRM.Web application.

### Naming Conventions in CRM.Designer

Use `<InfoAreaId>` as the name for the default configuration unit for the info area. For example, `FI` is the name of the default Search&List configuration, field group, table caption, etc. used for the **Company** info area.


### Deleting a Configuration

Click the **Delete Configuration** button to remove an existing configuration. A popup message displays requesting a confirmation. Click **Ok** to confirm the deletion. For more details, see [Deleting a Configuration](#) on page 76.

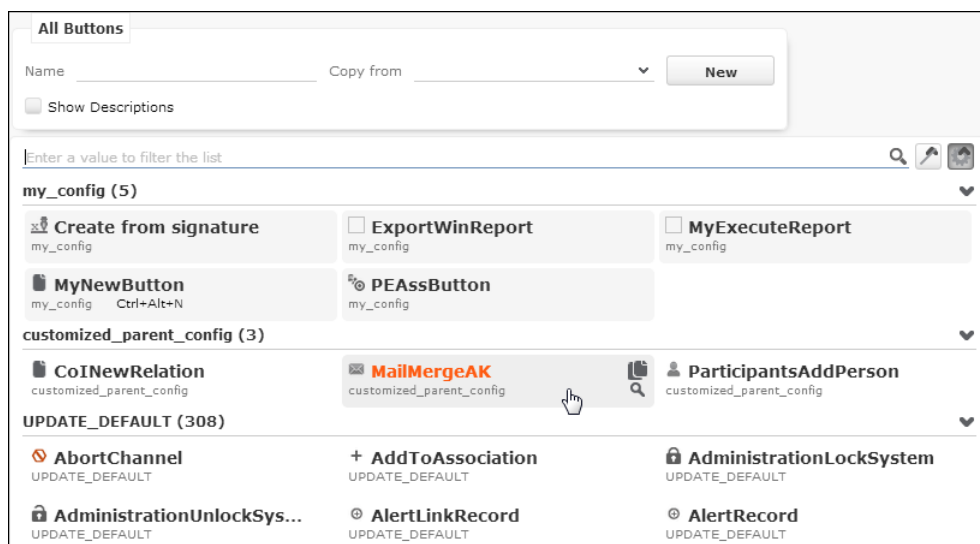
### Designing Configuration Units


Use CRM.Desginer to edit or create a configuration unit.

To edit or create a configuration unit:






1. On the CRM.Designer main page select the desired configuration or username.
2. Select the unit type by clicking the respective button, e.g., **Headers**, **Buttons**, or **Info Areas**. All CRM.Designer pages are also accessible via the  button.
3. Many pages in CRM.Designer offer a full-text "find as you type" search feature to find the item you want to edit quickly. Enter part of a configuration unit's name to filter the list of available items.





The search results are grouped by configuration, having the current configuration at the top. Click  **no groups**, to reset grouping. Configuration units that belong to the current configuration and, are therefore editable, are highlighted in both views.

#### 4. To edit an existing configuration unit:

- Click  **View** to view the definition of an item.
- Click  **Copy** to copy an item from a parent configuration to the configuration you are currently designing. Or click  **Copy to configuration**, if available, on the configuration unit's details page.
- Click  **Edit** to edit an item. Detailed information on all available settings for the various configuration units can be found in this book.
- Click  **Delete** to delete an item.

---

**Note:** You can only edit and delete items belonging to the current configuration.

If you delete an item from your configuration and an item with the same name exists in your parent configuration, the item from the parent configuration becomes valid in your configuration.

---

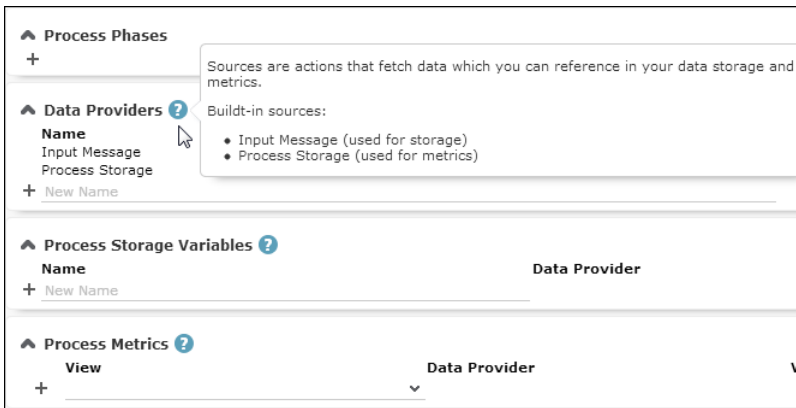
5. To create a new configuration unit, enter a name and click **New**. Or select an existing definition under **Copy from** before clicking **New**.

**Note:**

- The name identifies each configuration item. This name is unique within the vertical and the configuration within the configuration unit type.
- It is highly recommended to use the info area ID as the name for the default configuration item in the info area, e.g., FI is the default Search&List configuration used for the **Company** info area. For further information, see **Naming Convention in CRM.Designer** in [Using CRM.Designer](#) on page 63.

**Help Icons**

Many pages in CRM.Designer offer help icons  providing information for the current context, including syntax information, available options, etc.



**Adding Descriptions**

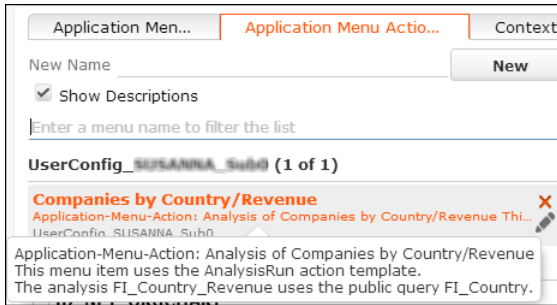
Use the Description field to document the configuration unit you changed or created.



You can resize the **Description** field using the  button.

**Note: Restriction:** Resizing the **Description** field is currently not supported by IE10.

The descriptions are displayed on the overview pages as well. Use the **Show Descriptions** checkbox to show or hide the descriptions.



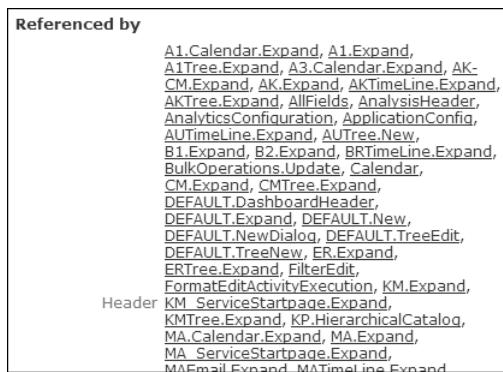
## Displaying References

The Referenced by section on button, header group, menu action, and table caption configuration pages displays all configuration units where the button, header, or menu action is currently used.

It is useful when you need to find all the instances that an item is used or when you plan to replace or remove all the instances.

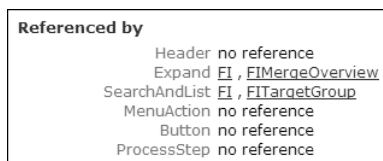
See the examples below:

- Buttons:

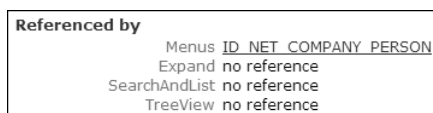


Headers belonging to a Header Group are listed as `<header group>.<header>`, e.g., `A1.Expand` or `DEFAULT.Expand`, special headers by their name only, e.g., `Calendar`.

- Header groups:



- Menu actions:



- Process Ask configurations:

**Referenced by**Steps [ChooseInfoAreaForRelation->SelectRecord](#)

**Note:** The usage information for processes and process steps can be displayed via the **Show Usage** button.

- Table captions:

**Referenced by**



Expand [FI](#), [FIMergeOverview](#), [FITree](#)  
 Expand Parent Relation no reference  
 TreeView [FI](#), [FI.Report](#)

Clicking on a link navigates to the respective configuration unit.


**Note:** Only selected references from certain configuration units are displayed. The **Referenced by** section **does not** show if a configuration unit is used in a form or the code.

## Showing or Hiding Arguments and Field Attributes

Use the Show and Hide buttons to control the display of argument and field attributes.

When defining an action or process step, use the  or  **Show all arguments** or **Hide empty arguments** buttons to toggle between views and add new arguments. The following input arguments are visible by default:

- Mandatory arguments that are not read-only.
- Arguments that have another than the default value.

All other arguments are hidden and can be displayed by clicking  **Show all arguments**.

When creating a new action or process step, all input arguments are visible by default.

The same buttons are used for showing or hiding field attributes in the Field Control Editor. The default is **Show all attributes**.

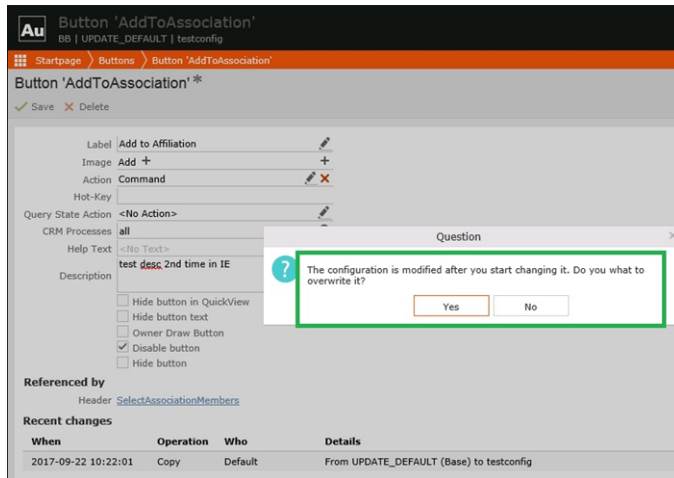
## Check-in and Checkout Support

The Check-in and Checkout feature alerts the user that the same item was modified previously by another user.

When multiple users are updating the same CRM.Designer element within a specified time frame the last user to update the element overrides the changes made by the previous user. Check-in and checkout support alerts the last user about the changes made by another user and gives him a choice of overriding the previous configuration changes. This feature ensures that the appropriate entry is made in the CRM.Designer.

**Note:** If the updates to the designer element are the same, then the user is not prompted about the previous change.

See an example in the screenshot below:



## Logging Changes of CRM.Designer Items

CRM. Designer allows you to generate log files to audit changes done in the system.

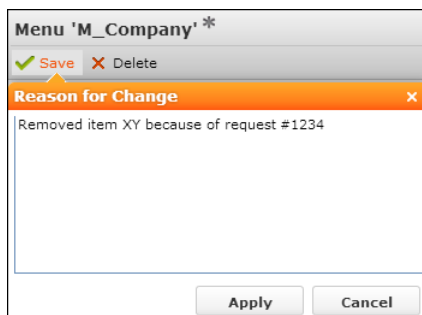
### Change History

Use the Change History log file for auditing the history of changes applied to the CRM.Designer database.

All changes made to configuration units are saved in the `PORTAL_UNITHISTORY` table of the CRM.Designer database.

### Popup window Reason For Change

For units belonging to a configuration that is not a user configuration, i.e., a configuration with an ID < 100,000, a change reason must be entered with a minimum of six characters. Otherwise, the configuration unit cannot be saved.



Clicking **Apply** saves the changes and the reason, clicking **Cancel** returns to the configuration unit definition without saving.

You can use the Markdown syntax to format your text. For further information, see <http://en.wikipedia.org/wiki/Markdown>.

You can disable the **Reason for Change** popup for specific configurations by adding the following parameters to CRM.Designer's `settings.xml` file:

```
<update.configuration.designer>
  [...]
  <DisableReasonForChange>
    <ConfigName>Test_Config</ConfigName>
    <ConfigName>Another_Test_Config</ConfigName>
  </DisableReasonForChange>
</update.configuration.designer>
```

You **cannot** disable the **Reason for Change** popup for the configurations delivered by Aurea CRM: `UPDATE_DEFAULT`, `UPDATE_DEFAULT Offline`, `UNITTEST_DEFAULT`, `update.tablet` and all configurations whose names start with `TE_`, i.e., all configurations belonging to an industry solution.

---

**Note:** The configuration names are case-sensitive.

---

### Recent Changes Panel

The change history is displayed at the bottom of each page in the **Recent Changes** panel. The fifteen most recent changes are shown.

When	Operation	Who	Details
2014-03-26 12:38:36	Update	Default	
2014-03-26 12:38:28	Update	Default	
2014-02-20 14:56:42	Update	Default	n/a
2014-02-20 14:55:59	Update	Default	n/a
2014-02-20 13:21:13	Copy	Default	From UPDATE_DEFAULT to my_config


For information on how to display all changes, and filter them by date or configuration, see the article “How to Analyze the CRM.Designer Change History” <https://support.aurea.com>.

The change history contains the following information:

- **When:** Timestamp of the change.
- **Operation:** Type of change (Create, Copy, Update, Delete)
- **Who:** The user who performed the change, see [Authentication](#) on page 748.
- **Details:** Information about the change, either automatically created or the **Reason for Change** text entered by the user (for non-user configurations).

The **Recent Change** panel is available for the following configuration units:

• Action templates	• Menus
• Analyses	• Processes
• Buttons	• Process Asks
• CRM.Designer Logins	• Process steps
• CRM.Designer Settings	• Queries
• Expand configurations	• Search&List configurations
• Field controls	• Table captions
• Field groups	• Text groups
• Filters	• Texts (on the respective text group page)
• Forms	• Timelines
• Header groups	• TreeView configurations
• Headers	• Web Configuration parameters (s. below)
• Menu actions	

Some of these pages, e.g., form designer, and field control editor offer a  **Recent Changes** button instead of the panel. The list of changes is displayed in a popup in this case.

On the **Web Configuration** page, an  icon is displayed for each parameter, showing the changes in a popup.

## Statement Log

Activate an SQL statement protocol to log changes made to the CRM.Designer database from both Aurea CRM.Web and CRM.Designer components.

The SQL statement protocol logs UPDATE, DELETE and INSERT statements as well as calls to stored procedures.

**Note:** Activating the statement log may have a negative impact on the system performance.

---

To activate the statement protocol, add `<StatementProtocol Enabled="true" />` to the `<update.configuration>` section of CRM.Designer's `settings.xml` file.

The resulting log is located in the CRM.Designer database in the tables `PORTAL_PROTOCOL` and `PORTAL_PROTOCOL_PARAM`. The table `PORTAL_PROTOCOL` contains the SQL statement and, if applicable, the user ID. For each SQL statement, the `PORTAL_PROTOCOL_PARAM` table includes a row for each parameter value.

The user ID is only logged if **EnableAnonymousAccess** is disabled. For further details, see [Settings](#) on page 750.

## Providing the Enterprise Serial Key to CRM.Designer

Provide the Enterprise Serial Key when upgrading to an Enterprise License to enable the extra features available.

Use CRM.Designer maintenance setup to facilitate this process. Perform the following steps to provide the enterprise serial key :

1. Upgrade the Aurea CRM to the latest version.

---

**Note:** Enterprise features are supported from CRM version 9.1.0 onwards.

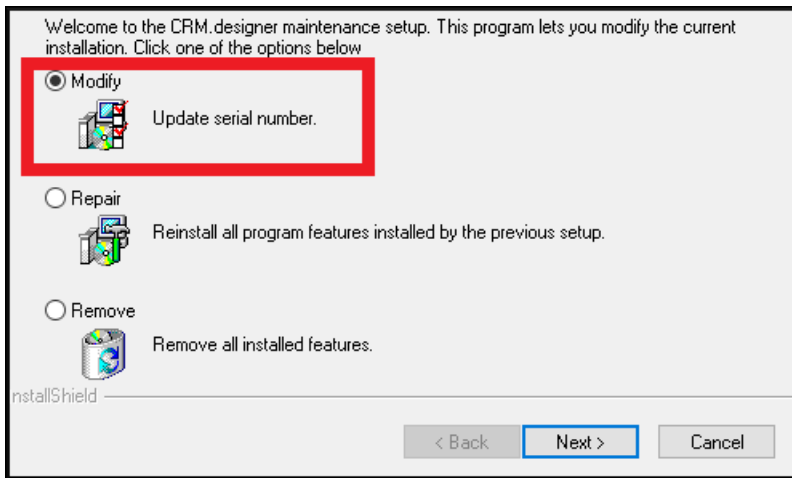
---

2. Replace the serial number that is saved in the CRM.Win installation directory in `...system\sys\Serialnumber.txt` file.
3. Initiate CRM.Designer to the new enterprise serial key using one of these methods: **CRM.Designer maintenance setup** or **Update the `settings.xml` file for the CRM.Designer** described below:

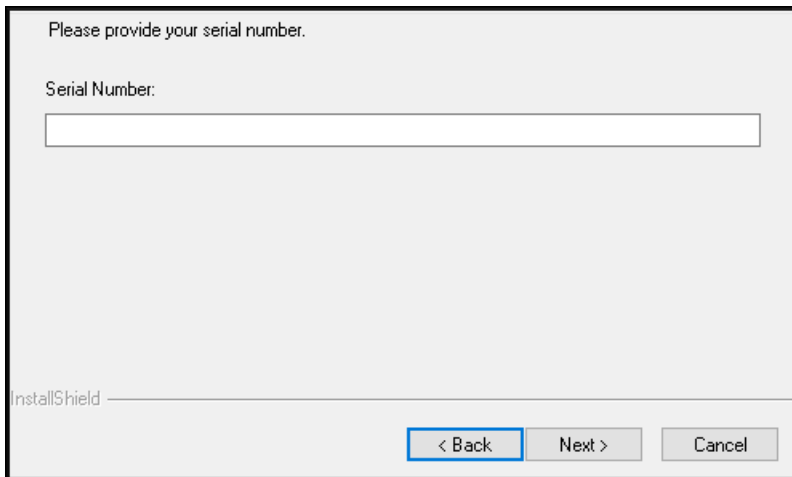
- **CRM.Designer maintenance setup**

1. Double click and open the **CRM.Designer** program. The CRM.Designer **Installshield Wizard** displays.





2. Click **Next**. The Serial Number page displays.



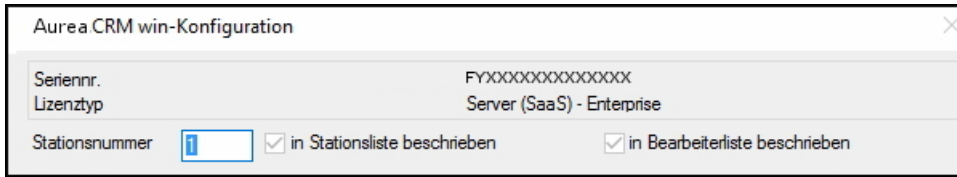
3. Enter the Enterprise **Serial Number** and click **Next** to continue with the maintenance setup.

- **Update the settings.xml file for the CRM.Designer**

1. Edit the settings.xml file located in the runtime\designer\settings folder.
2. Enter the <serial> sub-element of the <update.configuration> element. It should look as shown below.

```
<update.configuration>
<OleDbDsn>&designerDsn;</OleDbDsn>
<VersionCheckOff>>true</VersionCheckOff>
<Serial>XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXX</Serial>
</update.configuration>
```

4. Confirm the serial key change by performing the following steps:
  - a. Run `mmcfg.exe` located in the `\system\exe` directory. Confirm the message stating that the license rights have changed with 'Yes'.
  - b. The window should display the new serial key and the **License Type** field should show Enterprise. See the screenshot below:



5. Restart the CRM server application pool.
6. Login to the CRM.Web. The start page of the web now shows the mark [Enterprise].

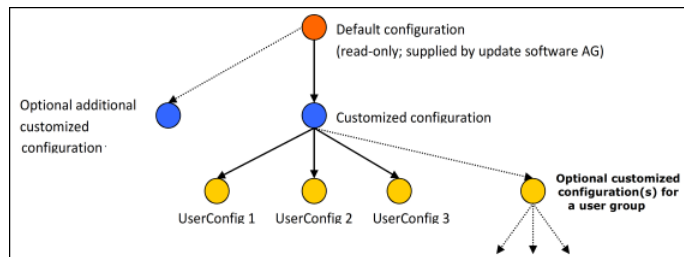
## Users and Configurations

This section describes how CRM.Designer configurations work and how to control assign configuration to users.

### Configurations

Aurea CRM.Web is certified and shipped with Aurea CRM's standard configuration called `UPDATE_DEFAULT`. This default configuration allows you to revert to the original application configuration at any time.

Each user has his configuration derived from the customer's sub-configuration:



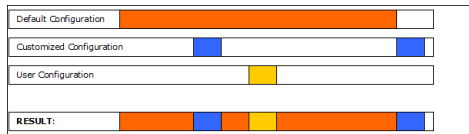
Use `UPDATE_DEFAULT` as the base for defining your customized configurations. If you create an empty customized configuration, this configuration is logically equal to its parent because it inherits the configuration units from the parent configuration.

---

**Note:** Note that you cannot change or delete the `UPDATE_DEFAULT` configuration. Only derived configurations can be edited. All customizations have to be done in sub-configurations of this default configuration.

---

To customize elements of the Aurea CRM.Web application, you have to change existing configuration units and add new ones. For details, see [Designing Configuration Units](#) on page 64.



There is no limit for the number of configurations stored in one CRM.Designer database. Neither is there a limit on how many of those are directly derived from update's default configuration nor a limit in the depth of the configuration.

To view the configuration hierarchy:

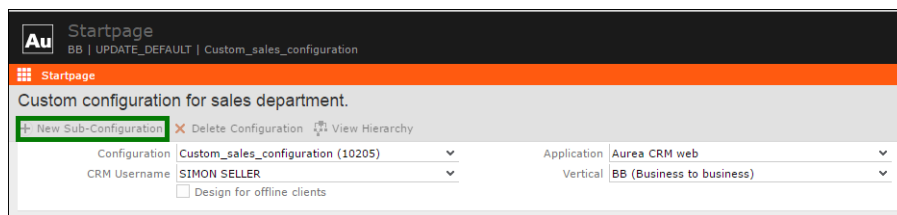
1. On the CRM.Designer main page select a configuration and click on **View Hierarchy**. See [Configuration Hierarchy](#) on page 85 for further information.

## Creating a New Configuration

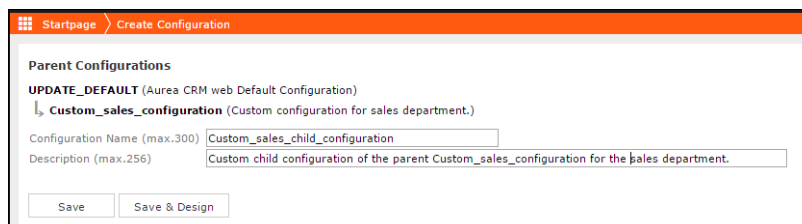
Create a new configuration to customize the different elements of CRM.Web.

To create a new configuration:

1. On the CRM.Designer main page select the parent **Configuration** and click on **New Sub-Configuration**.



2. Enter a **Configuration Name** and an optional **Description**.



3. Click on **Save**.

You can edit the configuration's settings at a later time. For further information, see [Editing Configurations](#) on page 84.

## Deleting a Configuration

Administrators can delete configurations that are no longer in use.

You can delete an existing configuration from the database if

- either the configuration has no sub-configurations.
- or all sub-configurations of this configuration do not have ID references to items of the configuration you want to delete.

---

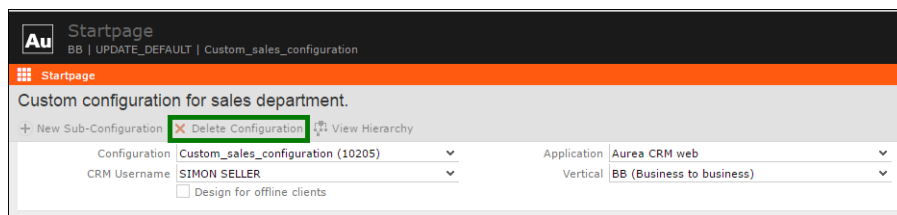
**Note:** You cannot delete Aurea's default configurations.

If a configuration you want to delete has sub-configurations, it is strongly recommended to delete those first or not to delete such a configuration at all.

---

To delete a configuration:

1. On the CRM.Designer main page, select a **Configuration** and click **Delete Configuration**.



2. A confirmation popup is displayed: If you confirm with **Yes – Delete Configuration**, the following changes are applied to the CRM.Designer database:
  - All configuration units of this configuration are deleted.
  - The configuration IDs of all texts of text groups of the parent configurations are set to NULL.
  - All sub-configurations of the deleted configuration get its parent as their new parent.
  - The configuration itself is deleted.

## Configuration Overview

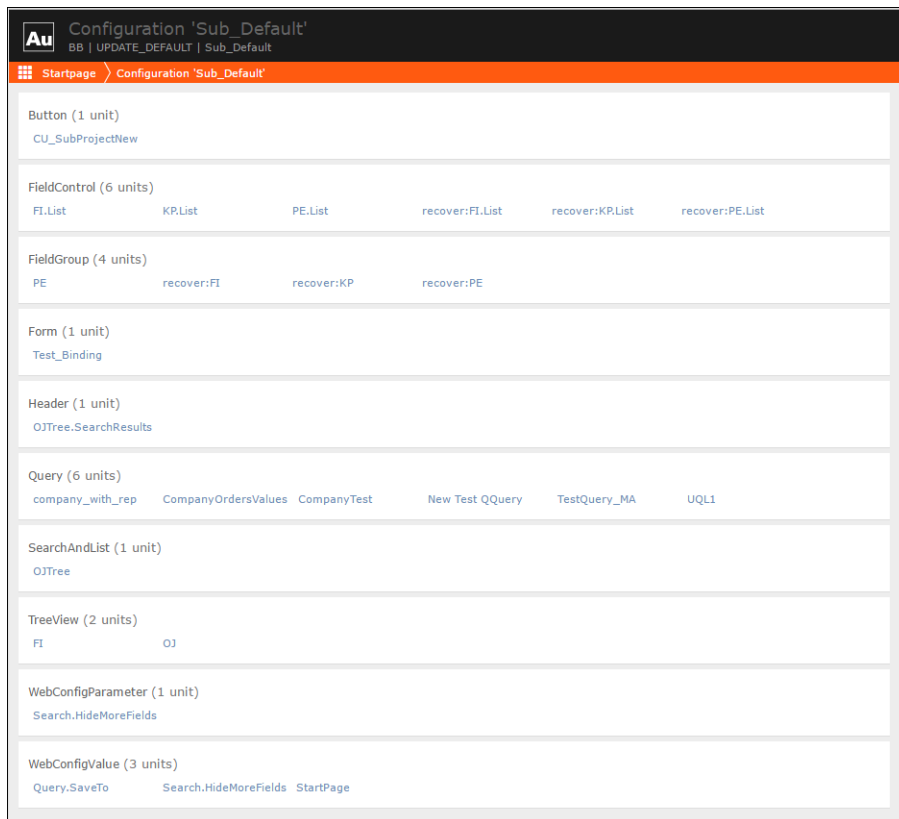
CRM.Designer main page offers an overview of the different settings available.

To access the configuration overview:

1. On the CRM.Designer main page in the **Administration** area, click **Configuration Overview** to display an overview of all elements filters, queries, field groups, menus, headers, etc. defined in the current configuration.

## Configuration Overview

2. Click on an item to navigate to the respective configuration unit.



To control the maximum number of units displayed in a configuration overview page, use the element `ConfigurationOverviewMaxUnitsToDisplay` in the `settings.xml` file located in the `runtime\designer\settings` folder. The default maximum number of units displayed is 100.

For example, to increase the maximum units displayed to 200 add the element to the `settings.xml`, as shown below:

```
<update.configuration.designer>
  ...
  <ConfigurationOverviewMaxUnitsToDisplay>200</ConfigurationOverviewMax-
UnitsToDisplay>
  ...
</update.configuration.designer>
```

## Transferring Configurations to Another CRM.Designer Database

You can transfer configuration data as XML files to other CRM.Designer databases using CRM.Bulkloader:

During the transfer you can move:

- configurations
- configuration units
- the complete database

To transfer a configuration:

1. Create a directory for the XML files.
2. Download the desired configuration to this directory. For further details, see [Downloading Configuration Data from the CRM.Designer Database](#) on page 761.
3. Create a configuration with the same name in the target database. The configuration should have the same parents as in the source database. If such a configuration already exists, it is overwritten on upload.
4. Upload the XML files containing the configuration to the target database. For further information, see [Uploading Configuration Data to the CRM.Designer Database](#) on page 764.

---

**Note:** Users, languages, logins, and authentication settings are **not** transferred via configurations or packages; you need to create a database dump to transport these items.

---

For more information about transferring **Configuration Units**, see [Packages](#) on page 711.

For further details on how to transfer the complete database, see [Database Dump](#) on page 767.

## Users | Configurations

Use the Configuration area to manage CRM.Designer configurations and assign users and groups to configurations.

---

**Fastpath:** **Main Page: Configuration > Users | Configurations**

---

Users / Configurations					
BC   UPDATE_DEFAULT   UNITTEST_DEFAULT					
Startpage > Users / Configurations					
View <input type="radio"/> Users <input checked="" type="radio"/> Configurations					
Root Configurations					
	Name	Parent Configuration	Role	Description	Users
0	UPDATE_DEFAULT	readonly root	root	update.CRM web Default Configuration	<a href="#">DefaultUser</a> , <a href="#">DefaultUser</a>
6	UPDATE_DEFAULT Offline	readonly UPDATE_DEFAULT	UPDATE_DEFAULT	update.CRM web offline Default Configuration	
7	update.mobile	readonly root	root	update CRM mobile Default Configuration	
8	update.touch	readonly root	root	update CRM touch Default Configuration	
15	update.tablet	readonly root	root	CRM.pad Default Configuration	
16	update.ras	readonly root	root	update RAS Default Configuration	<a href="#">DefaultUser</a>
Group Configurations					
	Name	Parent Configuration	Role	Description	Users
1001	UNITTEST_DEFAULT	UPDATE_DEFAULT	UPDATE_DEFAULT	Contains special configurations used by unit tests	
1002	TE_CORE	UPDATE_DEFAULT	UPDATE_DEFAULT	Template	
1003	TE_LSI	TE_CORE	TE_CORE	Template	<a href="#">MARTIN</a> , <a href="#">ONEKEY</a> , <a href="#">TE_LSI</a>
1004	TE_CONSUMERGOODS	TE_LSI	TE_LSI	Template	<a href="#">SERVICE_CG</a> , <a href="#">SUSANN T</a>
1005	XX_LSI_Country_Customer	TE_LSI	TE_LSI	Template	
1006	TE_LSI_Base	XX_LSI_Country_Customer	XX_LSI_Country_Customer	Backoffi...	Template

**Note:** The **Users | Configurations** page is configuration-independent and vertical-independent.

The **Users | Configurations** page consists of two views selectable via radio buttons:

- The **Users** view allows you to create users and user groups and assign configurations to them. For further information, see [Users View](#) on page 80.
- The **Configurations** view lists all existing configurations. See [Configurations View](#) on page 83 for details.

The configuration hierarchy is either **fixed**, i.e., it can only be changed explicitly or **dynamic**, i.e., the configuration is determined by the current organizational group of the rep.

### Fixed Configurations:

Changes of the configuration for the default user, the user's group, or a participation of this user in a **User Group** do not affect the user's configuration hierarchy. This is the default option.

If a user logs on to Aurea CRM.Web, the following checks are applied to determine the configuration for that user:

- If the user already has an entry on the **Users | Configurations** page, that configuration is applied.
- If there is no entry on the **Users | Configurations** page, but an entry for the user exists in the **User Groups** section, that configuration is applied.
- If no entry is found for the user or his group, the `DefaultUser` determines the configuration. The `DefaultUser` configuration is also used for the login page.

## Dynamic Configurations:

When using dynamic configurations, i.e., **dynamic parent** is checked for a user, the user is always assigned the configuration that is defined for his current organizational group, as set in the **Org. Group ID** field in the rep record, independent of the configuration assigned in CRM.Designer.

If no configuration corresponding to the organizational group exists, the `DefaultUser` configuration is applied.

## Users View

The Users view lists all currently defined users with their Configuration and Application.

The **Users | Configurations** page offers two views, the user and configuration view. When you select the **User** view the screen below appears with the following sections:

The screenshot shows the CRM interface for 'Users / Configurations'. The 'View' selector is set to 'Users'. The interface is divided into three main sections:

- Default Users:** A table listing default users and their configurations.
 

Default User Name	Configuration	On Change	Parent	Application
DefaultUser	UPDATE_DEFAULT	create new config	STATIC	Aurea CRM web
DefaultUser	update.tablet	create new config	STATIC	CRM.pad
DefaultUser	update.ras	create new config	STATIC	CRM.RAS
- User Groups:** A table listing user groups and their configurations.
 

User Group ID	Configuration	On Change	Parent	Application
#10000003	Development	modify config	STATIC	Aurea CRM web
#10002530	Sub_David	create new config	STATIC	Aurea CRM web
- User:** A table listing individual users and their configurations.
 

User Name	Configuration	On Change	Parent	Application
ALI	Testconfig3107	create new config	STATIC	Aurea CRM web
ASSVIEW	UNITTEST_DEFAULT	create new config	STATIC	Aurea CRM web
HEMANT	Hemant Yadav Configuration	create new config	STATIC	Aurea CRM web
HEMRAJ	Sales	create new config	STATIC	Aurea CRM web
KIFPAD	CRMpad_KIF Tests sp5	modify config	STATIC	CRM.pad
KIFPAD	User_KIFPAD_ChildOf_Hemant Yadav Configuration	modify config	STATIC	Aurea CRM web
LKZ	LKZConfig	modify config	STATIC	Aurea CRM web
MANUEL	Custom_sales_configuration	create new config	STATIC	Aurea CRM web
REVIEW	MAX_CONFIGURATION	create new config	STATIC	CRM.pad
SIMON SELLER	Custom_sales_configuration	create new config	STATIC	Aurea CRM web
SU	UPDATE_DEFAULT	create new config	STATIC	Aurea CRM web
SUSI	Sub_Default	create new config	STATIC	Aurea CRM web
TEST	Test New	create new config	STATIC	Aurea CRM web

- **Default Users:** The `DefaultUser` is used for users for whom no configuration is found.
- **User Groups:** You can assign a configuration to an organizational group. Users that belong to that group use this configuration. Users are assigned to organizational groups via the **Org. Group ID** field in the **Rep** info area ID:2.



To create a user group entry:

1. Enter the group's **Rep/Group ID** number (ID:0) as **User Group ID**: Skip the leading zeros and prefix the ID with a '#'.
  2. Click **+ Add**.
  3. Select a **Configuration** and **Application** for the group.
- **User**: Users that have their configuration are listed here. You can manually add a user by entering his Aurea CRM login name as **User Name**. Select a **Configuration** and **Application** for the user.

The **On change** option defines whether a new configuration, with the current configuration as its parent, is automatically created when the user changes his settings in Aurea CRM.Web. The default for **Default Users** and **User Groups** is to create new config. For further details, see [Editing Users](#) on page 81.

In a standard environment, assign your customized configuration explicitly to the `DefaultUser`, with **On change** set to **create new config**. See [Automatically Creating User Configurations](#) on page 82 for further information.

---


**Note:** By default, the SU (Super User) has the `UPDATE_DEFAULT` configuration assigned. If the SU needs to be able to change his configuration, e.g., modify list fields in Aurea CRM.Web, a different configuration must be assigned to this user.

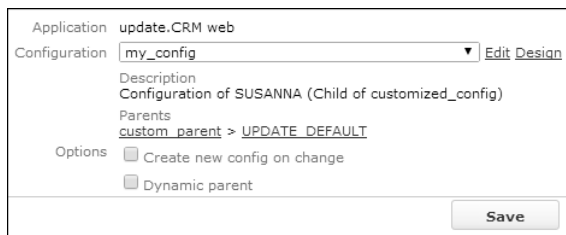
---

## Editing Users

The edit user feature allows administrators to select a different configuration and to assign a dynamic parent to the user.

To edit a user's configuration settings:

1. Open the CRM.Designer and go the **Users | Configurations** page.
2. Check the **Users** view radio button
3. Click on  **Edit** beside the desired user.



Application: update.CRM web

Configuration: my\_config [Edit Design](#)

Description: Configuration of SUSANNA (Child of customized\_config)

Parents: custom\_parent > UPDATE\_DEFAULT

Options:  Create new config on change  
 Dynamic parent

[Save](#)

The following settings are available:

- **Configuration:** Select the configuration you want to assign to the user.

Only configurations for the current **Application** are available. To assign a configuration, you need to create a new user first.

- **Create new config on change:** This setting determines the **On Change** setting.

- If checked, a new configuration with the current configuration as its parent is automatically created when the user changes his settings in Aurea CRM.Web **On change** = create new config.

Use this setting for configurations that are automatically assigned to users and that should not be changed from within Aurea CRM.Web.

Check this setting for configurations assigned to the default user and to user groups.

- If unchecked, the user's changes are saved in the current configuration **On change** = modify config.

Use this setting for configurations that users should be able to change directly from within Aurea CRM.Web.

- **Dynamic parent:** If checked, the user is always assigned the configuration that is defined for his current organizational group as set in the **Org. Group ID** field in the rep record, independent of the configuration assigned in CRM.Designer.

If no configuration corresponding to the organizational group exists, the `DefaultUser` configuration is applied.

#### 4. Click on **Save**.

Besides, the page offers the following links to configurations:

- Click on the **Edit** link to edit the current configuration's settings. For details, see [Editing Configurations](#) on page 84.
- Click on the **Design** link to return to the CRM.Designer main page and view or design the current configuration. See [Designing Configuration Units](#) on page 64 for further information.
- Click on a link under **Parents** to view or design the respective configuration.

### Automatically Creating User Configurations

User configurations are automatically created when Aurea CRM.Web modifies data in the CRM.Designer database to store changes in the user configuration, define lists or store private queries or analyses.

To automatically create users and configurations as children of your customized parent configuration:

1. Set the configuration of the `DefaultUser` to your customized configuration in the **Default Users** section of the **Users | Configurations** page.
2. Check **Create new config on change**.

For any user who logs on to Aurea CRM.Web for the first time and changes, e.g., a list definition, a user and configuration is automatically created in CRM.Designer. The user's configuration is created with your customized configuration as parent.

If you are using group configurations, the same mechanism is applied if the group configuration is set to **Create new config on change**.

For the administrator, when using the SU (Super User) account, the **On change** setting is ignored to ensure that the administrator always changes his current configuration. If the administrator has the parent configuration as his configuration, changing this configuration influences all users.

If you do not want the administrator to change configuration data for all users, create an extra user and configuration for the SU.

## Configurations View

The configuration view lists all currently defined configurations with their parent configuration and the users that use them

The **Users | Configurations** page offers two views, the user and the configuration view. When you select the **Configuration** view the screen below appears with the following sections:

The screenshot shows the 'Users | Configurations' page in Aurea CRM. It displays three sections: Root Configurations, Group Configurations, and User Configurations. Each section contains a table with columns for Name, Parent Configuration, Role, Description, and Users.

Root Configurations				
Name	Parent Configuration	Role	Description	Users
0 UPDATE_DEFAULT	root		Aurea CRM web Default Configuration	DefaultUser, SU
6 UPDATE_DEFAULT Online	UPDATE_DEFAULT		Aurea CRM web online Default Configuration	
7 update-mobile	root		Aurea CRM mobile Default Configuration	
8 update-touch	root		Aurea CRM touch Default Configuration	
16 update-tablet	root		CRMpad Default Configuration	DefaultUser
16 update-ras	root		update RAS Default Configuration	DefaultUser

Group Configurations				
Name	Parent Configuration	Role	Description	Users
1001 Sub_Emanuel	UPDATE_DEFAULT		null	
1003 Sub_Default	UPDATE_DEFAULT		Sub_Default	SUSI
1004 Sub_Joe	Sub_Default		Joe	
1006 Sub_DA_base	UPDATE_DEFAULT		null	
1007 Sub_DA	Sub_DA_base		null	
1008 Sub_David	UPDATE_DEFAULT		null	#10002530
1009 UserConfig_IV	UPDATE_DEFAULT		null	
1011 UNITTEST_DEFAULT	UPDATE_DEFAULT		Contains special configurations used by unit tests	ASSVIEW
1013 Sub_Christ	UPDATE_DEFAULT		Sub_Christ	
1022 Sub_Christ	UPDATE_DEFAULT		Sub_Christ	
1023 sv_conf	UPDATE_DEFAULT		null	
1026 Roles	UPDATE_DEFAULT		For assigning roles 'Sales', 'Marketing'...	
1027 Marketing	Roles	Marketing	Configuration for role 'Marketing'	
1028 Sales	Roles	Sales	Configuration for role 'Sales'	HEMBA
1029 Sales_SelfAdmin	Sales	Sales	Self-Admin-Configuration of Sales	
1030 Sales_SelfAdminDev	Sales_SelfAdmin	Sales	Self-Admin-Dev-Configuration of Sales	
1031 susi-config	UPDATE_DEFAULT		susi-config	
1037 Sub_J	UPDATE_DEFAULT		jonathani	
1041 Conf_From_DEFAULT	UPDATE_DEFAULT		null	
1043 Sub_David_Empty	UPDATE_DEFAULT		null	
1044 JaxiTablet	update-tablet		null	
1050 DalecTablet	update-tablet		null	
1052 ALEX	update-tablet		ALEX@CRMpad	

User Configurations				
Name	Parent Configuration	Role	Description	Users
10000 User_KIPPAD_ChildOf_Hemant Yadev Configuration	Hemant Yadev Configuration		Configuration of KIPPAD (Child of Hemant Yadev Configuration)	10000

- **Root Configurations:** The default configurations delivered with the Aurea CRM product.
- **Group Configurations:** Group configurations and configurations that were manually created. For details, see [Creating a New Configuration](#) on page 75.
- **User Configurations:** Configurations that have been created for a specific user, either manually or automatically by a user changing, for example, a list in Aurea CRM.Web.


These configuration types use IDs from the following number ranges:

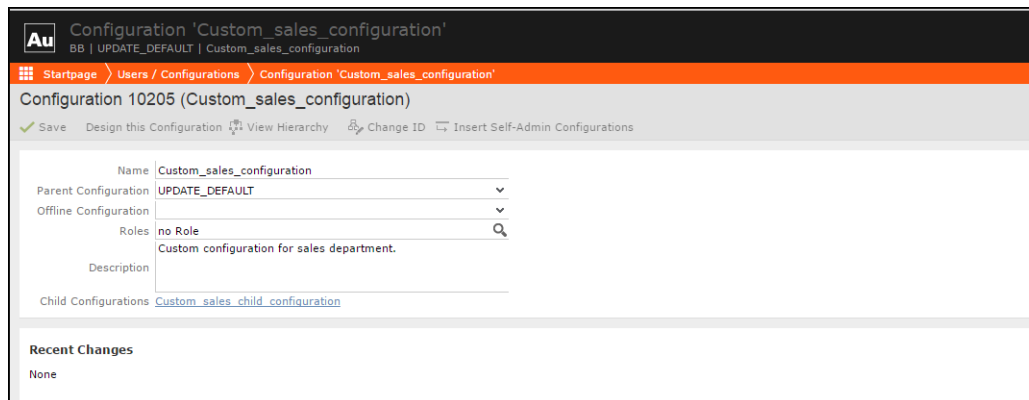
- 0-999 for root configurations
- 1,000-99,999 for group configurations
- >=100,000 for user configurations

### Editing Configurations

The edit configuration feature allows administrators to change name, parent configuration, roles and child configuration for a user.

To edit a configuration's settings:

1. Open CRM.Designer
2. Go to the **Users | Configurations** page
3. Select **Configurations** view radio button.
4. Click  **Edit** beside the desired configuration. Alternatively when in **Users** view you can click on a configuration link to access this page.



The following settings are available:

- **Name:** Since configurations are identified and referenced by their numeric ID, you can change a configuration's name.
- **Parent Configuration:** You can change the configuration hierarchy by selecting another parent.

---

**Note:** No sanity check is performed when selecting another parent. Use the **View Hierarchy** button to verify if your selection makes sense.

---

- **Roles:** For information on assigning roles to CRM.Designer configurations. For further information, see [Dynamic Configuration Assignment Based on Roles](#).
- **Child Configurations:** The current configuration's children. Click on a link to edit the respective configuration's settings.

5. Click on **Save**.

In addition, the page offers the following buttons:

- Click on **Design this Configuration** to view or design the current configuration. More details at [Designing Configuration Units](#) on page 64.
- Click on **View Hierarchy** to display the configuration hierarchy. See [Configuration Hierarchy](#) on page 85 for details.
- Click on **Change ID** to assign a new numeric ID to the configuration. For further information, see [Changing Configuration IDs](#) on page 85.
- Click on **Insert Self-Admin Configurations** if you are setting up customer admin mode, e.g., in a SaaS environment. See [Creating SelfAdmin Configurations](#) on page 868 for more information.

---

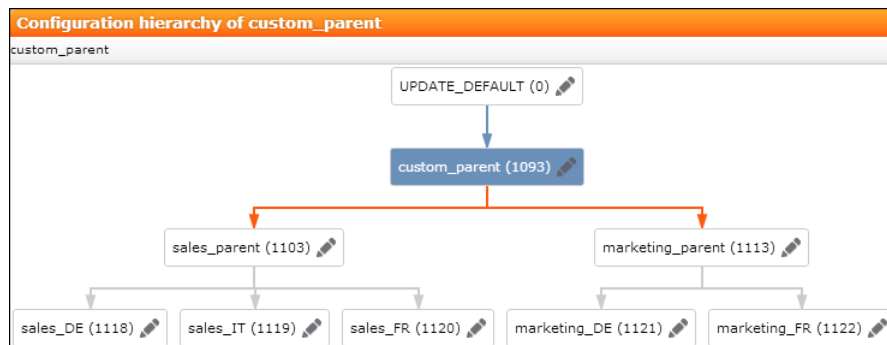
**Note:** The **Change ID** and **Insert Self-Admin Configurations** buttons are not available for user configurations IDs  $\geq 100,000$ .

---

### Configuration Hierarchy

The configuration hierarchy displays all parents and children of the current configuration.

User configurations, which IDs  $\geq 100,000$  are not displayed.



The **View Hierarchy** button is available on the CRM.Designer main page and when editing a configuration's settings. See [Editing Configurations](#) on page 84 for further details.

Clicking on **Edit** in a configuration bubble closes the popup and switches to the CRM.Designer main page with that configuration active.

Double-clicking a configuration displays the hierarchy for that configuration.

### Changing Configuration IDs

Use the Change ID button to create number ranges for configuration IDs.

CRM.Designer configurations are identified and referenced by their numeric ID. Use the Change ID button to create number ranges for configuration IDs and to shift configurations when inserting SelfAdmin configurations. For further information, see [Creating SelfAdmin Configurations](#) on page 868.

**Note:** It is strongly recommended to back up the CRM.Designer database before changing configuration IDs.

You need access to the complete configuration hierarchy to be able to use this feature, i.e., your authentication settings must be set to **Configuration = all**. For details, see [Authentication](#) on page 748.

To change a configuration's ID:

1. On the Edit configuration page click **Change ID**.

**Note:** This button is not available for user configurations IDs  $\geq 100,000$ .

The **Change configuration ID** popup is displayed.

The screenshot shows a dialog box titled "Change configuration ID 1113". The main window behind it is "Configuration 1113 (marketing\_parent)". The dialog has a red error message at the top: "There is not enough space to insert two...". Below this, it shows the current configuration ID as 1113 and a new configuration ID of 1113. There is a checked checkbox for "Update configurations with higher IDs". Below this, it says "The configuration IDs will be updated as follows:" and shows a table:

Name	Old ID	New ID
User_Group_A	1114	1123
User_Group_B	1116	1133
Sample_User	1117	1143
sales_DE	1118	1153
sales_IT	1119	1163
sales_FR	1120	1173
marketing_DE	1121	1183
marketing_FR	1122	1193

Below the table, it says "Leave the following gap between the new configuration IDs 10". There is a text input field for "Enter a reason for the change" with the placeholder text "Please enter a reason for your change." and "Ok" and "Cancel" buttons at the bottom.

The current ID is displayed as **New configuration ID**.

2. Enter the desired ID.

**Note:** You can not enter an ID  $\geq 100,000$ .

3. Determine how configurations with higher IDs should be handled:
  - **Update configurations with higher IDs:** If checked, all configurations with higher IDs are listed below providing a preview of the new IDs.
  - **Leave the following gap between the new configuration IDs:** Creates the new IDs with the specified gap. the default: is 10.
4. Click **OK**.

A confirmation message is displayed.

- Click **Cancel** to return to the popup and revise your settings or click **OK** to apply them.

Once you click on **OK** the IDs are updated throughout the CRM.Designer database.

## Deleting All User Configurations

Learn how to remove all user configuration in one click.

: Use with caution. This operation is not reversible.

To delete all user configurations:

- On the **Users | Configurations** page select the **Configurations** radio button.
- Click on **Delete All User Configurations**.

## Web Configuration

Web Configuration parameters are key-value pairs that define the behavior of Aurea CRM.Web.




**Fastpath: Main Page: Configuration > Web Configuration**

### Web Configuration

All available Web Configuration parameters are listed.

Parameters		Layouts
New parameter	Type	<input type="button" value="New"/>
Enter a web config name to filter the list		
Other (199)		
Activity.ActivityType_E-Mail Catalog ID of "E-mail" activity type	5	UPDATE_DEFAULT
Activity.ActivityType_Info Catalog ID for "Info" activity type	6	UPDATE_DEFAULT
Activity.CollisionCheck Collision Check	1	UPDATE_DEFAULT
Activity.CollisionCheckExcludedActivityTypes Activity Types to be excluded in the collision check	<empty>	UPDATE_DEFAULT

On the **Parameters** tab, you can:

- Search for a parameter and thus filter the list.
- Group parameters by topic or by configuration:
  - Click  **group by parameter** to sort parameters by their group, e.g., Activity, Calendar, etc.
  - Click  **group by config** to sort parameters by the configuration in which its value is saved.
  - Click  **no groups** to reset sorting.

Parameters that do not belong to a group are listed under **Other**, deleted parameters under **Deleted** at the end of the list.

- Navigate between parameters using the arrow and tab keys.
- Change a parameter's value. See [Setting Values of Web Configuration Parameters](#) on page 88.
- Create new parameters. See [Creating Web Configuration Parameters](#).
- Customize parameters and values. See [Modifying Web Configuration Parameters](#)
- Delete parameters. See [Deleted Web Configuration Parameters](#).

On the **Layouts** tab, you can design configuration layouts to allow users to change some settings directly in Aurea CRM.Web. See [Configuration Layouts](#) on page 93.

For a list of all Web Configuration parameters as defined in `UPDATE_DEFAULT`, see [Web Configuration Parameters](#) on page 93.

## Setting Values of Web Configuration Parameters

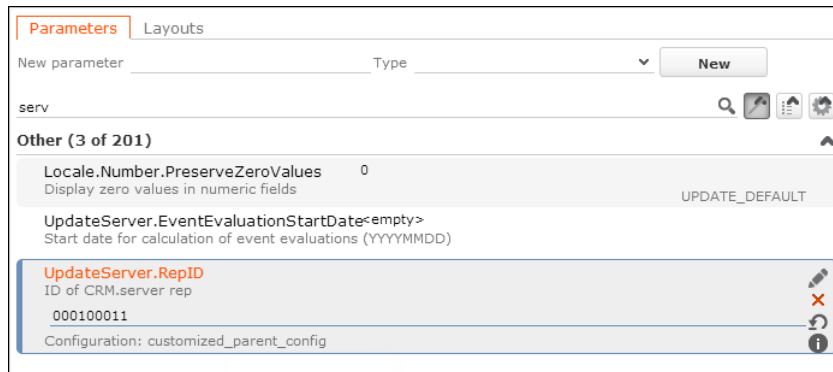
CRM.Designer allows administrators to edit the parameter of the CRM.Web configuration.

Each Web Configuration parameter without an assigned value in the current configuration inherits the value from the parent configuration.

To change the value of a Web Configuration parameter:

1. Search for the parameter you want to change.





The following information is displayed per list entry:

- the parameter's name
  - the parameter's label, i.e., **Display Name**
  - the current value, if available
  - the configuration where the value is currently defined
2. Enter the desired value. The input control depends on the parameter's type such a text field, drop-down list, etc. For details, see [Parameter Types](#) on page 89. The changed value is automatically saved as soon as you leave the entry field.
  3. Click **Edit** to view or edit the parameter's properties or values for parameters offering predefined values like Combobox, etc. For further information, see [Modifying Web Configuration Parameters](#).

Click **Delete** to remove the parameter from the current configuration.

Click **Clear** to remove the parameter's value from the current configuration. This resets the parameter to the value of the parent configuration.

Click **Info** to display the parameter's change history. See [Change History](#) on page 69.

## Parameter Types

Review this list to see all the parameters types ready for use in the CRM.Web configuration.

The following Web Configuration parameter types are available:

- **Catalog** for displaying catalog values in a drop-down list. Web Configuration parameters of type Catalog reference a catalog field from the Aurea CRM database, e.g., `Locale.BaseCurrency` gets its values from the **Currency** field in the **Exchange Rate** info area **WK**.
- **Character**.
- **Checkbox** for Boolean parameters.

- **Checkboxes** for selecting multiple values.
- **Combo:Config** for displaying a drop-down list containing all available designer configurations.
- **Combo:WriteConfig** for displaying a drop-down list containing all writable designer configurations.
- **Combobox** for displaying values in a drop-down list. For parameters of type Combobox containing up to three values. The values are displayed as radio buttons.
- **List** for entering multiple values as text comma-separated.
- **Number**.
- **Radiobutton** for choosing between two values.
- **JSON** Use this to type handle extra files on the server and make sure they get copied between the different environments. It shows an edit icon at the end of the row that offers a code mirror editor for convenient data input or viewing. This function is similar to the Hook function in Field controls.

## Customizing Web Configuration Parameters


This section describes the customization available in CRM configuration.

You can create, modify or delete web configuration parameters as per your needs.

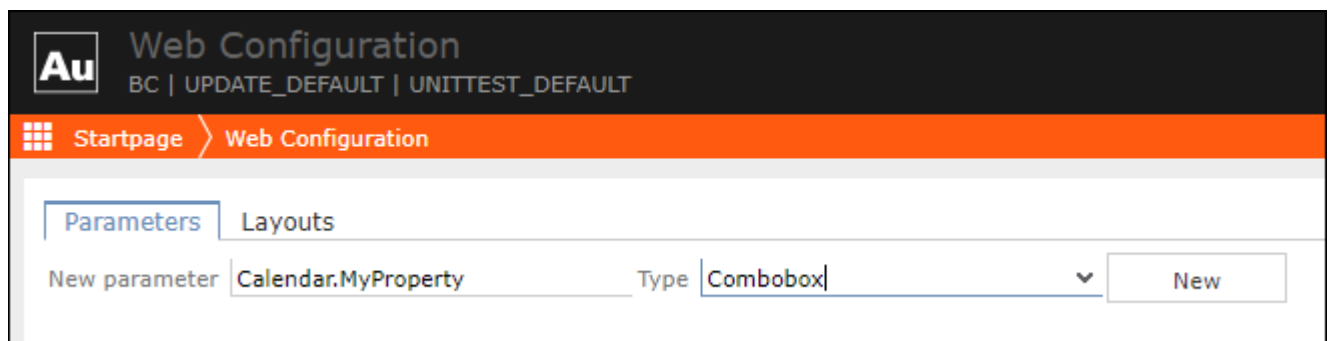
### Creating Web Configuration Parameters

To create a new Web Configuration parameter:

1. On the **Web Configuration** page enter a parameter name in the **New parameter** field.

To make use to the  **group by config** capability of the **Web Configuration** page, comply with the naming convention `<prefix>.<property>` e.g., Calendar.myProperty. `<prefix>` is used as group header.

2. Select a parameter **Type**. For further details, see [Parameter Types](#) on page 89.
3. Click **New**.




The screenshot shows the 'Web Configuration' page with a dark header containing the 'Au' logo and navigation links 'BC | UPDATE\_DEFAULT | UNITTEST\_DEFAULT'. Below the header is an orange breadcrumb trail: 'Startpage > Web Configuration'. The main content area has two tabs: 'Parameters' (selected) and 'Layouts'. Under the 'Parameters' tab, there is a form with a 'New parameter' label, a text input field containing 'Calendar.MyProperty', a 'Type' dropdown menu set to 'Combobox', and a 'New' button.

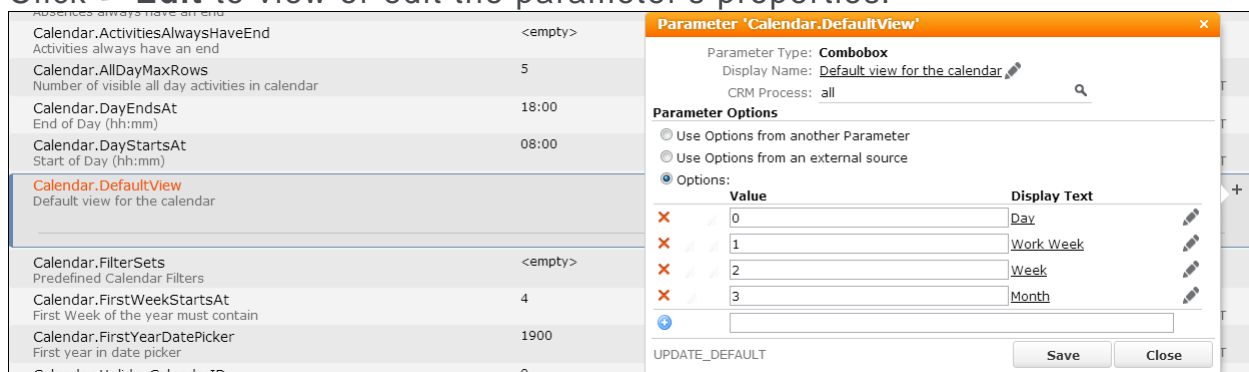
The parameter is created in the current configuration.

- Define the parameter's properties such as label, available values, etc. See [Modifying Web Configuration Parameters](#).
- Enter a value. See [Setting Values of Web Configuration Parameters](#) on page 88.

## Modifying Web Configuration Parameters

To modify a parameter:

- Search for the parameter whose properties you want to change.
- Click  **Edit** to view or edit the parameter's properties.



The screenshot shows a list of parameters on the left and a popup dialog for editing the 'Calendar.DefaultView' parameter on the right.

Parameter Name	Value
Calendar.ActivitiesAlwaysHaveEnd	<empty>
Calendar.AllDayMaxRows	5
Calendar.DayEndsAt	18:00
Calendar.DayStartsAt	08:00
<b>Calendar.DefaultView</b>	
Calendar.FilterSets	<empty>
Calendar.FirstWeekStartsAt	4
Calendar.FirstYearDatePicker	1900
Calendar.HolidayCalendarID	0

The popup dialog for 'Parameter 'Calendar.DefaultView'' shows the following configuration:

- Parameter Type: **Combobox**
- Display Name: **Default view for the calendar**
- CRM Process: **all**
- Parameter Options:
  - Use Options from another Parameter
  - Use Options from an external source
  - Options:
- Options table:
 

Value	Display Text
0	Day
1	Work Week
2	Week
3	Month

- The following properties are displayed in the popup dialog:
  - Parameter Type.** See [Parameter Types](#) on page 89.
  - Display Name:** Define/select a label to be displayed in the parameter's list in CRM.Designer as well as in Aurea CRM.Web if a parameter is part of a configuration layout. See [Texts](#) on page 176.
  - CRM Process:** Assign the parameter to a CRM process. See [Assigning CRM Processes](#)
  - the name of the configuration in which the parameter definition is saved.
- For parameter types offering predefined values for selection, e.g., Combobox, Catalog, etc. you can customize the available values. Select one of the following:
  - Use Options from another Parameter:** Select another Web Configuration parameter from the drop-down list.
  - Use Options from an external source:** Currently, only Aurea CRM catalogs are implemented. Enter the key for the external source in the **Value** field, e.g., for an Aurea CRM catalog the key is `$CAT;<info area>;<fieldnr>`. CRM.Designer is not connected to the Aurea CRM database. Therefore, you cannot see these values if you set the value for an Aurea CRM parameter. However, you can enter this value directly, but it is only displayed in Aurea CRM.Web.
  - Options:** Enter the values directly with the language-dependent **Display Text** to be displayed in Aurea CRM.Web. Example in `UPDATE_DEFAULT: Calendar.HolidayCalendarID`.
- Click **Save**.

## Parameter type Catalog

For parameters of type **Catalog** the property dialog is displayed as follows:

Parameter 'Locale.BaseCurrency'

Parameter Type: **Catalog**

Display Name: **Base Currency**

CRM Process: **all**

InfoArea: **Exchange Rate (WK)**

Field: **Currency (4005)**

UPDATE\_DEFAULT

Select the **Info Area** and **Field** of the catalog whose values you want to use.

The **Catalog** parameter type allows you to use catalog values which were synchronized using default language plus tenant number as input arguments for actions, e.g., as `defaultValues` in an `ExpandNew` action or `defaultFieldValues` or `fixedFieldValues` in a `Search` action. For details, see [defaultFields/defaultValues](#), [fixedFieldValues](#) and [defaultFieldValues](#). For more information on synchronizing catalogs, see [Catalog Value Synchronization](#).

**Syntax:** `$<WebConfigParameterName>`

### Example:

1. Create a Web Configuration parameter of type **Catalog** named `CountryAustria`.
2. Assign the value `Austria` to the parameter.
3. Use the Web Configuration parameter in the action, e.g. `defaultValues = $CountryAustria` in an `ExpandNew` action.

You need to create a Web Configuration parameter for each catalog value you want to use as a default value.



Since you can define a Web Configuration parameter's value once and use it multiple times, you can change it consistently in a central place at any time.

### Deleted Web Configuration Parameters


If a value was defined for a Web Configuration parameter and the parameter deleted afterward, this value is still valid, because it is not deleted physically and the key can be called from inside the code. Deleted parameters that still have a value are listed in an extra group **Deleted** at the end of the parameter list. Values of deleted parameters can still be changed.

If you find deleted parameters that you still need, you have to re-create the respective Web Configuration parameter.

To delete definitively a Web Configuration parameter that is no longer used anywhere in the code:

1. Delete the parameter's value by clicking  **Clear**,
2. Delete the parameter itself by clicking  **Delete**.

---

**Note:** This applies only to the configuration where a parameter is defined. If you click  **Delete** in a child configuration, the parameter is reset to the value in its parent configuration. For more information, see [Setting Values of Web Configuration Parameters](#) on page 88.

---

## Configuration Layouts

Use the layouts tab to design configuration layouts allowing users to change some settings directly in Aurea CRM.Web.

A configuration layout groups several Web Configuration parameters by topic.

Activity.ActivityType_E-Mail (Character)	Tab #1	X	↓	Layout		
Activity.ActivityType_Info (Character)						
Activity.CollisionCheck (Checkbox)			X	↓	Rep.ABCStrings (List)	
Activity.CollisionCheckExcludedActivityTy			X	↑	↓	Rep.Dialog.Tree.MaxResults (Number)
Activity.ImagePrefix (Character)			X	↑	↓	Tree.DefaultNodeSettings (Checkboxes)
Analysis.CountFormatting (Character)			X	↑	↓	List.ListParams (Character)
Analysis.ShowLockedCatalogValues (Con			X	↑		List.SubList.ListParams (Character)
Analysis.TraceMode (Combobox)						
Calendar.AbsencesAlwaysHaveEnd (Chec	Tab #2	X	↑	↓	Handling	
Calendar.ActivitiesAlwaysHaveEnd (Chec						
Calendar.AllDayMaxRows (Number)			X	↓	Document.OverwriteMode (Combobox)	
Calendar.DayEndsAt (Character)			X	↑	↓	Email.IconAction (Combobox)
Calendar.DayStartsAt (Character)			X	↑	↓	Email.Separator (Combobox)
Calendar.DefaultView (Combobox)			X	↑	↓	Search.AlwaysReadNumRecords (Number)
Calendar.FilterSets (Combobox)			X	↑	↓	Export.MaxRecords (Number)
Calendar.FirstWeekStartsAt (Combobox)			X	↑	↓	Record.DefaultAction (Character)
Calendar.FirstYearDatePicker (Number)			X	↑	↓	List.SubList.DefaultAction (Character)
Calendar.HolidayCalendarID (Combobox)			X	↑		
Calendar.LastYearDatePicker (Number)			X	↑		
Calendar.ShowAllReps (Checkbox)			X	↑		

UPDATE\_DEFAULT contains three configuration layouts (called using the `UserCon-figuration` action template):

- **User Configuration:** Contains parameters that are available for the standard user. See User Configuration in the Aurea CRM.Web User Interface documentation.
- **System Configuration:** Only the administrator should change these settings. See [System Configuration](#) on page 29.
- **Application Configuration:** Only the administrator should change these settings. See [Application Settings](#) on page 28.

You can edit the existing layouts such as add/remove/rearrange parameters or create new ones. See [Designing Configuration Units](#) on page 64.

## Web Configuration Parameters

The following chapters list all available Web Configuration parameters.

### Activity Parameters

This topic holds the Activity parameters.

#### Activity.ActivityType\_email

**Description:** The catalog value ID of an activity of type `email`.

**Default:** BTB = 5

**Type:** Checkbox

### **Activity.ActivityType\_Inf**

**Description:** The catalog value ID of an activity of type `Info`.

**Default:** BTB= 6

**Type:** Checkbox

### **Activity.CollisionCheck**

**Description:** Activates the collision check for appointments **Activities** (MA) and **Absences** (A3). The collision check works for Expand and List views.

If checked, the **Scheduling** dialog is automatically opened if the main rep already has another appointment during the same time.

To disable the collision check for certain appointment types, see **Activity.CollisionCheckExcludedActivityTypes**.

**Default:** Checked

**Type:** Checkbox

### **Activity.CollisionCheckExcludedActivityTypes**

**Description:** Determines which activity types do **not** trigger the collision check. The collision check functionality usually only makes sense for certain activity types like visits or events, but not for emails or faxes as an example, .

Enter a comma-separated list of catalog value IDs from the **Activity Type** catalog to be ignored by the collision check. Emails are excluded from the collision check.

**Default:** BTB = 5

**Type:** List

### **Activity.ImagePrefix**

**Description:** Defines the prefix used for registering activity type icons in the designer database.

**Defaults:**

**BTB:** CAT\_CONTACT\_BTBT\_, **FS:** CAT\_CONTACT\_FS\_

**OTC:** CAT\_CONTACT\_

**Type:** List

## **Analysis Parameters**

This topic holds the Activity parameters.

### **Analysis.CaseSensitiveCriteria**

**Description:** If checked, the input the user provides for filters is case-sensitive.

**Default:** Unchecked

---

**Type:** Checkbox

### **Analysis.CountFormatting**

**Description:** Defines the number of decimal places for Count values in analyses.

**Default:** #,0, i.e., no decimal places

**Type:** Character

### **Analysis.DateCategories**

**Description:** Defines which date segmentation options are available:

Separate multiple entries by a comma. Users see the categories in the order they are defined here.

For further information, see [Analysis Date Categories](#) on page 532.

**Default:** DateMonthly,DateQuarterly,DateYearly,\$

**Type:** List

### **Analysis.MaxRecords**

**Description:** Defines the maximum number of records read for the base query of an analysis. If the number of rows returned exceeds this value, a message is displayed.

Enter -1 for All.

**Default:** 1000

**Type:** Number

### **Analysis.EnableFusionChartsNumberFormat**

**Description:** Applies the Fusion Chart's number formatting to analysis charts instead of the standard CRM datamodel number formatting.

**Default:** Undefined

**Type:** CheckBox

### **Analysis.RegardEmptyValues**

**Description:** If you use a text field or a variable catalog as an analysis category and `Analysis.RegardEmptyValues` is checked, all values – including empty ones – are regarded by the analysis. To ignore empty catalog values, uncheck the parameter.

This parameter is not part of `UPDATE_DEFAULT` but needs to be created if required.

**Default:** Unchecked

**Type:** Checkbox

### Analysis.Rights

Defines the user's rights for loading, saving and executing analyses. Same options as for the `Query.Rights` parameter, see [Query.Rights](#).

If set to `disabled`, the **Analysis** page in Aurea CRM.Web is not available. However, analyses that are directly integrated into a menu can still be executed, e.g., the **Sales Funnel** in the BTB version.

**Default:** All

**Type:** Combobox

### Analysis.ShowLockedCatalogValues

**Description:** Use this parameter to define whether catalog values locked via the **Maintenance** module are displayed in analyses. The following options are available.

**Default:** Display Locked Catalog Values

**Type:** Combobox

### Analysis.ResultGrid.PageSize

**Description:** Defines the number of result rows displayed in the table grid view of an analysis result page. It is part of the `UPDATE_DEFAULT` configuration.

**Default:** 10

**Type:** Number

### Analysis.SkipDuplicateEntries

**Description:** Defines whether the duplicate entries are skipped or not. If set to `True`, the record id in the duplicate record is the same and hence skipped in the sum calculation. If set to `False`, the duplicate record fetched in AQL is maintained as a unique record by generating unique record id. This is resulting in counting each entry in the count (default behaviour of any analysis).

**Default:** False

**Type:** Checkbox

## Buying Center Parameter

This topic holds information about Buying Center parameter.

### BuyingCenter.MissingRoles.Y1.Definition

**Description:** This parameter specifies the path and name of a JSON file containing the definition of missing roles used by the Buying Center Wall.

Specify the path relative to the application path, e.g.:

```
data\settings\crm\BuyingCenter_MissingRoles_Y1.json.js
```

This parameter is undefined in `UPDATE_DEFAULT`, and no JSON file is delivered out-of-the-box since the **Participation** catalog determining the roles is a variable catalog and, therefore, different for every project.



---

For examples and a complete description of the required JSON structure and supported placeholders, see the article “How to Configure Missing Roles for Buying Center” at <https://support.aurea.com>.

**Default:** Undefined

**Type:** Character

## Calendar Parameters

Use the following parameters to configure Aurea CRM.Web's scheduling features as well as the display of calendar items such as Activities, Tasks, Absences, and To-Dos.

These settings can have a positive effect on the performance of your system, as they reduce the load on your server.

### Calendar.AbsencesAlwaysHaveEnd

**Description:** If set, only absences with a valid end date are displayed in the calendar. This improves calendar performance.

**Default:** False

**Type:** Checkbox

### Calendar.ActivitiesAlwaysHaveEnd

**Description:** If set, only activities with a valid end date are displayed in the calendar. This improves calendar performance.

**Default:** False

**Type:** Checkbox

### Calendar.AllDayMaxRows

**Description:** Defines how many all-day appointments, i.e., activity, absence, and task records are shown at the top of the calendar. If more all-day appointments exist, the list is truncated and `+<n> more` is displayed. This parameter applies to **Day**, **Work Week**, and **Week** views only.

**Default:** 5

**Type:** Number

### Calendar.DayEndsAt

**Description:** Defines the end time of the work day in the **Calendar**, **Dispatching Dashboard**, and **Resources** view.

**Default:** 18:00

**Type:** Character

### Calendar.DayStartsAt

**Description:** Defines the start time of the work day in the **Calendar**, **Dispatching Dashboard**, and **Resources** view.

**Default:** 08:00

**Type:** Character

### Calendar.DefaultView

Defines the **Calendar** default view: **Day**, **Work Week**, **Week**, and **Month**. This setting is overruled by the user's selection.

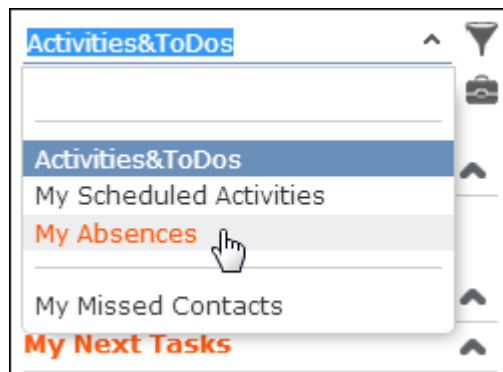
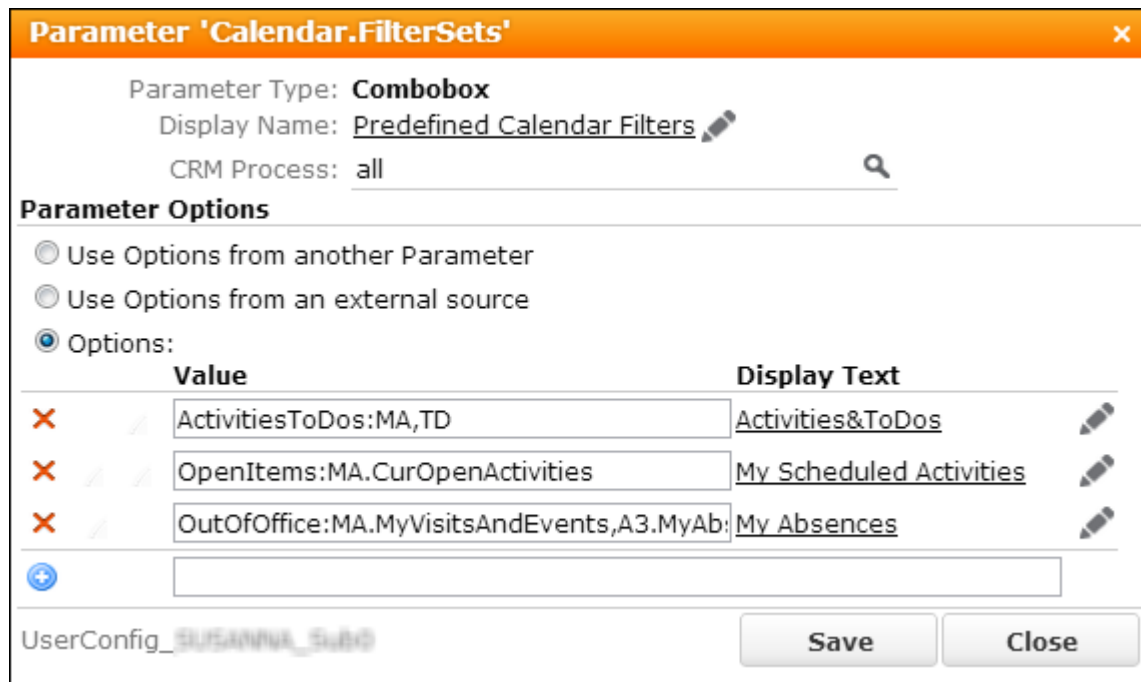
**Default:** No value

**Type:** Combobox

### Calendar.FilterSets

**Description:** Specifies options for filtering calendar items. Define each entry you want to offer in the calendar's **Filter** drop-down list as a parameter value. For details, see [Modifying Web Configuration Parameters](#).

These filters are listed in the calendar's **Filter** drop-down list above the user-defined filters.



The name displayed in the drop-down list is the language-dependent text entered in the **Display Text** column. See [Texts](#) on page 176 for further information.

### Syntax:

Define the values using the following syntax:

```
<Alias>:<InfoAreaId>[.<FilterName>],<InfoAreaId>[.<FilterName>],...
```

**Alias:** unique name to identify the filter option.

**InfoAreaId:** the ID of the concerned info area.

**Filtername (optional):** an actual filter's name as defined on the **Filter** page. If `Filtername` is omitted, all records of the specified info area are displayed. Use this functionality to filter calendar items per info area, e.g. specify `ContactsToDos:MA,TD` to only display **Activities** (MA) and **To-Dos** (TD).

### Examples:

- `All:MA,A1,A3,TD`

If the user selects this filter, all calendar items (i.e. records from the info areas MA, A1, A3 and TD) are displayed in the calendar.

- `ContactsToDos:MA,TD`

Only activities and to-dos are displayed.

- `OutOfOffice:MA.MyVisitsAndEvents,A3.MyAbsences`

Only activities of type visit and event, and absences are displayed.

### Note:

- Filter and rep sets created by users are saved in the Aurea CRM database.
- Filter sets defined as values of the `Calendar.FilterSets` parameter are saved in the CRM.Designer database.
- Filters are always saved in the CRM.Designer database. for details, see [Defining Filters](#) on page 489.

**Default:** Undefined

**Type:** Combobox

### Calendar.FirstWeekStartsAt

**Description:** Defines which day in January must be within of the first week of the year.

**Default:** 4th January

**Type:** Combobox

### Calendar.FirstYearDatePicker

**Description:** Defines the first year displayed in the date picker.

**Default:** 1900

**Type:** Number

### **Calendar.HolidayCalendarID**

**Description:** Defines the holidays that are highlighted in the calendar and date picker. To add more holiday calendars to the drop-down list, add the desired calendar IDs as new values. For details, see [Modifying Web Configuration Parameters](#).

The holiday calendars need to be configured in the **Configuration** info area (MC). For details on calendar definitions and a list of IDs, see **Configuration Info Area** in the [CRM.Core Administrator Guide](#) (**Calendar** category, **Holidays** option).

In addition, `Calendar.HolidayCalendarID` contains the value `Use calendar of rep.` If this value is selected, the calendar defined in the current user's **Rep** record (ID) is used.

For changes to take effect, users need to log on again.

**Default:** International

**Type:** Combobox

### **Calendar.LastYearDatePicker**

**Description:** Defines the last year displayed in the date picker.

**Default:** Undefined.

**Type:** Number

### **Calendar.ShowAllReps**

**Description:** If checked, users may select displaying the calendar items for all reps at once using an extra **All** checkbox in the **Reps and Rep Groups** dialog of the calendar.

**Default:** Unchecked

**Type:** Checkbox

### **Calendar.SidebarItems**

**Description:** Determines which upcoming items are displayed in the calendar sidebar: **My Next Appointments**, **My To-Dos**, **My Next Tasks**, **My Absences**. Check or uncheck the checkboxes to show or hide these areas from the calendar sidebar.

**Default:** All checked

**Type:** Checkboxes

### **Calendar.SidebarMaxRecords**

**Description:** Determines the number of upcoming items such as activities, to-dos, etc., to be listed in the Calendar sidebar.

**Default:** 5

**Type:** Number

**Calendar.TasksAlwaysHaveEnd**

**Description:** If set, only tasks with a valid end date are displayed in the calendar. This improves calendar performance.

**Default:** False

**Type:** Checkboxes

**Calendar.TodosAlwaysHaveEnd**

**Description:** If set, only to-dos with a valid end date are displayed in the calendar. This improves calendar performance.

**Default:** False

**Type:** Checkboxes

**Calendar.WeekSelectedDays**

**Description:** The number of days that are displayed in the calendar if the user clicks on a week number in the date picker. This setting is taken into account by the **Calendar** and **Daily & Weekly Planning** pages.

**Default:** 5

**Type:** Number

**Calendar.WeekStartsAt**

**Description:** Determines the weekday with which the week starts. This setting applies to both the 5- and 7-day view in the calendar, the mini calendar and the date picker start.

The values `Sunday` and `Monday` are defined in `UPDATE_DEFAULT`. To define other weekdays, add more values to the `Calendar.WeekStartsAt` definition. For details, see [Modifying Web Configuration Parameters](#). Valid values are numbers between zero and six as showing in the table below:

**Table 1: Weekdays numbering**

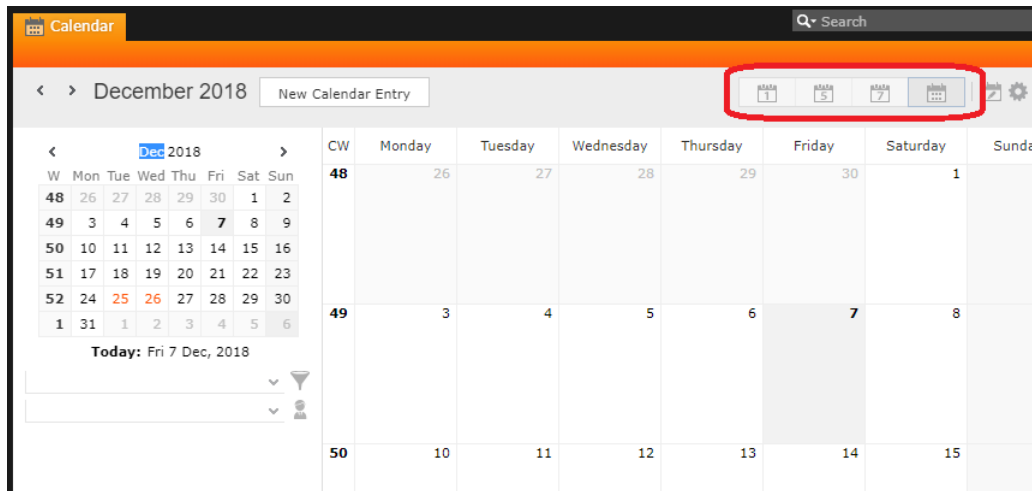
Day	Number
Sunday	0
Monday	1
Tuesday	2
Wednesday	3
Thursday	4
Friday	5
Saturday	6

**Default:** Monday

**Type:** Combobox

## Calendar.RememberLastView

**Description:** If enabled, it opens calendars using the last view selected by the user.



**Default:** Unchecked

**Type:** Checkbox

## Change Notification Parameters

This topic holds the Change Notification parameters.

### Email.ChangeNotification.From

**Description:** Specify a valid company email address to send the email change notifications.

**Default:** No value

**Type:** String

### Email.ChangeNotification.Body

**Description:** Defines the format of the body of the email change notification. You can modify the default value, specified below.

```
<p>As requested, you are receiving a notification for a change done to the record {0}. </p>
```

```
<p>To open the record in Aurea CRM please click on the link below.</p><br />
```

```
<a href='{1}' title='{2}'>{2}<a/>
```

---

**Note:** You can modify the text content, but all the format parameters in curly braces should be retained. The parameter {0} fetches the record name.

---

**Default:** Undefined

**Type:** String

---

## Email.ChangeNotification.Subject

**Description:** Specify a subject format for the email change notifications. The default value is specified below.

```
<h1>You have a new notification for {0}</h1>
```

---

**Note:** You can modify the text content, but all the format parameters in curly braces should be retained. The parameter {0} fetches the record name.

---

**Default:** Undefined

**Type:** String

## Email.ChangeNotification.MaximumEmailSendAttempts

**Description:** Specify the number of resend attempts for the email change notification, in the case of consecutive mail delivery failures.

**Default:** 3

**Type:** Number

## Watchlist.AvailableInfoAreas

**Description:** Contains the comma-separated list of inforareas that are enabled for watchlist subscriptions. This list is made available in the combobox in the search pages for watchlist subscriptions and change notifications.

**Default:** FI, KP, CP, A1, AK, AP

**Type:** List

## Watchlist.RecentNotifications.PeriodInDays

**Description:** Specifies the period in days to display change notifications by the new change notifications widget in the dashboard.

**Default:** 30

**Type:** Number

## Database Parameters

This topic holds the Database parameters.

### Database.AssociationMode.Default

**Description:** Determines the initial state of the global switch for the Affiliation View. For details, see [Affiliation View](#) on page 752.

Available values:

- **IncludeConfigured:** The **Affiliation View** checkbox in `$System` is checked. Child records of the configured info areas are displayed for all affiliates.
- **Exclude:** The **Affiliation View** checkbox is unchecked.

The user's selection overrules this initial default setting.

**Default:** Undefined

**Type:** Character

### **Database.HintModeList**

**Description:** Globally determines the behavior of database hints for lists.

**Default:** Undefined

**Type:** Combobox

### **Database.HintModeQuery**

**Description:** Globally determines the behavior of database hints for queries.

**Default:** No value

**Type:** Combobox

### **Database.SortCollation**

**Description:** Determines the sort collation to be used for the database, e.g., `SQL_Latin1_General_CP1_CI_AS`, `German_phonebook_CI_AS`, etc.

**Default:** Undefined

**Type:** String

## **Designer Configuration Parameters**

This topic holds information about Configuration parameters.

### **DesignerConfiguration.AccessPublicParentConfigurations**

**Description:** If checked, users can select analyses, queries and filters from parent configurations of `DesignerConfiguration.PublicSaveConfiguration` as well.

**Default:** Unchecked

**Type:** Checkbox

### **DesignerConfiguration.PublicSaveConfiguration**

**Description:** Defines to which configuration public queries and analyses are saved. Set this value for users allowing them to save public analyses and public queries if the `Query.SaveTo` option is set to `CRM.Designer`. See [Query.SaveTo](#) for more information.

If undefined, public queries and analyses are stored in the highest writable configuration within the user's configuration hierarchy.

---

**Note:** This integer value is downloaded and uploaded as it is. You may have to adjust this value each time you import your configuration from XML files containing data from another CRM.Designer database.

---

**Default:** Undefined



**Type:** Combo: WriteConfig

## Document Parameters

This topic holds the Document parameters.

### Document.AutoUpload

**Description:** If a user opens a document via **Open Document** or **Check Out & Edit**, for MS Office documents, he is asked upon closing the modified document if it should be uploaded to the Aurea CRM database. `Document.AutoUpload` defines whether users should also be prompted for automatically uploading a non-MS Office document like PDF, TXT, etc., after editing. When the option is checked the user is prompt for automatic upload. Otherwise, the user needs to upload manually such documents.

**Default:** Checked

**Type:** Checkbox

### Document.DefaultContentDisposition

**Description:** Defines how a document is displayed on the page. There is two option available:

Value	Result
Attachment	The document is not opened in the browser but with the associated program.
Inline	The document is opened in the browser.

See the next parameter for the possible combinations.

**Default:** Attachment

**Type:** Combobox

### Document.TargetForOpening

**Description:** Defines in which window the document appears. There are two options available:

Value	Result
CurrentWindow	The document is displayed inside Aurea CRM.Web.
NewWindow	A new window is opened to display the document.

**Default:** CurrentWindow

**Type:** Combobox

Using this and the previous parameters the following combinations are possible:

		<b>Document.DefaultContentDisposition</b>	
		<b>Attachment</b>	<b>Inline</b>
<b>Document.TargetForOpening</b>	<b>CurrentWindow</b>	Regardless of the settings of <code>Document.TargetForOpening</code> , the browser has its default download-behavior (standard "Do you want to open/save this file" dialog).	If the browser can handle the file, Aurea CRM.Web navigates the workframe to the downloaded file and displays it. If not, the same behavior as with attachment is used.
	<b>NewWindow</b>		If the browser can handle the file, a new window is opened, displaying the file. If not, the same behavior as with attachment is used.

### **Document.MailMergeCache**

**Description:** Check this parameter to set mail merge documents to be cached. Uncheck the option to disable cache.

For more information, see [Document Caching & Performance Considerations](#) on page 852.

**Default:** Checked

**Type:** Checkbox

### **Document.OverwriteMode**

**Description:** Defines the action taken when a user tries to overwrite an existing document. Available values:

- Never overwrite existing documents
- Prompt before overwriting documents
- Always overwrite existing documents

**Default:** Never overwrite existing documents

**Type:** Combobox

### **Document.UploadFieldGroup.D1/D2**

**Description:** Use the parameters `Document.UploadFieldGroup.D1` and `Document.UploadFieldGroup.D2` to specify the field group used for the **Upload Documents** dialog. If undefined, the default field groups `D1.Upload` and `D2.Upload` are used. For details, see [Special Field Groups Naming Conventions](#) on page 190.

**Default:** Undefined

---

**Type:** Character

## Email Parameters

This topic holds information about Email parameters.

### Email.AttachmentsExcludedContentTypes

**Description:** Allows you to limit the email attachments to certain content types, for the **Send as Email** and **Notify Rep** functionality. For details, see [Notify Rep](#) on page 837 and [Send As email](#) on page 835.

For example, enter `application/pdf,text/xml` as parameter values, to exclude PDF and XML documents from the attachments list in the **Notify Rep** and **Send As Email** dialog.

**Default:** Undefined

**Type:** Character

### Email.AttachmentsExcludedDocumentClasses

**Description:** Allows you to limit the attachments offered for **Notify Rep** to certain document classes. For details, see [Notify Rep](#) on page 837.

Enter `Spam, Internal` to exclude documents classified as spam or internal, via the **Document Class** field in D1 from the attachments list in the **Notify Rep** dialog.

**Default:** Undefined

**Type:** Character

### Email.AttachmentsMaxSizeInKB

**Description:** Determines the allowed maximum size for email attachments, measured in KB, when sending server-side emails.

**Default:** No value

**Type:** Number

### Email.AutoSendOnNew

**Description:** Defines whether an email is sent immediately from the server after adding a new activity, or whether the user's email client opens to allow the operation to proceed.

**Default:** Undefined

**Type:** Checkbox

### Email.BodyFieldID

**Description:** By default, the **Text** field of an activity record (MA) is used as the email body when sending an email from Aurea CRM.Web. To use a different field from MA, specify its XML name or unique field ID in `Email.BodyFieldID`.

**Default:** No value

**Type:** Character

**Email.ClientSide.MailEncoding**

**Description:** Use this parameter to set a different encoding system when sending emails with special characters.

Examples for valid code pages are:

932	Japanese (Shift-JIS)	1253	Greek (Windows)
1250	Central European (Windows)	1254	Turkish (Windows)
1251	Cyrillic (Windows)	50221	Japanese (JIS-Allow 1 byte Kana)
1252	Western European (Windows)	50225	Korean (ISO)

**Default:** 1252

**Type:** Character

**Email.ClientSide.ProtocolMaxLength**

**Description:** Determines the maximum length of messages when sending client-side emails using the `mailto:` protocol. For further details, see [EMail.SendingMethod](#). If an email message exceeds the specified value, the message is truncated and a notification is displayed.

Set this parameter to the value appropriate for your default email program.

If the email contains the current record as a link, as when using the **Notify Rep, Insert Aurea CRM.Web link** option, the link takes precedence over the email text. First, the text is truncated; if the `mailto-string` still exceeds the defined maximum length, the link is truncated as well.

**Default:** <= 0, 200

**Type:** Number

**Email.EnableWebLink**

**Description:** If checked, users can choose to insert a link to Aurea CRM.Web when using the **Notify Rep** function for server-side emails.

If emails are sent from the client, the link is inserted automatically.

**Default:** Checked

**Type:** Checkbox

**Email.NotifyRep.WebLinkInitialState**

**Description:** If checked, the **Insert Aurea CRM.Web link** checkbox is checked by default in the **Notify Rep** dialog when a user opens the dialog for the first time. If the user decides not to insert the web link and deselects it, then that setting is used for subsequent notifications.

**Default:** Unchecked

**Type:** Checkbox

**Email.EnableWinLink**

**Description:** If checked, users can choose to insert a link to Aurea CRM win when using the Notify Rep function with server-side emails. If emails are sent from the client side, the link is inserted automatically.

**Default:** Unchecked

**Type:** Checkbox

**Email.ExcludeCurrentUser**

**Description:** Controls whether the current user is added to the internal recipients' list when sending emails.

**Default:** Unchecked

**Type:** Checkbox

**Email.FallbackToCompany**

**Description:** If checked, the **Company** address is used for **Person** recipients that have no email address. Otherwise, there is no fallback and persons without an email address do not get an email.

**Default:** Unchecked

**Type:** Checkbox


**Email.From**

**Description:** Defines whether the sender **From:** of an email is the currently logged-in user or the rep entered in the **Rep ID** field of the activity record. Using the record's rep as the email sender allows emails to be sent **on behalf of**. Using the current user prevents it.

**Default:** Record's Rep

**Type:** Combobox

## Email.IconAction

**Description:** Defines the action taken when a user clicks on an email address on view mode or the **Send email** icon  edit mode of an email field in a **Company** or **Person** record. The following values are available:

- Open email
- Add activity
- Ask user

**Default:** Ask user

**Type:** Combobox

## Email.NotifyRep.BodyFieldIDs

**Description:** Determines the fields to be used as email body text by the **Notify Rep** functionality. For more details, see [Notify Rep](#) on page 837.

By default, only a field with the XML name `Text` is used. Use this parameter to specify other text fields.

### Syntax

```
<InfoAreaId>:<fieldId>,<InfoAreaId>:<fieldId>,...
```

You can specify either unique field IDs or XML field names as `<fieldId>`.

### Example

```
A1:7001,PR:7023,FI:7005
```

The field specified here overrides a field with the XML name `Text`. One field per info area is supported. If more than one field is specified for an info area, the first field is used.

**Default:** Undefined

**Type:** Character

## Email.SendCopyToRep

**Description:** When an activity record is sent by email, whether and how the current rep is included in the email's recipients is defined by this parameter.

**Default:** CC

**Type:** Combobox

## Email.SendDeliveryNotification

**Description:** If checked, a delivery receipt is requested when sending server-side emails. For further details, see <http://www.ietf.org/rfc/rfc3461.txt>. If this parameter is checked, the **Request a delivery receipt** check box in the **Notify Rep** dialog is initially checked.

**Default:** Unchecked

**Type:** Checkbox

## Email.SendingMethod

**Description:** Select whether emails are sent from the server or the client.

If emails are sent from the server, attachments linked to the activity record are sent as well, and Unicode characters are transferred correctly. However, the email is sent with the exact text of the activity record, and no editing is allowed. Furthermore, is not displayed in the user's outbox.

If emails are sent from the client, documents linked to the activity record are not automatically sent as attachments.

### Notes for sending client-side emails:

- The number of characters that can be exported from the Text field of an activity record in Aurea CRM.Web to your mail client is limited by the operating system and the mail client software. If the limit is exceeded, Aurea CRM web displays a notification and the text is truncated with "...". Use the `Email.ClientSide.ProtocolMaxLength` Web Configuration parameter to configure the maximum length of email texts.
- If you want to send texts with unlimited length, you have to send server-side emails.
- Special characters in email addresses are supported only with Microsoft Outlook 2007 or higher and with Lotus Notes 7 or higher.
- The `mailto:` protocol does not support HTML-formatted text. For details, see <http://tools.ietf.org/html/rfc2368>. Therefore you cannot define an HTML field in `Email.BodyFieldID` when sending client-side e-mails. Specify the linked plain text field instead, see [HTML Fields](#) on page 219.

### Note for sending server-side emails:

You need to define a valid mail server in the Mail Server section of the settings.xml file to be able to send server-side emails. For further details, see **Mail Server** in the [Aurea CRM.Web Installation Guide](#).

**Default:** Server

**Type:** Combobox

## Email.SendReadNotification

**Description:** If checked, a read receipt is requested when sending server-side emails. For details, see <http://www.rfc-editor.org/rfc/rfc3798.txt>. If this parameter is checked, the **Request a read receipt** check box in the **Notify Rep** dialog is initially checked.

**Default:** Unchecked

**Type:** Checkbox

## Email.SendRecordNotification

**Description:** If checked, the Email Notification should include external recipients.

**Default:** Unchecked

**Type:** Checkbox

### **Email.SendWithoutConfirmation**

**Description:** If checked, the email is sent without any further user input, i.e., no confirmation dialog is displayed.

This parameter applies to server-side emails only.

**Default:** Unchecked

**Type:** Checkbox

### **Email.Separator**

**Description:** Defines the separators used in the **To:**, **CC:**, **BCC:** fields of email programs, e.g., semicolon in Outlook and comma in Lotus Notes. The same separator is used in the **Participant** fields in the **Activity** info area.

**Default:** Semicolon

**Type:** Checkbox

## **Error Handling Parameter**

This topic holds the Error Handling parameter.

### **ErrorHandling.Textgroups**

**Description:** Comma-separated list of text groups that contain the language-specific error messages used by Aurea CRM.Web.

**Default:** net\_error\_texts, net\_component\_error\_texts, net\_customfield\_texts

**Type:** List

## **Expand Parameters**

This topic holds the Expand parameters.

### **Expand.AddressCheck.InfoAreas**

**Description:** To determine for which info areas the external address should be performed, provide a comma-separated list of info area IDs, e.g., FI, KP, see [External Address Check](#) on page 840.

**Default:** Undefined

**Type:** List

### **Expand.FieldHook**

**Description:** If you enter a field hook in this field, the hook function is executed each time a field is changed in any **Expand** view, except for fields that have an explicit hook function defined in the field attributes. For details, see [Hook Function](#) on page 238.

**Default:** Undefined

**Type:** Character



### Expand.InitialValuesFromLastInput

**Description:** This parameter defines the behavior when creating new records: Depending on this setting, some or all values from the last data entry, i.e., the previously added new record are used for the `ExpandNew` action automatically. Available values:

- Do not transfer the last input
- Use all values from last entry
- Use Admin-defined values from last entry

**Default:** Do not transfer the last input

**Type:** Combobox

### Expand.PanelReorderingAllowed

**Description:** If checked, users can rearrange the panels on in an **Expand** view using drag & drop.

**Default:** Checked

**Type:** Checkbox

### Expand.SignatureParsingModes

**Description:** Defines the available parsing modes for the signature parser. For details, see [Creating/Updating Records From email Signatures/Websites](#) on page 758.

Add the modes `o.modus` as values, see [Modifying Web Configuration Parameters](#). Set a mode as the initial default.

**Default:** Undefined

**Type:** Combobox

### ExpandViewExtrasWidget. DoNotHideIfEmpty

**Description:** Check this parameter to disable auto-hiding of records displayed in the expanded **Extras Widget** view.

**Default:** Checked

**Type:** Checkbox

### Expand.RespectFieldAlignment

**Description:** Select this parameter to ensure that any alignment format specified on numeric fields in the data model are rendered accordingly when the field values are displayed in CRM.Web.

**Default:** Unchecked

**Type:** Checkbox

## Expand.ValidationSummaryPosition

**Description:** Use this parameter to customize the positioning of a validation message for all Expand actions. For example, saving a record results in a validation message about the save action. You can select one of three positions: **Top**; **Bottom**; **Hidden**. This value is applied to all Expand actions globally.

---

**Note:** To customize the positioning of a specific action, use the Expand action parameter. For details, see [validationSummaryPosition](#).

---

**Default:** Top

**Type:** Combobox

## Expand.ExpandHeaderPosition

**Description:** Use this parameter to customize the positioning of the Expand header. You can select one of two options: **Top** and **Bottom**.

**Default:** Top

**Type:** Combobox

## Export Parameters

This topic holds information about Export parameters.

### Export.CSV.Separator

**Description:** Defines the separator for CSV files.

**Default:** Tab

**Type:** Combobox

### Export.EmitRepNames

**Description:** By default, when a list or query contains rep ID fields with field type `uid`, these fields are transformed to include the rep name as string when displaying and exporting the list or query result. The same logic applies to participant fields.

If you want to export the IDs instead, set `Export.EmitRepNames` to `false`.

You can specify this option per report by using the `Export.EmitRepNames` export option, see [Export Options](#) on page 858.

**Default:** Undefined

**Type:** Character

### Export.MaxRecords

**Description:** Defines the maximum number of records to export. Enter `-1` to set the value to **All**.

**Default:** 1000

**Type:** Number

## Export.QueryEncoding

**Description:** Defines the character encoding for exporting queries, e.g., `Export.Encoding=UTF-16`. This setting is used as a fallback for reports where there is no character encoding specified in the action.

**Default:** Undefined

**Type:** Character

## Export.ShowCombinedArrayFields

**Description:** If a query field is of the type Array, then on exporting the query result, all the fields in the Array are exported. Set this parameter to false to prevent all the fields from getting exported.

For example, on exporting the result of the query `select (CoType, Company, Country) from (FI)`, all the fields of the array [`Co.Type`, `Co.Type2`, `Co.Type3`, `Co.Type4`, `Co.Type5`] are exported. To prevent this, set this parameter to false, in which case only `Co.Type` is exported.

**Default:** True

**Type:** Boolean

## Export.SingleListRecord

**Description:** If only one record is selected in a list and the user clicks on an export button, all currently visible records are exported. For further information, see [Export/Report Buttons](#) on page 310.

Check `Export.SingleListRecord` to offer the possibility to export only the selected record.

**Default:** Unchecked

**Type:** Checkbox

## Export.Type

**Description:** Defines the type for data exports used for the Export buttons `Direct Export` and `Direct Export All`. The options available are:

- CSV
- XML

For further information, see [Export/Report Buttons](#) on page 310.

**Default:** CSV

**Type:** Combobox

## Export.WinReportAlwaysInMemory

**Description:** If checked, the Aurea CRM win report is created in-memory, instead of as a file. For details, see [Creating Aurea CRM win Reports in Aurea CRM.Web](#) on page 863.

Use `Export.WinReportMemoryThreshold` to specify a maximum file size used for in-memory creation. For further information, see the next parameter.

**Default:** Unchecked

**Type:** Checkbox

### Export.WinReportMemoryThreshold

**Description:** Defines the maximum file size used for in-memory creation of Aurea CRM.Win reports. Value in KB.

If `Export.WinReportAlwaysInMemory` is checked and the defined `Export.WinReportMemoryThreshold` is exceeded; the report creation is aborted with an error message.

**Default:** 256

**Type:** Number

### Export.Xml.EmitConfiguration

**Description:** If checked, additional information like field types and labels is included in the exported XML file.

```

...
    <Configuration UnitType="update.Configuration.Contracts.FieldControl"
ControlName="Details" UnitId="5442" InfoAreaId="FI">
    <Tabs>
    ...
    <Tab TabIndex="4">
        <Caption>Classification</Caption>
        <Fields>
            <Field Index="35" InfoAreaId="FI" TabNr="4">
                <FieldAttributes>
                    <FieldAttribute Id="35" Edit-
Mode="15">CAT_ABC_{javascript:( $('[VALUE'] || '' ).toUpperCase();}</FieldAt-
tribute>
                    </FieldAttributes>
                </Field>
            ...

```

You can specify this option per report by using the `Export.Xml.EmitConfiguration` export option. For further information, see [Export Options](#) on page 858.

**Default:** Undefined

**Type:** Checkbox

### Export.Xml.EmitLineBreaks

**Description:** By default, if a field contains line breaks, these line breaks are exported.

If you want to export data that is to be processed by some third-party system that does not support line breaks, uncheck `Export.Xml.EmitLineBreaks`.

You can specify this option per report by using the `EmitLineBreaks` export option. For details, see [Export Options](#) on page 858.

**Default:** Checked

**Type:** Checkbox

---

## Export.Xml.ExportPhysicalInfoAreaIds

**Description:** By default, exported XML contains the actual info area ID and XML name of virtual info areas, see [XML Export of Virtual Info Areas](#) on page 280.

To export the physical info area ID and XML name instead, create and check the Web Configuration parameter `Export.Xml.ExportPhysicalInfoAreaIds`.

You can specify this option per report by using the `Export.Xml.ExportPhysicalInfoAreaIds` export option. See [Export Options](#) on page 858 for details.

**Default:** Undefined

**Type:** Checkbox

## Reporting.Pdf.EnableAsianFont

Set this parameter to `true` to enable the use of Asian fonts to generate PDF reports for export. This allows the use of Chinese, Japanese, and Korean alphabets. Aurea CRM.Web uses Verdana fonts to generate PDF reports.

**Default:** False

**Type:** Checkbox

## Favorites Parameter

This topic holds the Favorites parameter.

### Favorites.InfoAreas

**Description:** Determines for which info areas users can create favorites.

**Defaults:**

**BTB:** FI,KP,PE,OJ,Y1,CM

**FS/OTC:** FI,KP,PE,Y1,CM

---

**Note:** Only records from info areas whose key consists of station number plus serial number can be added as favorites.

To enable the user to add a favorite for these info areas, you also need to add the menu action `D_Favorites` to the respective context menu. For details, see [Favorites](#) on page 842.

---

**Type:** List

## Geolocation Parameter

This topic holds the Geolocation parameter

### Geolocation.FindAddressStrategy

**Description:** Determines the fallback behavior for displaying Bing Maps for maps in child records, e.g., MA. For details, see [Displaying Bing Maps in Related Areas and Forms](#) on page 826. The following values are available:

- Returns the first record which has address fields
- Returns the first record with address fields that are not empty

---

**Note:** A change to this parameter only takes effect with the next login.

---

**Default:** Undefined

**Type:** Combobox

## Google Analytics Parameters

This topic holds information about Google Analytics Parameters.

For detailed descriptions and purpose of the Google Analytics web configuration parameters, see [Configuring the Google Analytics Tracker](#) on page 660.

### GoogleAnalytics.CRMInstanceID

**Description:** Specify a unique string to identify unique members or departments in your organization. The parameter is then tagged to the custom reports generated.

**Default:** Undefined

**Type:** Catalog

### GoogleAnalytics.CustomDimension.Infoarea

**Description:** Specify a value in the format `<Dimension><index number>`. The index number is the index value of the Infoarea dimension configured in the Google Analytics tracker. See [Configuring Custom Dimensions in the CRM Designer](#). For example, an index number of 1, has a value of `dimension1`.

**Default:** Undefined

**Type:** Catalog

---

**Note:** This parameter is not part of `UPDATE_DEFAULT` and needs to be created if you want to use it.

---

### GoogleAnalytics.CustomDimension.CRMVersion

**Description:** Specify a value in the format `<Dimension><index number>`. The index number is the index value of the CRM Version dimension configured in the Google Analytics tracker. For example, an index number of 3, has a value of `dimension3`.

---

**Default:** Undefined

**Type:** Catalog

### GoogleAnalytics.CustomDimension.InstanceID

**Description:** Specify a value in the format `<Dimension><index number>`. The index number is the index value of the InstanceID dimension configured in the Google Analytics tracker. For example, an index number of 4, has a value of `dimension4`.

**Default:**Undefined

**Type:** Catalog

### GoogleAnalytics.CustomDimension.UserID

**Description:** Specify a value in the format `<Dimension><index number>`. The index number is the index value of the UserID dimension configured in the Google Analytics tracker. For example, an index number of 5, has a value of `dimension5`.

**Default:**Undefined

**Type:** Catalog

### GoogleAnalytics.CustomDimension.View

**Description:** Specify a value in the format `<Dimension><index number>`. The index number is the index value of the View dimension configured in the Google Analytics tracker. For example, an index number of 2, has a value of `dimension2`.

**Default:** Undefined

**Type:** Catalog

### GoogleAnalytics.CustomerTrackerID

**Description:** Specify the tracker ID generated by Google Analytics for the tracker property configured for your instance of Aurea CRM. See [Configuring the Google Analytics Tracker](#) on page 660.

**Default:** Undefined

**Type:** Catalog

### GoogleAnalytics.MainTrackerEnabled

**Description:** You can use this parameter as follow:

- Set the value to **true**, if you want your CRM.Web instance to be tracked by the default Google Analytics tracker provided by Aurea.
- Set this value to **false**, if you have an enterprise license for Aurea CRM.Web and want to set up your custom Google Analytics tracker.

---

**Note:** You can only use the default Google Analytics tracker provided by Aurea, if you have a standard license.

---

**Default:** Unchecked

**Type:** Checkbox

### GoogleAnalytics.SendUserID

**Description:** Set this value to **true** if you want to send the user name with tracker data.

**Default:** Unchecked

**Type:** Checkbox

## Header Parameters

This topic holds information about Header Parameter.

### Header.KeepGroupTopElement

Disables the storing of the last selected button in a Button Group, >> [Grouping Buttons](#) on page 328. The first button of the group is always displayed.

This parameter is not part of UPDATE\_DEFAULT but needs to be created as Character if required. Set the value to "true" or "1" to activate this option.

**Type:** Character

### Widget.Header.View.Options

This parameter is used to configure the Header's default view options.

```
Widget.Header.View.Options = {  
  "showPaging": true  
}
```

```
// default value:  
Widget.Header.View.Options = {  
  "showPaging": "if-required"  
}
```

**Type:** JSON

## History Parameters

Use the following parameters to customize Aurea CRM.Web's history feature.

### History.Contents

**Description:** Defines which type of information is displayed in Aurea CRM.Web's browsing history. The options are:

- All
- Only record-related
- Only page-related

**Default:** All

**Type:** Combobox



### History.MaxPopupEntries

**Description:** Specifies the maximum number of historical entries per popup menu item. You can use this in combination with the `History.MaxEntries` parameter to get better control on the historical records that is maintained for a CRM.Web application running in the browser.

---

**Note:** `History.MaxPopupEntries` parameter is recommended to be smaller than `History.MaxEntries`.

---

**Default:**

**Type:** Number

### History.MaxEntries

**Description:** Defines the maximum number of entries in Aurea CRM.Web's browsing history.

**Default:** 10

**Type:** Number

### History.Persist

**Description:** If checked, Aurea CRM.Web's browsing history is saved between sessions. The history data is saved in the user's roaming settings.

---

**Note:** History is only saved if a session is ended by logging out.

---

**Default:** Unchecked

**Type:** Checkbox

### History.RecordHistory.MaxItems

**Description:** Use this parameter to control the number of items maintained in a record's history.

**Default:** 10

**Type:** Number

## Import Parameter

This topic holds information about Import parameter.

### Import.MaximumFileSize

Defines the maximum allowed size for import files in MB. See [Import](#) on page 43 for details.

You need to configure the `web.config` to allow a file size larger than 4 MB for the parameter setting to take effect. To set the file size in the `web.config` file:

```
<httpRuntime requestValidationMode="4.5" maxRequestLength="10240" target-  
Framework="4.5" />  
  <security>  
    <requestFiltering>  
      <!-- limit the maximum request size to 10MB -->  
      <requestLimits maxAllowedContentLength="10485760" />  
    </requestFiltering>  
  </security>
```

ASP.NET defaults are:

- `maxRequestLength`: **4096KB (4 MB)**
- `maxAllowedContentLength`: **30000000 bytes (30 MB)**

For more information, see:

- [http://msdn.microsoft.com/en-us/library/e1f13641\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/e1f13641(v=vs.85).aspx)
- [http://msdn.microsoft.com/en-us/library/ms689462\(v=vs.90\).aspx](http://msdn.microsoft.com/en-us/library/ms689462(v=vs.90).aspx)

**Default:** 4

**Type:** Number

## Jobs Parameters

This topic holds information about Jobs parameters.

### Jobs.PollingInterval

**Description:** Defines the polling interval, in minutes, for new **To-Dos**, **Messages**, and **Reminders**. The time stamp of the last check is saved in the user's roaming settings. If a user logs on for the very first time, the login date/time is used instead.

You can specify `-1` to completely turn off polling, or `0` to check for new records only once after the user logs in.

**Default:** 30

**Type:** Number

### Jobs.SessionCheckInterval

**Description:** You can define this parameter to periodically check on a job session and prevent a job session from timing out. The value is provided in seconds.

**Default:** 5

**Type:** Number

---

## Language Parameters

This topic holds information about the Language parameters.

### Languages.Editable/Languages.Visible

**Description:** These parameters determine which languages are visible and editable in Aurea CRM.Web. The settings apply to the following areas:

- [Multi-Language Fields](#) on page 221
- Languages in the catalog maintenance, see [Catalog Maintenance](#) on page 33:
  - `Languages.Visible` determines which language checkboxes are displayed on the search form.
  - `Languages.Editable` determines for which languages the user can update Text and External Key.
- Editing field names. See [Data Model](#) on page 57 for further information.

Specify the desired languages, in the format **Language No.** from the **Language (00)** info area, see [Language](#) on page 52. Enter 0 to configure language-independent. Enter -1 to get all defined languages.

Defaults if undefined:

`Languages.Editable`: Only texts in the current user's login language are editable.

`Languages.Visible`: All available texts are visible.

---

**Note:** `Languages.Editable` and `Languages.Visible` do not apply to the SU (Super User), i.e., the SU can view and edit all languages.

---

**Default:** Undefined

**Type:** Character

## List Parameters

This topic holds information about List parameters.

### List.AutoOpenMiniDetails

**Description:** If checked, the list header offers an icon allowing users to open the **MiniDetails** section of all search result records with one click.

**Default:** Checked

**Type:** Checkbox

### List.ListParams/List.SubList.ListParams

**Description:** These parameters allow the global definition of list and sub-list parameters. The settings can be overridden for individual sub-lists and lists.

For information on the available list and sub-list parameters, see [ListParams](#) on page 374.

### Syntax:

<Name>,<Value>;<Name2>,<Value2>; etc.

### Example:

```
canEdit, false; cachedMode, true;
```

**Default:** Undefined

**Type:** Character

### List.SubList.ActivationHotKey

**Description:** Defines the hotkey for activating the Related Data area.

**Default:** Alt+0

**Type:** Character

### List.SubList.DefaultAction

**Description:** The parameter defines the default action that is called when a user double-clicks an item in a sub-list. For more details, see [Default Action](#) on page 289.

**Default:** A\_ShowDefaultAction

**Type:** Character

### List.SubList.DefaultMenu

**Description:** Defines the default context menu used in sub-lists. The default value displays the info area default menu for each sub-list.

**Default:** %%

**Type:** Character

### List.TruncateTextsInList

**Description:** If set to `true`, the texts in list headers and list rows are truncated, and an ellipsis (...) is added to indicate the omission of text. Tooltips display the complete text with an icon to copy the text. If set to `false` the text is wrapped. This parameter is part of `UPDATE_DEFAULT`.

**Default:** False

**Type:** Checkbox

### Catalog.ArrayMultipleSelector

**Description:** The parameter allows the selection of an item in a catalog list of items in a catalog array multiple times. If this property is set to `false`, the item is removed from the list of items displayed. The default value is `true`. This parameter is part of `UPDATE_DEFAULT`.

**Default:** False

**Type:** Checkbox

## HideNotUsedFields

**Description:** Setting this parameter to true hides the fields not used in the **Modify List Fields** and **Query definitions**.

**Default:** False

**Type:** Checkbox

## Locale Parameters

Use the following parameters to configure number, date, and time formats used in Aurea CRM.Web.

The server's locale settings are used as defaults.

---

**Note: Type Checking:** Aurea CRM.Web prevents users from entering date, time or numbers in an incorrect format. The following user inputs are possible when editing fields in lists and masks:

---

- **Numeric Fields (type short and float):** numbers 0-9 as well as % and -, plus the defined `Locale.Number.DecimalSeparator` and `Locale.Number.ThousandsSeparator`.
  - **Date Fields:** numbers 0-9 plus the defined `Locale.Date.Separator`.
  - **Time Fields:** numbers 0-9 plus the defined `Locale.Time.Separator`. If `Locale.Time.DisplayFormat` is set to 12-hour format, a, p, m, and space are valid as well.
- 

**Note:** No logical checks are performed in Aurea CRM.Web, i.e., if a user enters e.g., February 31st as a date, he does not get a message upon saving the record.

---

## Locale.BaseCurrency

**Description:** Defines the base currency used.

---

**Note:** The selected value of the currency catalog is passed to the core interface, which performs the conversion during the log-in process. Thus, if a user changes this value, he needs to log off and on again to activate the new setting.

---

**Default:** Undefined

**Type:** Catalog

## Locale.Date.DisplayFormat

**Description:**

Defines the display format of dates.

**Note:** Date auto-completion, which adds the current year to a date the user enters, is only available for formats where the year is at the end, i.e., DMY and MDY.

When the default value is set to "Server", it means that the settings of the application pool identity user are applied.

---

**Default:** Server

**Type:** Combobox

### **Locale.Date.Separator**

**Description:** Defines the character used to separate the values in the date.

**Default:** Server

**Type:** Combobox

### **Locale.LCID**

**Description:** Defines the language ID.

**Default:** Server

**Type:** Combobox

### **Locale.Number.DecimalSeparator**

**Description:** Defines the character used as the decimal separator for numeric values.

**Default:** Server

**Type:** Combobox

### **Locale.Number.ThousandsSeparator**

**Description:** Defines the character used as digit grouping symbol for numeric values.

**Default:** Server

**Type:** Combobox

### **Locale.Time.DisplayFormat**

**Description:** Defines the format used to display the time.

**Default:** Server

**Type:** Combobox

### **Locale.Time.Separator**

**Description:** Defines the character used as "time separator", e.g., the colon in 11:48:51.

**Default:** Server

**Type:** Combobox

---

## Locale.Time.Timezone

**Description:** Defines the time zone in which the user is located.

Available values:

- Server: The time zone of the web server is used.
- Client: Aurea CRM.Web determines the time zone currently used by the client. If no valid time zone is found, "Server" is used as a fallback.

---

**Note:** The client's time zone is only determined at login.

---

- All existing time zones.

**Default:** Server

**Type:** Combobox

## Login Parameter

This topic holds information about the Login parameter.

### Login.EnableLoginInDefaultLanguage

**Description:**

When checked for the default user's configuration that is used for the login page, the option **My Default Language** text group `login_text` is displayed in the language selection drop-down list. See [Login Page Languages](#) on page 175 for more details.

By selecting this option, the user can start Aurea CRM.Web in his default language, i.e., the language entered in his **Rep** record. For further details, see [Reps](#) on page 31.

---

**Note:** If this option is checked and the user's login language is not explicitly set, the user is logged on in Aurea CRM.Web's default language, as per **settings.xml (Aurea CRM.Web)** in the [Aurea CRM.Web Installation Guide](#).

---

**Default:** Unchecked

**Type:** Checkbox

## Lyris Parameters

For more detailed descriptions of the parameters, see [Configuring Lyris HQ](#) on page 676.

### Lyris.Enabled

Set this parameter to true to enable Lyris features in Aurea CRM. Default value is false.

Type: CheckBox

### **Lyris.SyncMiddlewareUri**

Specify the URI of the Lyris Sync Middleware's Active MQ endpoint. The URI should also include the port number. For example, `tcp://52.204.45.159:61616`.

Type: String

### **Lyris.SyncMiddlewareUsername**

The username for authenticating with the Lyris Sync Middleware configured at the URL specified in the parameter: `Lyris.SyncMiddlewareUri`. For example, **MQ2014000029**. You can provide the username or Aurea support can provide one, when you place a request. For more information on the ordering process, see [Ordering Access to the ACM SyncMiddleware](#).

Type: String

### **Lyris.SyncMiddlewarePassword**

The password for authenticating with the Lyris Sync Middleware configured at the URL specified in the parameter: `Lyris.SyncMiddlewareUri`. For example, **MQabc12345911**. For more information on the ordering process, see [Ordering Access to the ACM SyncMiddleware](#).

Type: String

### **Lyris.SegmentGeneratedNotificationEmail**

The email address that receives an information message from Lyris, when Lyris completes the generation of a segment requested by Aurea CRM.

Type: String

### **Lyris.TriggerOnLyrisMessageDeleted**

The name of the trigger to be executed by Aurea CRM, when an event notifying the deletion of a Lyris message is received by Lyris Sync Middleware consumer.

Type: String

### **Lyris.TriggerOnLyrisContactBounced**

Specify the name of the trigger to be executed by Aurea CRM, when a target group member's contact status is set to Bounced in Lyris. For detailed information on triggers and how to define them, see [Triggers](#) in the *CRM.core Administrator Guide*.

Type: String

### **Lyris.DoubleOptinMapping**

Specify the value to enable the mapping of Lyris demographics to CRM fields used to create and update CRM records during the processing of `ContactDataUpdateMessage` with DoubleOptin. The specified value should have the format:

`infoAreaId:demographicId=fieldId,demographicId=fieldId|infoAreaId`

An example value would be `FI:1=5,2=8|KP:1=4,2=19,3=8|PE:1=2,2=7`.



---

When a new recipient subscribes to a Lyris HQ mailing list on the Lyris HQ Landing page, he get a confirmation message before being added to the Mailing list. CRM adds contact to corresponding infoarea and not add to Target group, if the DoubleOption parameter is used. He is added to the appropriate info are based on the following heuristics:

1. Add the recipient to the Company contact if a Company contains the recipient as a contact email address.
2. If Company doesn't exist, then CRM.Web checks if a Company with the domain exists.
3. If Company with the domain exists, it checks if there is a PersonInCompany with a contact email of the recipient.
4. If PersonInCompany doesn't exist, the recipient is added to it.
5. If the Company with email or domain doesn't exist, CRM.Web checks if there is Person (PE) with contact email address of the recipient.
6. If Person doesn't exist, the recipient is added to the Person info area.

The DemographicID is defined in Lyris HQ and fieldID is defined in the ACRM infoarea. The types of DemographicID and fieldId should be of compatible types and are listed below:

1. String=String
2. String=Catalog value
3. Date=Date
4. Number=Number

**Type:** Character

### **Lyris.ActivityExecutionDemographicId**

Specify the demographics ID, CRM is required to provide when requesting demographics to be created for the CRM fields, in Lyris. For more information on retrieving this value from Lyris HQ, see [Getting Lyris Web Configuration Parameter Values from Lyris HQ](#)

**Type:** multi select

### **Lyris.Host**

Specify the URL of the server hosting the Lyris API. It should be either an IP address or a fully qualified name of the machine hosting Lyris HQ API. Example value, **elabs14.com**.

For more information on retrieving this value from Lyris HQ, see [Getting Lyris Web Configuration Parameter Values from Lyris HQ](#).

**Type:** String

### **Lyris.APIPassword**

Specify the authentication password to use, when Aurea CRM makes an API request to Lyris. For more information on retrieving this value from Lyris HQ, see [Getting Lyris Web Configuration Parameter Values from Lyris HQ](#).

**Type:** String

### **Lyris.MailingListNamePrefix**

Specify a prefix with which a new mailing list id is created in Lyris. A mailing list is created, when Aurea CRM transfers a marketing activity to Lyris.

**Type:** String

### **Lyris.MailingListUriTemplate**

Specify the string template used by Aurea CRM to generate the URL of a mailing list entity created by Lyris. The mailing list is created by Lyris on demand, when Aurea CRM transfers a marketing activity to Lyris. You can provide one of the following:

1. <https://Lyrishq.net> – In this case the user is forwarded to the Lyris HQ dashboard and he has to find the required mailing list object.
2. `https://<Lyris.Host>/v3/mailling_list.html?NEXTPAGE=mailling_list_view&setmlcookie=on&mlid={lyrisMailingListId}`
  - Where `<Lyris.Host>` is the value of the Emaillab domain name set in the **Lyris.Host** web configuration parameter described above.

The specified template value should contain the placeholder tag `{lyrisMessageId}` which is replaced by the Lyris message id generated by Lyris. A sample value is [https://www.elabs14.com/v3/mailling\\_list.html?NEXTPAGE=mailling\\_list\\_view&setmlcookie=on&mlid=385](https://www.elabs14.com/v3/mailling_list.html?NEXTPAGE=mailling_list_view&setmlcookie=on&mlid=385).

**Note:** Click on the Lyris mailing list link generated in Aurea CRM. This opens the required mailing list in Lyris Emaillabs, if the following conditions are satisfied:

1. You have already logged into Lyris HQ at <https://lyrishq.net>.
2. An instance of Emaillabs is already launched from Lyris HQ.

If the above conditions are not satisfied then you are directed to the login page at <https://lyrishq.net>.

**Type:** String

### **Lyris.MessageUriTemplate**

Specify the string template used by Aurea CRM to generate the URL of a message entity created by Lyris. The message is created by Lyris on demand from Aurea CRM. You can provide one of the following:

1. <https://Lyrishq.net> – In this case the user is forwarded to the Lyris HQ dashboard and he has to find the required mailing list object.
2. `https://<Lyris.Host>/v3/mailling_list.html?NEXTPAGE=mailling_list_view&setmlcookie=on&mid={lyrisMessageId}`

Where `<Lyris.Host>` is the value of the Emaillab domain name set in the **Lyris.Host** web configuration parameter described above.

The specified template value should contain the placeholder tag `{lyrisMessageId}` which is replaced by the Lyris message id generated by Lyris. A sample value is [https://ip-172-30-90-193.ec2.internal/v3/mailling\\_list.html?NEXTPAGE=mailling\\_list\\_view&setmlcookie=on&mlid=459](https://ip-172-30-90-193.ec2.internal/v3/mailling_list.html?NEXTPAGE=mailling_list_view&setmlcookie=on&mlid=459).

**Note:** Clicking on the Lyris message link generated in Aurea CRM opens the required message in Lyris Emaillabs, if the following conditions are satisfied:

1. You have already logged into Lyris HQ at <https://lyrishq.net>.
2. An instance of Emaillabs is already launched from Lyris HQ.

If the above conditions are not satisfied then you are directed to the login page at <https://lyrishq.net>.

**Type:** String

### Lyris.SegmentUrlTemplate

This web configuration parameter is used to enable the administrator to configure the URL template that is used by ACRM to construct a URL that leads to the UI page where a Lyris segment is displayed.

This URL is used for the creation of an anchor link that is displayed in the 3rd step of "Transfer of Target group to Lyris" process (see [Configuring target group creation and scheduling](#)). This enables the user to view the corresponding segment in Lyris UI on a single click. For further details on configuring Lyris HQ, see [Configuring Lyris HQ](#) on page 676.

The template URL stored in **Lyris.SegmentUrlTemplate** supports two placeholders:

- **{lyrisSegmentId}**: This placeholder exist inside Lyris.SegmentUrlTemplate and is replaced during the construction of the URL by the Lyris segment ID.
- **{mailingListId}**: This placeholder may or may not exist in Lyris.SegmentUrlTemplate as it is not mandatory. If it exists, it is replaced by the mailing list ID during the construction of the URL. If it is not present in the template then it is just ignored.

You can use one of the following two strings in the template to be used by CRM in generating the URL of a segment in Lyris:

- `https://Lyrishq.net` – This takes the user to the Lyris HQ dashboard, where he needs to find the required mailing list object.
- `https://<Lyris.Host>/v3/mailling_list.html?NEXTPAGE=mailling_list_filter_create&ID={lyrisSegmentId}`, where `<Lyris.Host>` is the value of the domain name set in the [Lyris.Host](#) web configuration parameter as described in the above section. The specified template value should contain the placeholder tag `{lyrisSegmentId}` as explained above. For example, `https://ip-172-30-90-193.ec2.internal/v3/mailling_list.html?NEXTPAGE=mailling_list_filter_create&ID=459`.

**Note:** The following conditions must be met to open the required message in Lyris s, when you click on the Lyris segment link generated in Aurea CRM:

- You must be logged into Lyris HQ at <https://lyrishq.net>.
- An instance of Emaillabs is already launched from Lyris HQ.

If the above conditions are not met, the user is redirected to the login page at <https://lyrishq.net>.

---

**Type:** String

### **Lyris.SiteID**

Specify the site id, a lyris parameter that is sent as part of all API requests by Aurea CRM. For more information on retrieving this value from Lyris HQ, see *Getting Lyris Web Configuration Parameter Values from Lyris HQ*.

**Type:** String

### **Lyris.TriggerOnLyrisContactBounced**

Specify the name of the trigger to be executed by Aurea CRM, when a target group member's contact status is set to Bounced in Lyris. For detailed information on triggers and how to define them, see [Triggers](#) in the *CRM.core Administrator Guide*.

**Type:** String

### **Lyris.TriggerOnLyrisContactUnsubscribed**

Specify the name of the trigger to be executed by Aurea CRM, when a target group member's contact status is set to Unsubscribed in Lyris. For detailed information on triggers and how to define them, >> [Triggers](#) in the *CRM.core Administrator Guide*.

**Type:** String

### **Lyris.DemographicOptionsLanguage**

The language in which the demographics are created in Lyris. Valid values are: ger, eng, fre, spa, por, nld, dan, ita, cze, hun, slk, pol, gre, ukr, sln, rus, swe, fin, nor, tur, cro, ser, ron, jpn, chn, kor, bul, tha, and alb.

**Type:** String

## **Maintenance Parameters**

This topic holds information about the Maintenance parameters.

### **Maintenance.BulkExecutionStart**

**Description:** Specifies the time (hh:mm) for mass data operation to-dos to be processed.

**Default:** Undefined

**Type:** Character

### **Maintenance.CatalogMaintenance.ShowExtendedFields**

**Description:** If unchecked, the additional catalog maintenance fields **Description** and **Valid from/to** of the **Catalog** info area (KA) are hidden from in the **Edit value** dialog. For further details, see [Editing Catalog Values](#) on page 37.

---

If unchecked, the additional catalog maintenance fields and the valid from/to catalog. This parameter is not part of `UPDATE_DEFAULT` but needs to be created as Checkbox if required.

**Default:** Undefined

**Type:** Checkbox

### **Maintenance.CheckForSystemLocksInterval**

**Description:** Defines how often Aurea CRM.Web polls the server for system locks for the current user. The value is set in minutes. For further details, see **System Locks** in the [CRM.Core Administrator Guide](#).

**Default:** 4

**Type:** Number

## **Marketing Parameters**

This topic holds information about the Marketing parameters.

### **Marketing.ControlFileSeparators**

---

**Important:** Internal parameter - do not change!

---

### **Marketing.MarketingCalendar.ContextMenus**

**Description:** Defines the context menus used in Marketing Calendar.

**Default:** Empty

**Type:** Character

### **Marketing.QuickCampaign.RecordsType**

**Description:** Defines whether activity (default) or mailing records are created for quick campaigns. For further details, see [Quick Campaigns](#) on page 845.

**Default:** Empty

**Type:** Combobox

### **Marketing.QuickCampaign.WarningLimit**

**Description:** Defines the limit above which a warning about a large number of records displays when creating a quick campaign.

**Default:** Empty

**Type:** Number

### **MarketingActivityMessageBuilder.UseDefaultEmailMessageBuilder**

**Description:** Enable this parameter to allow the of use the default email message builder for Marketing activities.

**Default:** Unchecked

**Type:** Checkbox

### **MarketingCalendar.FilterSets**

**Description:** Enable the predefined Marketing Calendar filters. See [MarketingCalendar](#) on page 432 for further details. These filters are listed in the marketing calendar's **Filter** drop-down list.

Filter sets for the marketing calendar are defined as described for calendar filter sets. For further details, see [Calendar.FilterSets](#).

**Default:** Empty

**Type:** Combobox

## **Campaign Designer Parameters**

This topic holds information about the Campaign Designer parameters.

### **CampaignDesigner.AKExpandMask**

**Description:** Defines the expand mask to be used for editing AK values in the Campaign Designer.

**Default:** AK.Designer

**Type:** Character

### **CampaignDesigner.CMExpandMask**

**Description:** Defines the expand mask to be used for editing CM values in the Campaign Designer.

**Default:** AK.Designer

**Type:** Character

### **CampaignDesigner.ExecutionStatePolling**

**Description:** Specifies the polling timeout interval, measured in milliseconds, to check the state of a marketing activity execution until the marketing activity execution is complete.

**Default:** 2500

**Type:** Integer

## **Matchup Parameters**

Use the following parameters to configure the match-up behavior when creating new records in Aurea CRM.Web.

### **Matchup.BusinessLogic**

**Description:** Defines whether the match-up logic applied when identifying duplicate records is the logic used in Aurea CRM.Web ("web") or the logic used in Aurea CRM.Win ("core").

---

If "web" is selected, the fields defined in the info area Matchup control are compared when matching up records. For further details, see [Matchup control](#).

If "core" is selected, the settings in the **Configuration** info area (MC) determine whether the internal logic or an external third-party application is used.

---

**Note:** Since the internal match-up logic of Aurea CRM win is only available for the info areas **Company**, **Person** and **Property** (BTB only), the option "core" without defining an external application should only be used if the match-up is solely configured for these info areas otherwise errors occur.

---

**Default:** Web

**Type:** Combobox

### **Matchup.DuplicateMaxRows**

**Description:** Defines the maximum number of possible duplicated rows shown after match-up.

**Default:** Empty

**Type:** Number

### **Matchup.PersonsInCompanyOnly**

**Description:** If checked, the person match-up takes into account only persons from the company for which the user wants to create a new record. By default, the match-up is performed globally.

**Default:** Unchecked

**Type:** Checkbox

## **Menu Parameter**

This topic holds information about the Menu parameter.

### **Menu.Delay**

**Description:** Determines the delay in milliseconds for opening sub-menus on mouse-hover.

**Default:** 200

**Type:** Number

## **Notification Parameters**

This topic holds information about the Notification parameter.

All notification parameters are measured in seconds. When the timeout = 0 the notification does not fade out:. Users must manually exit such notifications.

For more information on facilities, see the article "The Notification Service" at <https://support.aurea.com>.

### **Notification.TimeoutError/Information/Warning**

**Description:** Defines the timeout before hiding error messages displayed in the Inbox.

To define notification timeouts per facility, define and set the `Notification.Timeout` Web Configuration parameters using the following syntax:

```
Notification.TimeoutError.<FacilityName>
```

**Default:** 0

**Type:** Number

### **Notification.Information**

**Description:** Defines the timeout before hiding informational messages displayed in the Inbox.

To define notification timeouts per facility, define and set the `Notification.Timeout` Web Configuration parameters using the following syntax:

```
Notification.TimeoutInformation.<FacilityName>
```

**Default:** 3

**Type:** Number

### **Notification.Warning**

**Description:** Defines the timeout before hiding warning messages displayed in the Inbox.

To define notification timeouts per facility, define and set the `Notification.Timeout` Web Configuration parameters using the following syntax:

```
Notification.TimeoutWarning.<FacilityName>
```

**Default:** 10

**Type:** Number

## **Offline Parameters**

Learn to use Offline Parameters to configure Aurea CRM.Web Offline.

Information on how to configure Aurea CRM.Web offline can be found on **Web Configuration Parameters** in the *Aurea CRM.Web Offline Installation & Technical Guide*.

### **Offline.IncludeRootConfigurationInSync**

**Description:** Enables the inclusion of root in the offline synchronization.

**Default:** Checked

**Type:** Checkbox



## Offline.MaxTimeLimitForSync

**Description:** Defines the maximum number of hours before the user is prompted to synchronize.

**Default:** 24

**Type:** Number

## Offline.ShipDataModel

**Description:** Allows the export of the Data Model.

**Default:** Empty

**Type:** Checkbox

## Phone Parameters

Use the following parameters to configure the behavior of telephone fields and of CRM.Phone.

To learn more about Phone Parameters, see [CRM.Phone](#) on page 824.

---

**Note:** Fields are defined as telephone fields in the data model. For example, **Telephone** column, Type "Telephone" and "Cell phone". For further details, see **Data Model** in the [CRM.Core Administrator Guide](#).

---

## Phone.ActivateBrowserOnCall

If checked, an alert message is generated and the browser is activated and brought to the foreground when a call comes in.

---

**Note:** The browser may not always be brought to the foreground as it depends on the user's local system and browser settings.

---

Default: Unchecked

**Type:** Checkbox

## Phone.AutoSuggestFormat

Defines the formatting of country and area codes displayed as a ghost text in phone and fax fields (Expand and List view in Edit mode).

If you define the format as "+{countryCode}({areaCode})" the following ghost text is displayed:

The screenshot shows a CRM form with the following fields and values:

- Street: \_\_\_\_\_
- Postal Code: 1030
- City: Wien
- Country: Austria
- Address (selected tab): \_\_\_\_\_
- Type: \_\_\_\_\_
- Classification: \_\_\_\_\_
- Event: \_\_\_\_\_
- Country Code: 0043
- Area Code: 01
- Telephone: +43(1) | \_\_\_\_\_
- Fax: \_\_\_\_\_

The ghost text "+43(1)" is displayed in the telephone field.

---

**Note:** Telephone fields (and the mapping of country/area code fields) are defined in the data model, >> **Data Model** in the *CRM.Core Administrator Guide* (Info Area Options > Telephone).

---

**Note:** When creating new records the country and the area codes are only entered upon saving the record. Workaround: Define a `Hook Function` for the **Country Code** field (updating the country code as soon as the user selects a **Country**) and adapt the `Select-Function` for **Postal Code** to also update the **Area Code** field. Example in `UPDATE_DEFAULT: FI` details control.

---

The formatting of country and area codes (leading zeros, plus sign) as defined in the respective fields is stripped before the formatting defined with `Phone.AutoSuggestFormat` is applied.

To ensure the trimming of trailing blanks, you can define a format string like `"+{countryCode}({areaCode}) {number}"`. The placeholder `{number}` is replaced with nothing thus preventing the former trailing blank from being trimmed. Later it may be replaced with the telephone number of the company (e.g. for person records).

If `Phone.AutoSuggestFormat` is undefined, no ghost text is displayed.

Type: Character

### Phone.EmailSearchListGroups

**Description:** This field group(s) is used for the result listing the phone & email Search.

**Default:** FI.EmailPhone,KP.EmailPhone,PE.EmailPhone

Type: Character

### Phone.FastCopyEnabled

**Description:** If checked, the phone icon displayed next to a phone field (in Expand view in Edit mode) offers a context menu (`M_Telephone`) that allows the user to choose between dialing the number (`A_DialNumber`) or copying the number from the CRM.Phone toolbar to the field (`A_CopyNumber`). For more information, see Copying Numbers in Aurea CRM.Web in the CRM.Phone User Manual.

**Default:** Unchecked

Type: Checkbox

### Phone.IncomingCallAction

**Description:** Defines the action carried out when an incoming call is answered by CRM.phone. Available values:

- `Do Nothing`: No action is carried out in Aurea CRM.Web.
- `Popup Question (default)`: Asks the user if he wants to open the URL or launch the process defined in their phone profile. For details, see Special Aurea CRM.Web options in the CRM.phone Administrator Guide.
- `Automatic URL Redirect`: Directly opens the URL or launches the process (defined in the phone profile) upon answering an incoming call.

Type: Combobox

### **Phone.OpenIncomingInNewTab**

**Description:** If checked (default), the defined action for incoming call is executed in a new tab.

**Default:** Checked

**Type:** Checkbox.

### **Phone.ShowOnCall**

**Description:** If checked, the phone toolbar is displayed if collapsed upon incoming calls.

**Default:** Checked

**Type:** Checkbox.

### **TelephoneField.DisableLink**

**Description:** If checked, disables the display of links in the Aurea CRM telephone fields in expand views or lists.

**Default:** Checked

**Type:** Checkbox.

### **Phone.PhoneBook.CacheMaxRecord**

**Description:** If checked, disables the display of links in the Aurea CRM telephone fields in expand views or lists.

**Default:** Checked

**Type:** Checkbox.

### **Phone.PhoneBook.CacheSetting**

Use this parameter to specify the caching behavior of the CRM.Web handler for CRM.Phone profiles. You can specify one of the following values:

- **-1:** Set this value to disable caching.
- **0:** Set this value to enable caching on demand. This ensures that CRM.Phone profiles are not retrieved in bulk from the database but the records are retrieved and stored in the cache, only on receiving a request for a user's phone profile.
- **1 and above:** Set this value to enable caching and retrieve the specified number of records from the database initially.

**Default:** 3000

**Type:** Number

## Process Parameters

This topic holds information about Process parameters.

### Process.DefaultAction

**Description:** Defines the default action that is called when the user double-clicks a list item on a search page within a process. For further details, see [Default Action](#) on page 289.

**Default:** A\_Select

**Type:** Character

### Process.DefaultMenu

**Description:** Defines the default menu for list records on a search page within a process.

**Default:** M\_Wizard

**Type:** Character

## Query Parameters

Use the following parameters to configure the definition and behavior of queries in Aurea CRM.Web.

### Query.OptionalColumns

**Description:** Allows you to specify additional field ids to display in query selector results list. For example, specifying `{'ID': [11,12,0]}` displays the additional fields: **Last Name**, **First Name**, and **Rep/Group ID**.

**Default:** Empty

**Type:** JSON

### Query.BackgroundExecutionStart

**Description:** Allows you to define a start time for processing asynchronous queries on the server. For further details, see [Background Queries](#) on page 502. This allows the processing of background queries to be scheduled for off-peak hours to avoid performance issues.



Enter the desired start time using the format hh:mm e.g. "23:00".

**Default:** Empty

**Type:** Character

## Query.CountRecords

**Description:** The count feature does not execute the query itself but counts the rows that would be returned. This operation is resource-consuming for the server and thus `Query.CountRecords` should be activated with care and set to a sensible limit e.g. "500".

- If this parameter is set to 0, the count feature is disabled, i.e. the  Count Results button is not available.
- If set to any value > 0 (e.g. "500"), the  Count Results button is available in the `Query` and `QueryFooter` headers and the message "> x rows" is displayed if more rows exist.
- Use this setting to restrict the number of result rows. If fewer result rows are returned, they are displayed. If more result rows are returned, a message is displayed instead of the rows.
- Enter -1 for no limit.

**Default:** 0

**Type:** Number

## Query.FieldsSortOrder

**Description:** Determines the sort order of the field selection dialogs when defining filters and queries in Aurea CRM.Web: "alphanumeric" or "by id". This setting defines the default behavior and is overruled by the user's sort and column order settings.

**Default:** Alphanumeric

**Type:** Combobox

---

**Note:** Order clause (sort) in queries is parsed based on the info areas order in the query so, Order (sort) fields is rearranged by CRM if it is not in the correct order. For instance, in the following query :

```
select ('FI.SAP-No', AddressType, CompanyName, Country, ZipCode)
from (ZA)
plus (FI)
orderby ('FI.SAP-No', AddressType)
```

The actual sort order is "AddressType" then "FI.SAP-No" because "AddressType" is from the first info area (ZA) and "FI.SAP-No" from the second one.

---

## Query.HideSystemFields

**Description:** By default, system fields (**Created on**, **DeICd** etc.) are not visible when defining filters and queries. For further details, see **System Fields** in the *CRM.Core Administrator Guide*.

If you define `Query.HideSystemFields` and set it to false (= unchecked), system fields are displayed when defining filters and queries.

This parameter is not part of UPDATE\_DEFAULT but needs to be created if required.

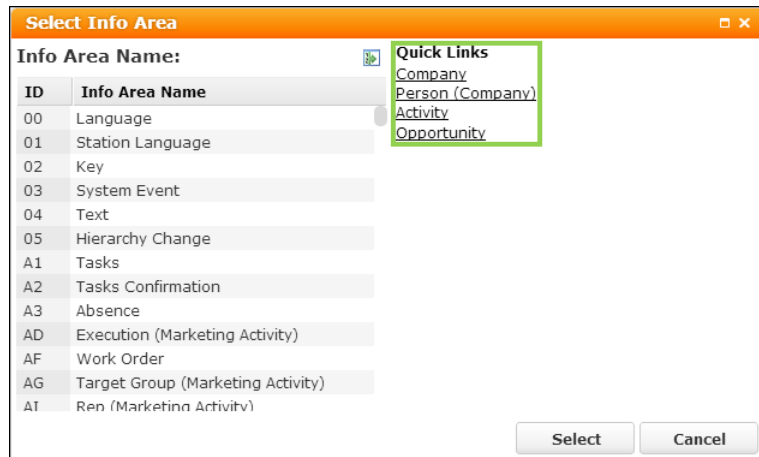
---

**Default:** Unchecked

**Type:** Checkbox

### Query.ImportantInfoAreas

**Description:** Defines the info areas and their order that are displayed as Quick Links in the info area selection dialog when creating new queries.



**Default:** "FI,KP,MA,Y1" in the BTB vertical, "FI,KP,MA" for the OTC and FS verticals

**Type:** List

### Query.LoadFrom

**Description:** Defines from which databases – Aurea CRM and/or CRM.Designer – Aurea CRM.Web loads queries. For further details, see [Storing Queries](#) on page 495.

Available values:

- `Aurea CRM`: Aurea CRM.Web loads queries from the Aurea CRM database only.
- `CRM.Designer`: Aurea CRM.Web loads queries from the CRM.Designer database only.
- `both`: Aurea CRM.Web loads queries first from the Aurea CRM database and then from the CRM.Designer database.
- `both, CRM.Designer first`: Aurea CRM.Web loads queries first from the CRM.Designer database and then from the Aurea CRM database.

**Default:** both, CRM.Designer first

**Type:** Combobox

### Query.MaxRecords

**Description:** Defines the maximum number of records that can be returned by a query.

Enter "-1" for "All".

**Default:** 1000

---

**Type:** Number

### **Query.NumberOfResults.Default**

**Description:** Number of records of the query result that are displayed at once. Possible values: any number (e.g. 20,50,100...), "search" or "all".

This setting affects the QueryResult in the query editor, the QueryRun page and the QueryResultPresenter widget in forms. (Restriction for QueryRun:Query.NumberOfResults.Default overrules the maxRows argument, i.e. its value is used for calculating the page size.)

If undefined, the value of the Web Configuration parameter Search.MaxResults is used. For further information, see [Search.MaxResults](#).

**Default:** 50

**Type:** Number

### **Query.NumberOfResults.DefaultForRun**

**Description:** Defines the default value for the number of records **Max. Rows** dropdown list returned by a query execution. Enter "-1" for "All".

**Default:** 100

**Type:** Number

### **Query.NumberOfResults.Steps**

**Description:** Steps for record count returned by a query. Users can choose between these steps when they execute queries. Enter "-1" for "All".

**Default:** 20,50,100,1000,10000

**Type:** List

### **Query.Parameters**

**Description:** With this parameter, you can prevent users from exporting and/or saving the results of a query. Available values:

- Disable exporting of query results
- Disable saving of query results

**Default:** Undefined

**Type:** Checkboxes

### Query.Rights

**Description:** Defines the user's rights for loading, saving and executing queries. Available values:

- `disabled`: Queries are not available.
- `readonly`: Allows the user to execute predefined queries.
- `read+define`: Allows the user to edit and create queries and save them as private queries.
- `all (default)`: Grants full access rights. The user can also create and save public queries.

**Default:** all

**Type:** Combobox

### Query.SaveDateExpressionWithoutEvaluation

**Description:** Enable this parameter to ensure that data expressions in a UQL statement are not evaluated, before saving the UQL statement to the database. This is useful when you are using expressions in a date condition. For example, with this option enabled, the expression `$$$##+1m` is not evaluated and is stored as it is in the FT table.

---

**Note:** Date expressions are not directly supported by the CRM.Core. You can use these queries only with .Net framework components.

---

**Default:** Unchecked

**Type:** Checkbox

### Query.SaveTo

**Description:** Defines to which databases a query defined in Aurea CRM.Web is saved. For further details, see [Storing Queries](#) on page 495. Available values:

- `Aurea CRM`: Aurea CRM.Web loads queries from the Aurea CRM database only.
- `CRM.Designer`: Aurea CRM.Web loads queries from the CRM.Designer database only.
- `both databases`: Aurea CRM.Web loads queries first from the Aurea CRM database and then from the CRM.Designer database.
- `both, CRM.Designer first`: Aurea CRM.Web loads queries first from the CRM.Designer database and then from the Aurea CRM database.

**Default:** Aurea CRM

**Type:** Combobox

### Query.ShowParameterMissingWarning

**Description:** Enable this parameter to validate a query's fields for required values. If a query's field is missing a value a warning message pops up informing the user about the empty fields.



---

**Default:** Checked

**Type:** Checkbox

### Query.ShowVirtualFields

**Description:** If checked, fields flagged as VirtualCore (i.e. virtual fields with fixed IDs from 4200 to 4230, e.g. **Created by ID**) are available for filter and query definitions in Aurea CRM.Web. For further details, see **Fields with Fixed IDs** in the *CRM.Core Administrator Guide* (2nd table).

If unchecked, those fields are hidden.

---

**Note:** Regardless of this setting, the Number of records (RecCount) field (4230) is always displayed for queries of type WITH SUMMED, PLUS SUMMED and HAVING. Reference fields (Z-fields) are never available for filter and query definitions in Aurea CRM.Web. However, they can be defined as output fields in queries.

---

The `Query.ShowVirtualFields` setting also applies to the fields displayed in the ViewAllFields dialog. For further details, see [ViewAllFields](#) on page 319.

**Default:** Checked

**Type:** Checkboxes

### Query.DisableAutoAddArrayFields

**Description:** Check this option to prevent all the fields present in an array field to show up in a query result. This ensures that only the array fields specified in a query filter are displayed.

---

**Note:** By default, queries are automatically adapted to include all associated fields of the specified array field.

---

**Default:** Unchecked

**Type:** Checkboxes

## Query Analysis Parameters

This topic holds information about Query Analysis parameters.

### Query\_Analysis.RememberLastSearch

**Description:** If checked, the last search criteria input by the user are automatically entered in query and analysis search masks.

**Default:** Checked

**Type:** Checkbox

## Query\_Analysis.ShowLockedCatalogValues

**Description:** Use this parameter to define whether catalog values locked via the **Maintenance** module or via tenant rights are available when defining filters, queries and analyses. Set `Query_Analysis.ShowLockedCatalogValues` to **Display Locked Catalog Values** to allow users to select locked catalog values, otherwise set it to **Hide Locked Catalog Values**.

The following table shows whether locked catalog values are displayed (Y | N), and the effect of `Query_Analysis.ShowLockedCatalogValues` (Show | Hide):

Where	Show	Hide	Comment
Search	N	N	
List (View mode)	Y	Y	
List (Edit mode)	Y	Y	The value is grayed out in the drop-down list
Expand	Y	Y	The value is grayed out in the drop-down list
Filter	Y	N	
Query	Y	N	
Query Snapshot	Y	Y	
Analysis	Y	N	For analyses, you can override <code>Query_Analysis.ShowLockedCatalogValues</code> using <code>Analysis.ShowLockedCatalogValues</code> . For further details, see <a href="#">Analysis.ShowLockedCatalogValues</a> .

**Default:** Display Locked Catalog Values

**Type:** Combobox

## QuickSearch Parameters

This topic holds information about QuickSearch parameters

### QuickSearch.Definition

**Description:** Specifies the path and name to a configuration-specific quick search definition. For further details, see [Quick Search](#) on page 346. Specify the path relative to the application path.

If the path is invalid or the file doesn't exist, the Quick Search is not displayed and a message is written to the client log.

Change `QuickSearch.Definition` to undefined (empty), to completely hide the Quick Search.

**Default:** `data\settings\QuickSearch.json.js`

**Type:** Character

## Record Parameters

This topic holds information about Record parameters

### Record.DefaultAction

**Description:** Defines the default action that is called when a record is clicked. For further details, see [Default Action](#) on page 289.

**Default:** `A_Expand`

**Type:** Character

### Record.History.Slim

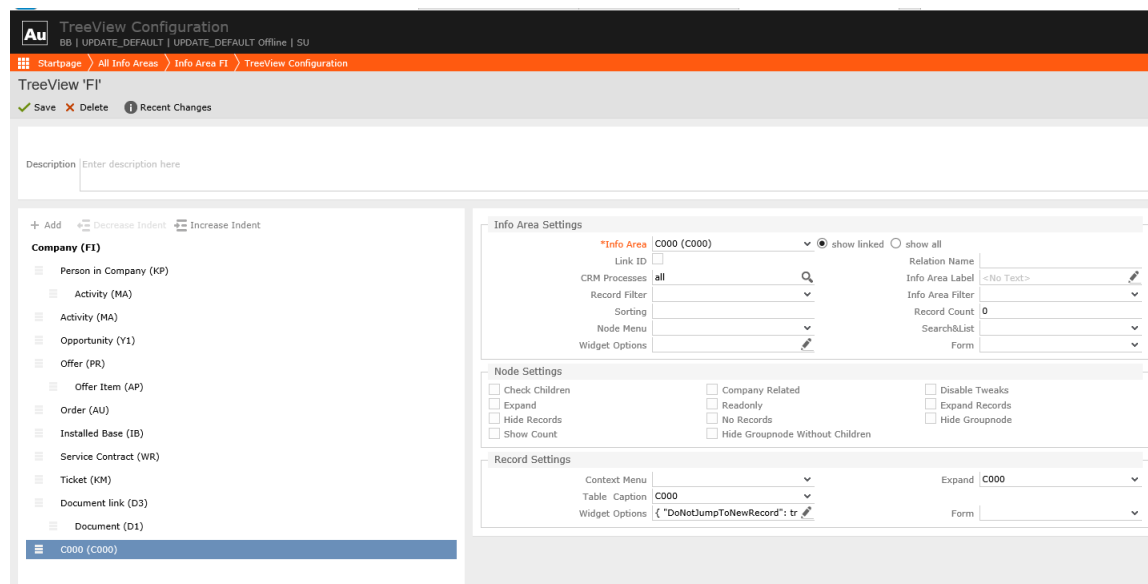
**Description:** If set to `true`, the record history (H0) only displays a sub-set of columns. For further details, see [Displaying a Record's History](#) on page 824. The **ID**, **Windows User Name**, **Windows Computer Name** and **Application** columns are hidden.

**Default:** Unchecked.

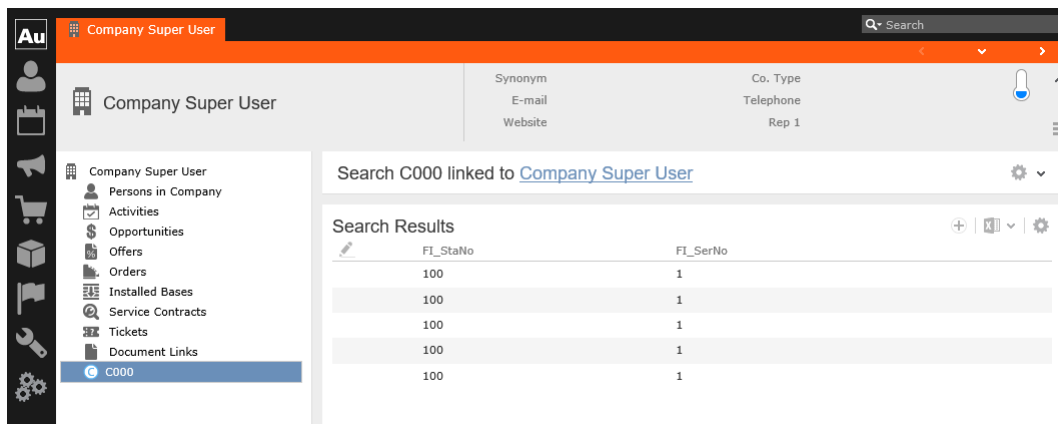
**Type:** Checkbox

### DoNotJumpToNewRecord

**Description:** Support of the new **Tree View** configuration option `DoNotJumpToNewRecord` is added.



If set to `true`, the record tree node does not switch content in case that a new record is added in the same **info area**. The list of records is kept in the content area and no details of newly added record are shown:



The option can have a value of `true` or `false`. If the value is blank it defaults to `false`.


**Default:** False

**Type:** Widget Options

## RecordSelector Parameters

This topic holds information about the RecordSelector parameters.

### RecordSelector.EnableDeleteLink

**Description:** If checked, a **Remove Link** button  is displayed next to link fields in Expand view allowing users to remove previously selected link records.

**Default:** Checked

**Type:** Checkbox

### Header.EnableRecordSelectorHeaderButtons

**Description:** Enable this parameter to display the buttons added to a record selector header of an info area. By default this parameter is not enabled and buttons added to a record selector header are not displayed.

**Default:** Disabled

**Type:** Check Box

### RecordSelector.GenericInfoAreas/RecordSelector.InfoAreasWithGenericLinks

**Description:** `RecordSelector.InfoAreasWithGenericLinks` determines which info areas can be linked to other info areas via generic links.

`RecordSelector.GenericInfoAreas` determines the info areas that can be accessed via generic links from the info areas defined under `RecordSelector.InfoAreasWithGenericLinks`.

These parameters control the use of generic links in the query designer. For example, a **Task** (A1) can be linked to any other info area using a link field. If you do not want to show all info areas in the query designer you may use these parameters.

For all info areas that are listed in `RecordSelector.InfoAreasWithGenericLinks`, all `RecordSelector.GenericInfoAreas` are offered as link info areas and vice versa.

For both parameters enter the info area IDs separated by comma.

`RecordSelector.GenericInfoAreas` also globally defines the info area selection (context menu) for generic link fields: The setting is used for all generic link fields that have no `Select-Function` defined. For further details, see [selectRecord for Generic Link Fields](#) on page 253.

---

**Note:** A link field's context menu only lists info areas that meet the following conditions:

- The info area's key consists of station number plus serial number.
  - The user has read access.
  - A Search&List configuration exists.
- 

### Examples:

```
RecordSelector.InfoAreasWithGenericLinks = A1
```


```
RecordSelector.GenericInfoAreas = MA,CM,KM
```


Tasks can be linked to activities, campaigns and tickets in the query designer.

**Default:** Undefined

**Type:** List

### RecordSelector.ShowSelectorIfDefined

**Description:** By default, the record selector icon  is not displayed for read-only fields, i.e., fields with either data model field property = `readonly` or `CRM.Designer` field attribute `Readonly = true`.

If this parameter is checked and a `Select-Function` is defined for a field, the record selector icon  is displayed regardless of a read-only setting. For further details on `Select-Function`, see [Select-Function](#) on page 240.

Use this feature e.g. to indicate read-only link fields that actually copy data to other editable fields or to let users reassign parent records if required.

---

**Note:** If a user has no update rights for an info area, no record selector icons  are displayed, regardless of `RecordSelector.ShowSelectorIfDefined`.

---

**Default:** Checked

**Type:** Checkbox

## RelationshipHierarchy Parameters

This topic holds information about RelationshipHierarchy parameter.

### RelationshipHierarchy.MaxRecursionDepth

**Description:** By default the recursion depth for the relationship hierarchy is 20. For further details, see [RelationTreeView](#) on page 446. Use this parameter to reduce the number of levels displayed in the relationship tree (recommended when performance problems occur).

This parameter is not a part of UPDATE\_DEFAULT but needs to be created as number if required.

**Default:** Undefined

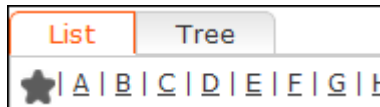
**Type:** Number

## Rep Parameters

This topic holds information about Rep parameter

### Rep.ABCStrings

**Description:** Letters offered as links in the rep picker dialog ("ABC bar"). Separate multiple values by comma.



If undefined, the language-dependent text no.58 from the text group `RepUtils_text` is used.

In order to display only the **All** option, you have to delete the values in the `Rep.ABCStrings` parameter AND the text no.58 from the `RepUtils_text` Text Group.

**Default:** A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z

**Type:** List

### Rep.AutoCompleteCharacters

**Description:** Defines the required number of characters for the auto-completion function to start when entering a name into a rep field.

**Default:** 1

**Type:** Number

### Rep.Dialog.AvailableViews

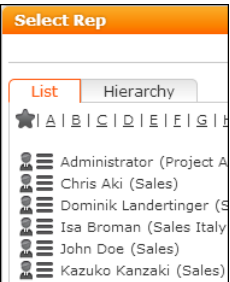
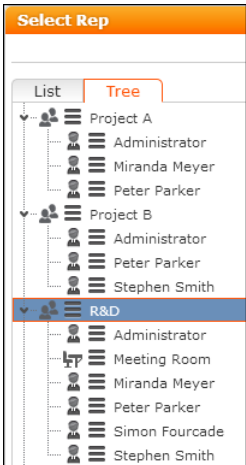
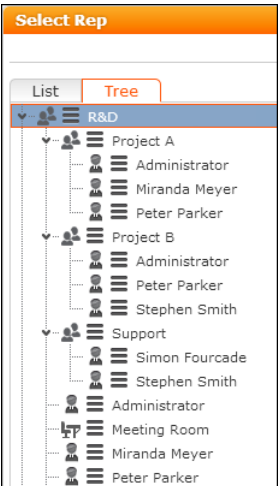
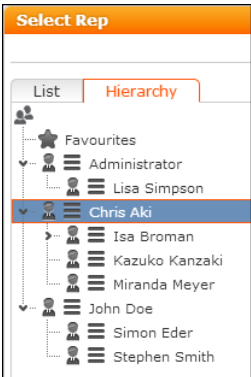
**Description:** Defines the available tabs in the rep picker dialog.

Group assignments via the ID **Org. Group** field and via the **Group Allocation** (GZ) info area are taken into account. For further details, see **Rep** in the *CRM.Core Administrator Guide*.

Reps, groups and resources are displayed depending on the context, e.g. the MA **Participants** fields list reps, groups and resources, the ID **Deputy** field only lists reps.

Available values:

- **Tree View**: Lists all organizational groups with their assigned reps. Hierarchical groups are not taken into account, i.e. all groups are listed on the first level regardless of the hierarchical structure of the groups.
- **List View**: Lists all reps in alphabetical order.
- **Tree View (hierarchical)**: Lists reps according to the rep hierarchy. For further details, see **Rep Hierarchy** in the CRM.Core Administrator Guide.
- **Tree View (organizationally grouped)**: Lists all organizational groups with their assigned reps. Hierarchical groups are taken into account. If a rep is assigned to multiple groups (via GZ), the rep is displayed below all relevant group nodes.

List	"Flat" Tree	Group Hierarchy Tree	Rep Hierarchy
			

**Default:** List View, Tree View (organizationally grouped)

**Type:** Checkboxes

### Rep.Dialog.DefaultView

**Description:** Defines the active tab of the rep picker dialog.

Available values:

- Tree View
- List View
- Tree View (hierarchical)
- Tree View (organizationally grouped)

**Default:** List View

**Type:** Combobox

### Rep.Dialog.HierarchyTree.ShowAssignedOnly

**Description:** Check this parameter to hide the **No group** node in the rep picker in the hierarchy tree view. Reps that don't belong to a group are not displayed in the hierarchical rep tree.

**Default:** Unchecked

**Type:** Checkbox

### Rep.Dialog.List.MaxResults

**Description:** Determines the maximum number of reps displayed in the list view of the rep picker. If more reps exist, "..." is added at the end and can be clicked to display more reps. `Rep.Dialog.List.MaxResults` applies to search results as well as to the user's selection of a letter or **All** from the ABC bar.

**Default:** 150

**Type:** Number

### Rep.Dialog.Tree.MaxResults

**Description:** Determines the maximum number of reps displayed in a tree node when the rep picker is expanded. If more reps exist in the organizational group, "..." is added at the end and can be clicked to display more reps.

**Default:** 30

**Type:** Number

### Rep.DisplayInternal

**Description:** By default, reps flagged as **Internal** in the **Rep** info area are not displayed in the rep picker and the auto-completion results. Use the `Rep.DisplayInternal` parameter if you want users to see internal reps.

This parameter is not part of UPDATE\_DEFAULT but needs to be created as Checkbox if required.

**Default:** Undefined

**Type:** Checkbox

### Rep.DownloadOptions

**Description:** Determines which reps are transferred to the client and thus available for selection. Available values:

- **Download All:** All reps are downloaded from the server. Inactive reps are available and can be used for defining filter criteria (searches, filters, queries). They are **not** selectable in rep fields when creating/editing a record.

Use the `Rep.HideInactiveInSearch` web Configuration parameter to disable inactive reps when defining filter criteria, i.e. same behavior as for rep fields in records. For further details, see [Rep.HideInactiveInSearch](#).

- **Exclude inactive reps:** Reps flagged as **Inactive** (in the **Rep** info area) are not downloaded.



---

**Note:** Reps that are not accessible due to restricted access rights (defined in the Aurea CRM.win **Rights** module) are never downloaded.

---

**Note:** The `Rep.DownloadOptions` parameter affects rep groups as well.

---

**Default:** Download All

**Type:** Combobox

### Rep.HideInactiveInSearch

**Description:** If set to `true`, inactive reps are no longer available (selectable) when defining filter criteria (searches, filters, queries).

This parameter is not part of UPDATE\_DEFAULT but needs to be created as Checkbox, if required.

**Default:** Undefined

**Type:** Checkbox

### Rep.MaxAutocompleteResults

**Description:** Limits the number of entries displayed in the auto-complete box for reps. If more than `Rep.MaxAutocompleteResults` reps match the input string "..." is displayed at the end of the list.

**Default:** 50

**Type:** Number

### Rep.RepSetSettings

**Description:** Use this parameter to hide one or more options in the **Reps and Rep Groups** dialog of the calendar. Available values:

- Hide option "CurrentRep"
- Hide option "Group"
- Hide option "Group Leader"
- Hide option "Deputy"
- Hide option "All"

**Default:** Undefined

**Type:** Checkboxes

## Resource View Parameters

This topic holds information about Resource View Parameters

### ResourceView.FilterSets

**Description:** Defines the predefined filters offered in the sidebar of the **Resources** page and the **Dispatching Dashboard**. For more information. For further details, see [Calendar.FilterSets](#).

This parameter is not part of UPDATE\_DEFAULT but needs to be created as Combobox if required.

**Default:** Undefined

**Type:** Combobox

### ResourceView.ShowHiddenItems

**Description:** If checked, the hidden activity records are displayed as little dots when the selected time scale is too large.

You can overrule the default setting by setting the **Show hidden items** flag in the Resource View settings.

This setting applies to the Resources view and the Dispatching Dashboard.

**Default:** Unchecked

**Type:** Checkbox

## Sales Parameters

This topic holds information about Sales Parameters

### Sales.TerritoryManagement.InfoAreas

**Description:** Determines the info areas for which the territory allocation is calculated. If this parameter is not set, "FI,KP" is used as default for both versions.

**Default:** "FI,KP,OJ" (BTB), "FI, KP" (FS)

**Type:** List

## Schema Parameters

This topic holds information about Schema Parameters

### Schema.EnableLiveCalculations

**Description:** Use this parameter to enable the calculation of fields values based on core business logic upon editing the field instead of upon saving the record. To enable calculations for specific info areas, enter a comma-separated list of info area IDs. To globally enable calculations, enter '\*'.  
Examples for core calculations:

Examples for core calculations:

- When entering the **Postal Code** field e.g. for a company, the **City** is automatically entered using the information retrieved from **City (OR)** info area.
- When changing the start time of an activity (MA), the end time is modified accordingly ensuring that the duration remains unchanged.

If disabled (i.e. `Schema.EnableLiveCalculations = undefined`), you need to define e.g. field hooks for updating such fields immediately.

For more information, see the article “How to Enable CRM.Core live calculations for Aurea CRM” in <https://support.aurea.com>.

**Default:** Disabled

**Type:** Character

## Search Parameters

This topic holds information about Search Parameters

### GlobalSearch.AddWildCardAt

**Description:** You can control the automatic addition of wildcards to a search key using this parameter. Global search function splits the words in a search term separated by space into keys. For each search key wildcards are added before and/or after the search key based on the configuration defined using this parameter.

You can select one of the following options:

- **None**—wild card entries are not added to the search keys.
- **Start**—wild card entries are added at the beginning of a search key.
- **End**—wild card entries are added at the end of a search key.
- **Both**—wild card entries are added at the beginning and end of a search key. This is the default behaviour of the global search function.

**Default:** Both

**Type:** Radiobutton

### Search.AlwaysReadNumRecords

**Description:** If this parameter is set to a value other than zero, regardless of the requested count of records the server always requests `<Search.AlwaysReadNumRecords>` records from the database. If more records exist for the entered search criteria, a message is displayed "More than `<Search.AlwaysReadNumRecords>` matching records; please refine your search."

Use this setting to restrict the number of search results and for Oracle databases to work around the ROWNUM problem with ROWNUM being applied before sorting. For further details, **Oracle: ROWNUM is applied before ORDER BY** in the *Aurea CRM.Web Installation Guide*.

**Default:** 0

**Type:** Number

### Search.AutoExpandOnSingleHit

**Description:** If this parameter is checked and a search returns exactly one record, the search results list is not displayed but the default action is automatically executed for the record. For further details, see [Default Action](#) on page 289.

You can override this setting for individual `Search` actions via the action parameter `autoExpandOnSingleHit=<true|false>`. For further details, see [autoExpandOnSingleHit](#).

**Default:** Not checked

**Type:** Checkbox

### Search.AutoStartWhenLinked

**Description:** Determines the auto search behavior of linked searches (but only if the `autoSearch` parameter in the `Search` action is not set). For further details, see [autoSearch](#).

If checked, the search is automatically started if a link record is passed to the search view.

**Default:** Checked

**Type:** Checkbox

### Search.ClearSearch

**Description:** Determines which options are available for clearing search criteria/resetting a search. For further details, see [Resetting a Search](#). Enable/disable the following options:

- Clear search criteria
- Reset search settings

Use the `Search.ShowSettingsButton` Web Configuration parameter to globally enable/disable the reset options on search pages. For further details, see [Search.ShowSettingsButton](#).

**Default:** Both enabled

**Type:** Checkboxes

### Search.CopyCriteriaOnNew

**Description:** If enabled, values in the search mask are automatically transferred to the Expand view when creating a new record from the search results.

---

**Note:** If `Search.CopyCriteriaOnNew` is enabled, then the 1st catalog value is copied into the expand view.

---

For a search value to be transferred, the respective field must be part of the Details control used by the `ExpandNew` action. Values from an enhanced search field are written to the field for which the `Enhanced Search Field` attribute is defined.

Ranges are not copied (e.g. the **From/To PC** range in the company search).

Values are not transferred if the field in the Expand view is read-only.

Default values, and values from copy fields and empty values specified in the `ExpandNew` action via the `copySourceFields/copyTargetFields` and `emptyFields` arguments overrule values that are transferred from the search.

For searches with `autoSearch = true` no values are transferred.

Search criteria are stored per info area in the current session. Expiration time is set to 5 minutes.

**Default:** Enabled

**Type:** Checkbox

---

### Search.FlavorSelectDefaultAction

**Description:** Defines the default action that is called when double-clicking a record in the search results of a process or record selector. For further details, see [Default Action](#) on page 289.

**Default:** A\_Select

**Type:** Character

### Search.FlavorSelectMenu

**Description:** Defines the default context menu used for the search results of a process or record selector.

**Default:** M\_Select

**Type:** Character

### Search.HideMoreFields

**Description:** Allows you to disable the **More Fields ...** option from the **Add more ...** drop down menu in a search control for all info areas by setting the parameter to true.

**Default:** False

**Type:** Boolean

### Search.MaxResults

**Description:** Default number of records displayed in search results lists and in sub-lists.

---

**Note:** The value specified here is displayed as default in the search mask's **Max. Records/Page** drop-down list.

---

---

**Note:** This parameter does not apply to the Quick Links list.

---

**Default:** 10

**Type:** Number

### Search.NoFilterDefinition

**Description:** Globally restricts the usage of filters to the selection of predefined filters.

If checked, the **More Filters** entry is removed from the **Add more** drop-down list on all Search pages. Users can only select predefined filters and can no longer create or edit any filters in Aurea CRM.Web.

This parameter is not part of UPDATE\_DEFAULT but needs to be created as Checkbox if required.

**Default:** Undefined

**Type:** Checkbox

### **Search.NotCompanyRelated**

**Description:** A list of info areas where the **Company-Related** check box is not displayed in the search mask, when items of that info area are displayed linked to a company.

**Default:** The following defaults are used:

- BTB: IB, Y1, ZA, BQ, AU
- FS: AU
- OTC: AU, VG, IB, PS, LS, KK, ST, WM, AB, FK, SA, TM, VF, WK

**Type:** List

### **Search.PhoneEmailMaxResults**

**Description:** Defines the maximum number of results returned by the **Phone & email Search**.

---

**Note:** Since the `ID_NET_PHONEEMAILSEARCH` menu action has `CachedMode` enabled by default, all of these results are transferred to the client to allow paging in the result list) Therefore, update recommends not to specify to high a value.

---

**Default:** 500

**Type:** Number

### **Search.RestoreSearchOnBrowserBack**

**Description:** Use this parameter to enable the use of the BackButton (**Previous** button) on opening a result record page to return to the info area's search page, with the search result list and search conditions restored to the previous search context. This parameter works with the `Search.RememberDynamicFilter` and `Search.RememberDynamicFields` parameters defined below.

**Default:**

**Type:** Checkbox

### **Search.Result.FocusHotKey**

**Description:** Defines a hotkey to automatically select the first row of a search result. This also activates the respective search sub-list.

**Default:** Alt+L

**Type:** Character

### **Search.ShowClearSearchButton**

**Description:** Enable this parameter to display the **Clear Search** button globally across all info areas providing search capabilities. This button allows the user to clear all search criteria (default and user-defined search values and filters) from the search mask.

---

**Default:** Unchecked

**Type:** Checkbox

### Search.ShowResetSearchButton

**Description:** Enable this parameter to display the **Reset Search** button globally across all info areas providing search capabilities. Users can use this button to clear additional search fields and remove added filters, clear search values or reset to default field values if applicable. It also clears the search results.

**Default:** Unchecked

**Type:** Checkbox

### Search.ShowSettingsButton

**Description:** Determines if the **Clear** (and **Reset**) options are available on search pages. For further details, see [Resetting a Search](#).

If undefined (default) or checked, the **Clear** (and **Reset**) options are available on all search pages. If unchecked, they are hidden.

Use the `Search.ClearSearch` Web Configuration parameter to determine which options (**Clear** and/or **Reset**) should be available. For further details, see [Search.ClearSearch](#).

**Default:** Undefined

**Type:** Checkbox

### Search.ShowOwnItems

**Description:** If set to `true`, the **With Me** checkbox is added to all search forms. Users can select this checkbox to search for their owned items.

---

**Note:** The checkbox is hidden if the info area does not contain non-virtual rep or participant fields.

---

**Default:** Unchecked

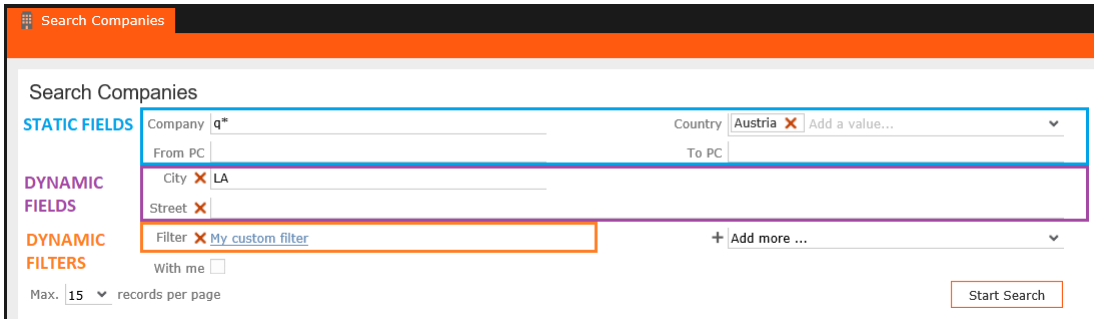
**Type:** Checkbox

## Remember Search Options

This topic holds information about Remember Search Options

Web configuration parameters used to configure search behaviors related to how the system remembers search have been enhanced.

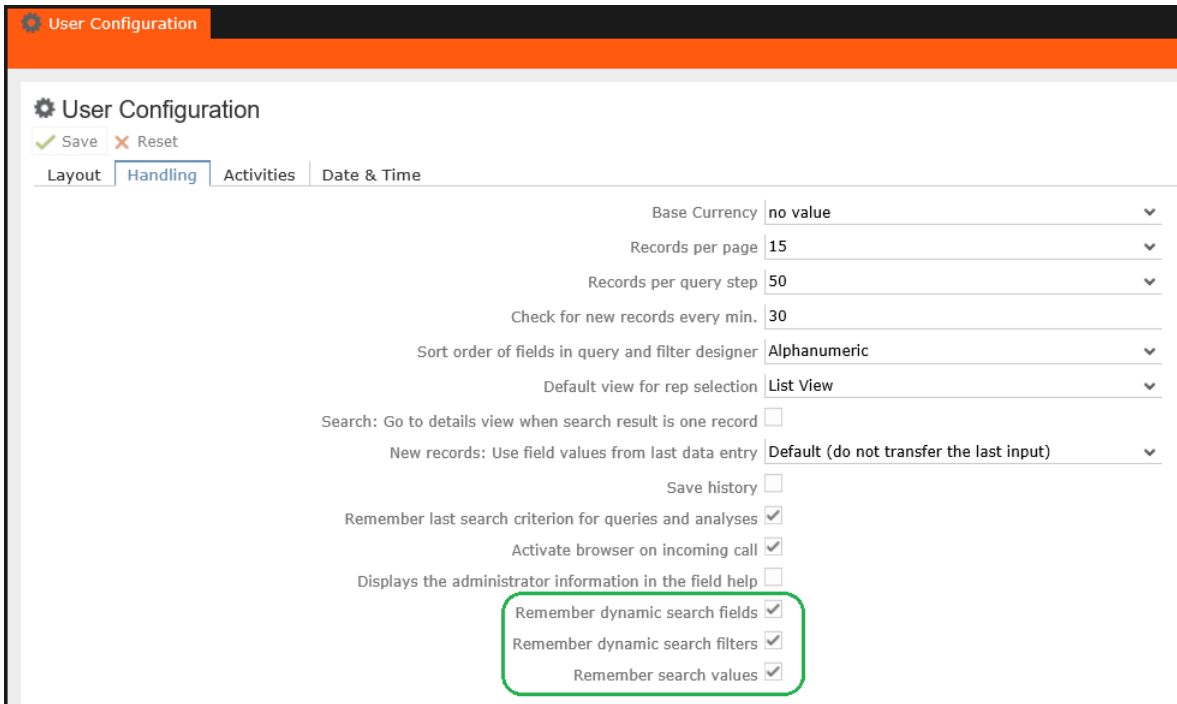
Search views are structured as follow:



CRM.Web version 11.5.0 and later have two parameters renamed:

- `Search.RememberSearch` is renamed to `Search.RememberDynamicFields`
- `Search.RememberFilter` is renamed to `Search.RememberDynamicFilters`

Users can change these configuration parameters using their **User Configuration** view, checking or unchecking to enable or disable the correspondent options.



The following tables show all the possible combinations of web configuration and their behaviors:



Remember Dynamic Fields	Remember Dynamic Filters	Remember Search Values	Static Field visible	Dynamic Field visible	Dynamic Filter visible	Static + Dynamic fields values remembered
True	True	True	Yes	Yes	Yes	Yes
True	True	False	Yes	Yes	No	No
True	False	True	Yes	No	Yes	Yes
True	False	False	Yes	No	No	No
False	True	True	No	Yes	Yes	Yes
False	True	False	No	Yes	No	No
False	False	True	No	No	Yes	Yes
False	False	False	No	No	No	No

For better visibility:

Web Configuration Parameters		
RememberDynamicFields	RememberDynamicFilters	RememberSearchValues
TRUE	TRUE	TRUE
TRUE	TRUE	FALSE
TRUE	FALSE	TRUE
TRUE	FALSE	FALSE
FALSE	TRUE	TRUE
FALSE	TRUE	FALSE
FALSE	FALSE	TRUE
FALSE	FALSE	FALSE

Fields Visible			Static + Dynamic Fields Values Remembered
Static Fields	Dynamic Fields	Dynamic Filters	
YES	YES	YES	YES
YES	YES	YES	NO
YES	YES	NO	YES
YES	YES	NO	NO
YES	NO	YES	YES
YES	NO	YES	NO
YES	NO	NO	YES
YES	NO	NO	NO

### Search.RememberDynamicFilter

Use this parameter to configure how the search screen remembers the dynamic filters values for future use.

This parameter works with `Search.RememberDynamicFields` parameter described in this section.

The default value for this parameter is `true`.

If you want the search screen to remember only the values of the default filter fields, then set `Search.RememberDynamicFilters` to `true` and `Search.RememberDynamicField` to `true`.

If you want the search screen to remember values of only the dynamic fields but not the filter fields used, then set `Search.RememberDynamicFilters` to `false` and `Search.RememberDynamicField` to `true`.

**Default:** Checked

**Type:** Checkbox

### **Search.RememberDynamicField**

Use this parameter to configure how the search screen remembers the dynamic field values for future use.

If checked (default), the user's search settings, including catalog filters are saved between sessions in the user's roaming settings). For more information see [Remember Search Settings](#).

The default value for this parameter is `true`.

Uncheck this parameter if you want to reset all searches once the user leaves the page. To override this setting for a specific search action, use the `rememberSearch` parameter. For more information, see [rememberSearch](#).

This parameter is used along with `Search.RememberSearchValues` to ensure that the values set for a filter are also remembered across search sessions. For more information, see [Search.RememberSearchValues](#).

---

**Note:** If you later re-activate the "remember search settings" behavior (and the users have not reset their browser cache), their last search settings is reloaded from the user's roaming settings.

---

**Default:** Checked

**Type:** Checkbox

### **Search.RememberSearchValues**

Use this parameter to configure how the search screen remembers the search values for future use.

Enable this parameter to ensure that values set for the search filters are saved and presented to the user during the next search session. By default, this parameter is unchecked and the values set in the search mask are not stored between search sessions.

The default value for this parameter is `true`.

This parameter is used along with the `Search.RememberDynamicFields` to ensure that the search mask used in a previous search is also remembered. For more information, see [Search.RememberDynamicField](#).

---

**Note:** This option requires the parameter `Search.RememberDynamicField` to be active. For more information see [Search.RememberDynamicField](#).

---

**Default:** Unchecked

**Type:** Checkbox

## Security Parameters

This topic holds information about Security Parameters

### Security.HidePathInformation

Use this parameter to hide or display the document path in the System Information page. By default this parameter is set to `True` and the document path is hidden. For more information see, **Security Settings > Hiding Document Path in System information** in the *Web Installation Guide*.

**Default:** Checked

**Type:** Checkbox

## SerialEntry Parameters

This topic holds information about Security Parameters

### SerialEntry.<Name>Filter

**Description:** Defines which filters are displayed in the options form, and the initially selected filter for the quick add `<Name>`. For further details, see [name](#).



The action parameter `filterType` overrides the initial value specified in `SerialEntry.<Name>Filter`. For further details, see [filterType](#).

The following `SerialEntry.<Name>Filter` parameters are defined in UPDATE\_DEFAULT:

**BTB version:**

#### SerialEntry.DefaultFilter

**Default:** None Selected

**Type:** Radio button

#### SerialEntry.OfferFilter

**Default:** None Selected

**Type:** List

#### SerialEntry.OrderFilter

**Default:** All

**Type:** List

OTC version:

**SerialEntry.DefaultFilter**

Default: None Selected

Type: Radio button

**SerialEntry.InstalledBaseFilter**

Default: None Selected

Type: Radio button

**SerialEntry.OfferFilter**

Default: None Selected

Type: List

**SerialEntry.OfferFilter.SpecialOffer**

Default: Special Offer

Type: Radio button

**SerialEntry.OrderFilter**

Default: All

Type: List

**SerialEntry.OrderFilter.SpecialOffer**

Default: Special Offer

Type: Radio button

**SerialEntry.POSFilter**

Default: All

Type: List

**SerialEntry.ProductDiscussionFilter**

Default: None Selected

Type: Radio button

**SerialEntry.PromotionalMaterialFilter**

Default: None Selected

Type: Radio button

**SerialEntry.RefundingFilter**

Default: None Selected

Type: List

**SerialEntry.SampleFilter**

Default: None Selected

Type: Radio button

**SerialEntry.StockControlFilter**

---

**Default:** None Selected

**Type:** Radio button

## System Parameters

This topic holds information about System Parameters

Use the following parameters to configure system settings for Aurea CRM.Web.

### System.CompiledScripts

**Description:** If checked, all JavaScript files are combined ("compiled") into one large script. Loading just this one large script is much faster than loading many individual small ones.

The compiled scripts named `$<name>.compiled.js` are created in the `..\scripts\compiled\` folder of your Aurea CRM.Web installation. Please note that every configuration can have a different set of global scripts, therefore more than one compiled script may exist in your installation.

In order to create the compiled script, the user "NETWORK SERVICE" must have MODIFY rights on the `..\scripts\compiled\` folder (the setup procedure of Aurea CRM.Web takes care of this). If the compiled script cannot be created (due to insufficient rights), Aurea CRM.Web defaults to the standard behavior of loading each script individually.

---

**Note:** It is NOT automatically determined whether a compiled script is still up-to-date (i.e. the script is not automatically re-created on demand). Hence, if you change a script or add a new global script, you need to manually delete the compiled scripts (the scripts are then re-created upon the next access of a web client). When updating your installation, the compiled scripts are automatically deleted by the setup routine.

---

**Note: Hint:** If you are a developer and therefore change your scripts frequently, we recommend disabling the compiled scripts feature in your user configuration.

---

General information on how to add scripts to extend Aurea CRM.Web, see the article "Scripts" at <https://support.aurea.com>.

**Default:** Checked

**Type:** Checkbox

### System.Debug

**Description:** If checked, the client log is displayed with LogLevel Error. For further details, see [Client Log](#) on page 22 . Use the URL parameter `?debug` for LogLevel Debug. For further details, see [URL Parameters](#) on page 887.

If you use CRM.phone, log messages prefixed with `PhoneInfo` are displayed in the client log showing the initialization of the phone toolbar.

**Default:** Unchecked

**Type:** Checkbox

## System.LauncherPort

**Description:** The port used by CRM.launcher to connect to the browser (default = "18188").

You can set `System.LauncherPort` to any valid free port if you want the clients to use CRM.launcher for document and telephony functionalities. For further details, see [CRM.launcher\(Enterprise\)](#) on page 793.

If set to "-1", the "old" Aurea CRM.Web client browser plug-in (`np.update.web.client.dll`) is installed and used instead. For further details, see [Plug-ins](#) on page 24.

**Default:** 18188

**Type:** Textbox

## System.ShowAdminHelp

**Description:** If checked, the online field help additionally displays the contents of the **Administrator Information** field.

For more information, see [Field Help](#) on page 50.

**Default:** Unchecked

**Type:** Checkbox

## TableCaption Parameters

This topic holds information about TableCaption Parameters

### TableCaptions.DisableCache

**Description:** Defines the info areas whose table captions should be requested from the server and thus refreshed every time a header control is drawn. Enter the desired info area IDs separated by commas.

Enter "ALL" to reload data for all info areas. This may have a negative impact on the system performance.

**Default:** no value

**Type:** List

## Tree Parameters

This topic holds information about Tree Parameters

### Tree.DefaultNodeSettings

**Description:** Use this parameter to define global settings for all Tree views. Available values:

- **Displays the number of records for a group node:** Globally sets the `DisplayCount` flag. For further details, see [Node Settings](#).
- **Show the symbol for expanding a branch only if the brach actually contains records:** Globally sets the `CheckChildren` flag.

This global setting override the individual tree configurations.

**Default:** Unchecked

**Type:** Checkboxes

### Tree.ListDefaultAction

**Description:** If a user clicks on an info area node in a Tree view, the Search mask plus results are displayed. `Tree.ListDefaultAction` defines what happens when the user double-clicks a result record in the list. For further details, see [Default Action](#) on page 289.

Available values:

- `Default Action for Info Area` The default action of the info area defined in the Expand configuration is executed.
- `SearchAndList Default Action` The default action defined in the Search&List configuration is executed.
- `Select Record in Treeview` An Expand view of the record is displayed in the Tree view and the corresponding tree node is selected (`A_TreeDefault`).

If `Tree.ListDefaultAction` is undefined, the default action of the info area defined in the Expand configuration is executed.

**Default:** Undefined

**Type:** Combobox

## UpdateServer Parameters

This topic holds information about UpdateServer Parameters

### UpdateServer.EventEvaluationStartDate

**Description:** By default, the calculations of topics & lectures and accommodations & venues are performed without any date restriction. For further details, see **Evaluating Events** in the *Web Business Logic User Guide*. To take into account only records after a certain date, enter the desired date in `UpdateServer.EventEvaluationStartDate` (format YYYYMMDD).

**Default:** Undefined

**Type:** Character

### UpdateServer.RepID

**Description:** Enter the rep ID of the CRM.server user as `UpdateServer.RepID` (e.g. "010001677"). For further details, see [CRM.Server](#) on page 15. To-dos for background processing are then created for this rep.

If `UpdateServer.RepID` is not defined, the following items are not displayed in Aurea CRM.Web:

- `ID_NET_CALCULATE_ACCOMODATION_VENUE_EVALUATION`
- `ID_NET_CALCULATE_TOPIC_LECTURE_EVALUATION`

- `Query_RunInBackground` and `Query_RunInBackground_CSV` buttons. For further details, see [Query Background Processing Buttons](#) on page 318.
- `ID_NET_BulkOperations`. For further details, see [Mass Data Operations](#) on page 39.

The following items are displayed as inactive (grayed out):

- `CreateTargetGroup` and `CreateMarketingActivity` buttons. For further details, see [Marketing Activity Buttons](#) on page 316.

---

**Note:** When CRM.server processes marketing activity executions, the query defined in the marketing activity execution (AD) record is run with the access rights of the rep defined by the **Sel. User ID** field. For further details, see **Automatically Executing Activities** in the *Aurea CRM win User Manual*.

---

---

**Note:** If you are transporting campaign templates from one server to another ensure that the field-5 station number of the info area is set to the current server.

---


**Default:** 1000200287

**Type:** Character

## Visit Planning Parameters

This topic holds information about Visit Planning Parameters

### VisitPlanning.FilterSets

**Description:** Specifies options for filtering the items displayed on the Visit Planning map. For further details, see [VisitPlanning](#) on page 479. These filters are listed in the  (**Settings**) drop-down list.

Filter sets for Visit Planning are defined as described for calendar filter sets. For further details, see [Calendar.FilterSets](#).

### VisitPlanning.SuggestionsDateFormat

**Description:** Visit Planning suggested dates are displayed in the format this parameter is set. You can use the following conventions to set this parameter:

- D: Shows a single digit date less than 10.
- DD: Shows the date as a double digit. For example, 03.
- DDD: Shows the name of the day. For example, SU for a Sunday.

---

**Note:** The names of the days are taken from the respective texts already defined in the designer.

---

- M: Shows a single digit month less than 10.
- MM: Shows the date as a double digit. For example, 04 for April.
- MMM: Shows the name of the month. For example, Feb for February.



---

**Note:** The names of the months are taken from the respective texts already defined in the designer.

---

- YY: Shows the last two digits of the year. For example, 17 of the year 2017.
- YYYY: Shows the four digits of the year. For example, 2017.

Using the above naming convention, if you set the parameter to the value MMM DD of YYYY, the suggested date displayed for visit planning is Feb 07 of 2017.

The names of the days and months are taken from the respective texts already defined in the designer.

## Other Web Configuration Parameters

This topic holds information about other miscellaneous web configuration parameters

### StartPage

**Description:** Specifies the page displayed after login. Select a menu action from the drop-down list.

You can add any (context-free) menu action to the list. For further details, see [Modifying Web Configuration Parameters](#). Users can select their start page via **Settings > User Settings > Layout > Startpage**.

**Default:** ID\_NET\_STARTPAGE\_GENERAL

**Type:** Combobox

### Style

**Description:** Determines the style sheet used by Aurea CRM.Web. Available value: **update**

To apply a project-specific style, you need to add another option to `style`. For further details, see [Modifying Web Configuration Parameters](#). Enter the name of the folder containing the custom style as **Value**. For further details, see [Customizing Styles](#) on page 15.

**Type:** Combobox

### DefaultLayout

**Description:** Use this parameter to enforce a default layout view for all clients logging in to CRM.Web from any device.

---

**Important:** This configuration applies to all users. It is not possible to have separate configuration per user, as the layout template selection happens from the login page.

---

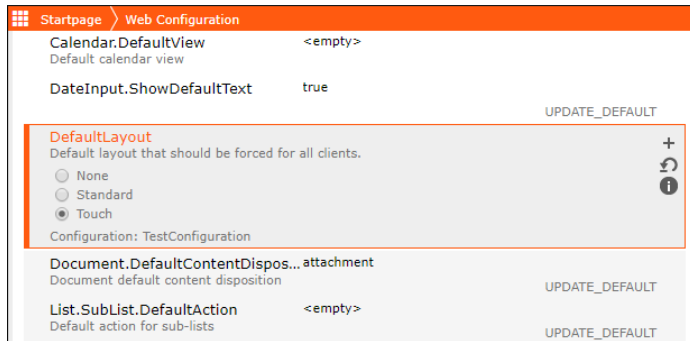
**Default:** None Selected

**Type:** RadioButton

**Options:**

- None—allows CRM.Web to automatically determine the device type and load the required layout.
- Standard—enforces the standard layout for all clients logging into CRM.Web from any device type.
- Touch—enforces the touch device layout for all clients logging into CRM.Web from any device type.

For example, if you wish to ensure that all clients get the standard layout on logging in to CRM.Web and not the touch layout, then set this parameter to `Standard`. A client can override this configuration setting by setting the URL parameter `touch=true/false` as described in [URL Parameters](#) on page 887.



---

**Note:** The layout view is loaded in the following order of priority:

1. Layout specified by the URL parameter `touch`. For more information, see [URL Parameters](#) on page 887.
2. `DefaultLayout` web configuration parameter described above.
3. If the above two are not specified CRM.Web uses autodetection to determine the layout view to load.

---

**HttpRequest.EnableHttpGetMethod**

**Description:** Enable this option to convert all HTTP requests to GET requests which allows for browser caching.

---

**Note:** If this is enabled, you should increase the `maxUrl` and `MaxQueryString` IIS Request Limits and `Http Runtime maxQueryStringLength` in the `web.config` for IIS.

---

**Default:** Unchecked

**Type:** Checkbox

**VirtualInfoArea.List.ReadMode**

**Description:** This parameter defines the default behavior of virtual info areas in lists (Search Result Sets, SearchSubLists, ExpandChildren and ListView-Widgets in Forms). For further details, see [virtualInfoAreaReadMode](#) on page 285.

This parameter is not part of UPDATE\_DEFAULT but needs to be created as Character or Combobox if required.

### **VirtualInfoArea.Query.ReadMode**

**Description:** This parameter defines the default behavior of virtual info areas in queries (query snapshot, saved queries, queries in forms and the `QueryRun` action). For further details, see [virtualInfoAreaReadMode](#) on page 285.

This parameter is not part of UPDATE\_DEFAULT but needs to be created as Character or Combobox if required.

### **Widget.GridEditorOptions**

**Description:** Use this parameter to configure the grid editor's default options. For example, setting it to `continueEditAfterSave` to `false` alters the default behaviour of serial entry grid editors, of reverting back to edit mode after editing and saving an entry.

**Default:** Undefined

**Type:** JSON

### **CKEditor.EnableImagePlugin**

**Description:** Use this parameter to insert and edit images on your CRM instance. Check to enable this feature for inserting and editing images.

Uncheck to disable the feature and restrict adding or editing images. On disabling this option, any existing image cannot be modified.

**Default:** Unchecked

**Type:** Checkbox

### **Browser.RefreshCurrentTabInsteadCurrentPage**

**Description:** Use this parameter to control how CRM.Web reacts on attempts to refresh a page from the keyboard using F5 and Ctrl-R keys.

When this parameter is enabled and there is some active tab on the page, only the tab is refreshed. If there is no active tab, the entire page is refreshed.

When the parameter is disabled, the whole page is refreshed.

Web Configuration	
BB   UPDATE_DEFAULT   UNITTEST_DEFAULT	
Startpage > Web Configuration	
Enables Fusion Charts format instead CRM number format	
Analysis.MaxRecords Max. records in analysis	1000
Analysis.ResultGrid.PageSize	10
Analysis.Rights Analysis rights	7
Analysis.ScrollableGrid If true, analysis result grid layout becomes horizontally scrollable	<empty>
Analysis.ShowLockedCatalogValues Display locked catalog values in analyses (overrides LockedCatalogMode)	SHOW
Analysis.TraceMode Quick Analysis Trace Mode	none
Browser.RefreshCurrentTabInsteadCurrentPage If enabled, refreshing via keyboard shortcuts would refresh the current tab instead of the whole page	<empty>
BrowsingTabs.AllowReordering	false

**Default:** Unchecked

**Type:** Checkbox

### Sync.SkipLinkedFields

**Description:** Use this parameter to exclude `z-fields` and `Linked InfoAreas` when preparing the dataset for CRM.Pad.

This parameter is disabled by default. Enable it to improve performance when generating the dataset.

**Default:** Disabled

**Type:** Boolean

### Sync.SkipChildInfoAreasAndFieldRights

**Description:** Use this parameter to exclude child `InfoAreas` when preparing the dataset for CRM.Pad and to stop checking for fields rights.

This parameter is disabled by default. Enable it to improve performance when generating the dataset.

**Default:** Disabled

**Type:** Boolean

## Languages and Texts

Define your language preference and custom texts for Aurea CRM.Web UI.

You can manage texts for Aurea CRM.Web UI elements (field labels, menu items, messages etc.) for all available languages in CRM.Designer, as well as define login languages for Aurea CRM.Web.

# Languages

Set your preferred language on Language setting page.

**Fastpath:** Main Page: Texts > Languages

Languages

**Note:** Not all languages listed on CRM.Designer's **Language Settings** page may be available for your service pack and vertical. For a list of available languages per product and vertical, see <https://support.aurea.com>.

## Global Language Settings

Use the **Global Language Settings** page to define languages available in the CRM.Designer database.

You can use the **Global Language Settings** page to define which languages are available in the CRM.Designer database based on the corresponding update.CRM language ID (**CRM Language**).

The settings specified here are saved in the "Language Default" configuration. To specify configuration-specific settings, use the **Language Settings** page (**Save in configuration** flag). For further details, see [Language Settings](#) on page 174.

Global Language Settings <span style="float: right;">✔ Save</span>					
Id	Language	Active	CRM Language	CRM Sub-Language	Parent Language
✗ 1	German	<input checked="" type="checkbox"/>	0	0	▼
✗ 2	English	<input checked="" type="checkbox"/>	1	0	▼
✗ 3	Czech	<input type="checkbox"/>	8	0	▼
✗ 4	Dutch	<input type="checkbox"/>	5	0	▼
✗ 5	Spanish	<input type="checkbox"/>	3	0	▼
✗ 6	Italian	<input type="checkbox"/>	7	0	▼
...					
✗ 15	Japanese	<input type="checkbox"/>	23	0	▼
✗ 16	Customized_English	<input checked="" type="checkbox"/>	1	100	English ▼
✗ 17	UK English	<input checked="" type="checkbox"/>	1	101	Customized_... ▼
✗ 18	Australian English	<input checked="" type="checkbox"/>	1	102	Customized_... ▼
+	New language name	<input checked="" type="checkbox"/>			▼

To define a language in CRM.Designer:

1. On the CRM.Designer main page click **Languages > Global Language Settings**.
2. Enter a name and click + (**Add**).
3. Check **Active** for active languages. This setting can be changed in sub-configurations (**Language Settings** page).

When downloading a configuration using CRM.bulkloader, the language selection dialog reflects this setting, i.e. active languages are checked by default in the **Available Languages** list. For further details, see [Downloading Configuration Data from the CRM.Designer Database](#) on page 761.

Languages flagged as inactive in the **Global Language Settings** are also deactivated as login languages. For further details, see [Login Page Languages](#) on page 175.

4. Enter the Aurea CRM language ID in the **CRM Language** field. For a list of valid Aurea CRM language IDs. For further details, see [Language IDs](#) in the *CRM.Core Administrator Guide*.
5. If you use language variants ("virtual languages") you need to define your sub-languages here as well: Enter the **Language No.** from the **Language** info area in the **CRM Sub-Language** field. For further details, see [Language](#) on page 52.
6. For deeper hierarchies, select the **Parent Language** for each sub-language.

---

**Note:** The limit of 16 hierarchy levels does not apply to CRM.Designer.

---

**Fallback scenarios:** If a text does not exist for a sub-language, the text from the parent language is used. If a text does not exist in any parent language, the default text is used.

7. Click **Save**.

---

**Note:** If cyclic dependencies have been defined, an error message is displayed and the changes are not saved.

---

Creating a language in **Global Language Settings** automatically creates it as a login page language. For further details, see [Login Page Languages](#) on page 175.

If you delete an existing language definition, all texts of that languages are deleted as well.

## Language Settings

You can store translations of texts in as many available languages as you want in your CRM.Designer database.

You can store translations of texts in as many languages as you want in your CRM.Designer database.

On the **Language Settings** page you can change the settings for your configuration:

Language	Editable	Active	Visible	Save in Configuration	Configuration
German	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Language Default
English	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Language Default
Czech	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Language Default
Dutch	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Language Default
Spanish	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Language Default
Italian	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Language Default
Danish	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Language Default
French	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Language Default
Greek	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Language Default
Romanian	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Language Default
Slovak	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Language Default
Polish	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Language Default
Hungarian	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Language Default
Japanese	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Language Default
Bulgarian	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Language Default
Chinese	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Language Default
Croatian	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Language Default
Finnish	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Language Default
Korean	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Language Default
Norwegian	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Language Default
Portuguese	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Language Default
Russian	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Language Default
Serbian	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Language Default
Slovenian	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Language Default
Swedish	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Language Default
Turkish	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Language Default

Check the **Save in Configuration** check box to define whether you want to save the settings for that language in your configuration or inherit the settings from its parent configuration. The settings of the "Language Default" are applied if nothing is defined. For further details, see [Global Language Settings](#) on page 173.

If **Save in Configuration** is checked, you can change three options for a language:

- **Editable:** If unchecked, you cannot define, change or delete translations for this language.
- **Active:** Checked for active languages in this configuration. When downloading the configuration or a package from it via CRM.bulkloader, this flag is exported as well. This ensures that inactive languages remains inactive after import (i.e. configuration/package upload).

---

**Note:** To globally define which languages are active, use the **Active** flag on the **Global Language Settings** page.

---

- **Visible:** If unchecked, the language is hidden from the configuration. You can only see it in the language settings.

## Login Page Languages

Here you can define the language choice for users on the login page.

Define which languages users can choose from the login page:

CRM Language	CRM Sub-Language	Language	Active	
0	1	German	Yes	
1	2	English	Yes	
3	0	Spanish	Yes	
7	0	Italian	Yes	

Make sure you define the login page languages in the configuration that is assigned to the "DefaultUser".

To activate a language, check the **Active** box. This is the same flag as in the global language settings. For further details, see [Global Language Settings](#) on page 173.

The **Description** is the text that is displayed in the **Language** drop-down list on the login page. The **Display Order Nr.** defines the sort order of the languages.

CRM Language-ID: 1 (0)

Active (global setting)

Description:

Display Order Nr.:

**Note:** If **Description** is undefined, the language is not displayed in the drop-down list on the login page.

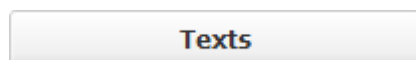
Check the `Login.EnableLoginInDefaultLanguage` Web Configuration parameter (in the default user's configuration), to additionally offer **My Default Language** as the first option in the **Language** drop-down list. For further details, see [Login.EnableLoginInDefaultLanguage](#).

## Texts

Here you can define texts for the various configuration units.

You can define texts (labels, table captions, texts in processes, messages etc.) for the various configuration units. For further details, see [Defining Texts for Configuration Units](#) on page 177. The texts are saved in the CRM.Designer database.

**Fastpath:** Main Page: Texts > Texts



For further details, see [Text Groups](#) on page 178 and [Editing Texts](#) on page 180.

You can add translations for all available languages to provide language-dependent process help and texts. For further details see [Languages](#) on page 173 and [Translating Texts](#) on page 183.



## Defining Texts for Configuration Units

Learn to define texts for configuration units.

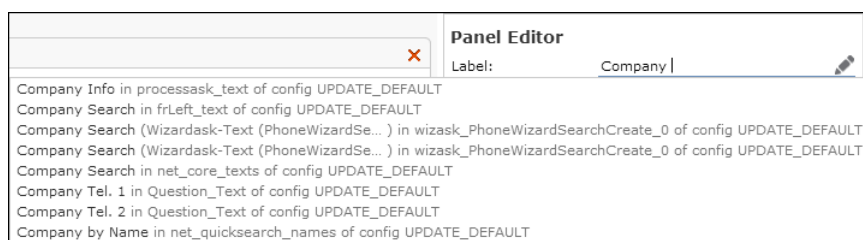
When defining a configuration unit, you need to define custom texts for UI elements e.g names for panels in a Details control, menu entries, labels on buttons, help texts and messages in processes etc.

The text fields access the texts stored in the CRM.Designer database.

To define a text:

1. Enter the desired text in the text field (e.g. **Label**).

Most text fields offer a full-text "find as you type" search to search for existing texts in all text groups.

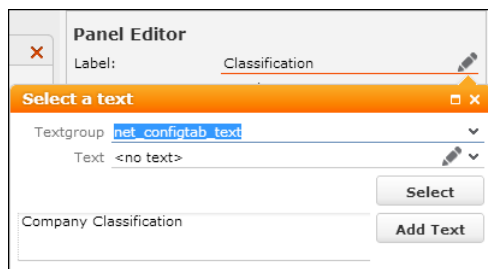


**Note:** Text fields in CRM.Designer and the text search drop-down list always display the language-independent **Default Text**. Therefore you can only search for texts defined as **Default Text**. For further details, see [Editing Texts](#) on page 180.

The drop-down list displays the meaning, text group and configuration of the text. The search results are sorted by configuration (current configuration first).

2. If available, select the desired text. Reusing existing texts helps reducing translation costs for your project.

If the desired text does not exist, click  (**Edit text**).



Your text is automatically copied to the field beside **Add Text**.

You can edit your text here. Press Enter to insert line breaks (e.g. when defining table captions).

3. Select another **Textgroup** if applicable. Text groups that are part of the current configuration are displayed in bold in the drop-down list. The selected text group is remembered per context (form, button, field group etc.) in the browser cache.

If you are adding a text to a text group that does not yet exist in your configuration, a new text group is created in your configuration. Naming: Current user name (or configuration name for non-user-specific configuration units) plus parent text group name.

---

**Note:** The text group name is limited to 50 characters. If the newly created text group's name exceeds that amount, you need to select another text group or add the text to a text group already present in your configuration.

---

4. Click **Add Text**.

The text is added to the selected text group.

To edit the text (or to add translations immediately), click  (**Edit text**) beside the **Text** field. For further details, see [Editing Texts](#) on page 180.

5. Click **Select**.
6. Save the configuration unit.

## Referencing Texts in JavaScript

Learn to reference texts in javascript.

Use the following syntax to reference texts in JavaScript (e.g. for multilingual text metrics in processes):

```
$t(TXT_<textgroup>_<number>)  
<textgroup> ...the name of the text group  
<number> ...the text's number (Nr column)
```

## Text Groups

Learn to manage texts in CRM.Designer.

Texts are organized in text groups.

On the CRM.Designer main page click **Texts** to display the **Text Groups** page.

The screenshot shows the 'Text Groups' interface. At the top, there is a header with the 'U' logo and the title 'Text Groups'. Below the header, there are navigation links: 'BB | UPDATE\_DEFAULT | customized\_parent\_config | my\_config'. The main content area has a 'Startpage' tab and three buttons: 'Create a New Text Group', 'Search', and 'Languages'. There is a checkbox for 'Show Descriptions' and a search input field with a magnifying glass icon. The list of text groups is organized into three sections: 'my\_config (15)', 'customized\_parent\_config (3)', and 'UPDATE\_DEFAULT (230)'. Each section contains a grid of text group cards, each with a name and a small icon.

All text groups are displayed listing their name, whether the text group applies only to a certain vertical, and for which (top-level) configuration this text group is valid. Check **Show Descriptions** to display a description for the contents of the text group if available.

- Click on a text group to display all strings defined for that group. For further details, see [Editing Texts](#) on page 180.

The screenshot shows the 'Textgroup 'ProcessHelp'' interface. At the top, there is a header with the 'U' logo and the title 'Textgroup 'ProcessHelp''. Below the header, there are navigation links: 'BB (Business to business)' and 'UserConfig\_SUSANNA\_Sub0 (1033) ->UPDATE\_DEFAULT (0)'. The main content area has a 'Startpage' tab and a 'Text Groups' sub-tab. There are links for 'Edit Textgroup Properties', 'Add a new Text to this group', and 'Delete this group'. There is a dropdown menu for 'Edit texts for this language:' set to 'German'. Below this is a table with columns: 'Id', 'Nr Default Text', 'Meaning', and 'Configuration'.

Id	Nr Default Text	Meaning	Configuration
9075	0	This process allows you to create new activities for companies.	<a href="#">Delete</a> UserConfig_SUSANNA_Sub0
9172	1	Help text for process.	<a href="#">Delete</a> UserConfig_SUSANNA_Sub0
9173	2	Help text for phase 1.	<a href="#">Delete</a> UserConfig_SUSANNA_Sub0

If you own the text group you can edit its properties, add new texts, delete the text group and edit texts for a specific language via the links displayed on the page.

- Click **Create a New Text Group** to create custom text groups. The maximum amount of characters for a text group name is 50.
- Click **Search** to search for text strings in a specific language or for a text ID.

Search for

Contains text 
  Starts with text 
  By text id

Language

<u><a href="#">Id</a></u>	<u><a href="#">Text</a></u>	<u><a href="#">Configuration</a></u>	<u><a href="#">Text Id</a></u>
5917	Currency	UPDATE_DEFAULT	TXT_StatisticControllerText_17
1164	Base Currency	UPDATE_DEFAULT	TXT_Currency_0
1247	Base Currency	UPDATE_DEFAULT	TXT_net_core_texts_115
5883	Currency field	UPDATE_DEFAULT	TXT_StatisticPageText_20
5887	Currency-dependent	UPDATE_DEFAULT	TXT_StatisticPageText_24
2802	Display locked catalog values (except currency)	UPDATE_DEFAULT	TXT_net_configoptions_text_106

**Note:** The search is performed with regard to the current configuration hierarchy, taking copied texts into account.

- Click **Languages** to manage texts by language. For further details, see [Viewing all Strings for one Language](#) on page 183.

## Editing Texts

Learn to edit texts on CRM.Designer.

You can edit texts

- In the corresponding text field. For further details, see [Defining Texts for Configuration Units](#) on page 177.
- By selecting a text group.

**U** Textgroup 'ProcessHelp' BB (Business to business) UserConfig\_BUSANNA\_Sub0 (1033) ->UPDATE\_DEFAULT (0)

Startpage > Text Groups

[Edit Textgroup Properties](#)  
[Add a new Text to this group](#)  
[Delete this group](#)  
 Edit texts for this language:

<u><a href="#">Id</a></u>	<u><a href="#">Nr</a></u>	<u><a href="#">Default Text</a></u>	<u><a href="#">Meaning</a></u>	<u><a href="#">Configuration</a></u>
9075	0	This process allows you to create new activities for companies.	<input type="text" value="🔍"/>	<input type="button" value="Delete"/> UserConfig_BUSANNA_Sub0
9172	1	Help text for process.	<input type="text" value="🔍"/>	<input type="button" value="Delete"/> UserConfig_BUSANNA_Sub0
9173	2	Help text for phase 1.	<input type="text" value="🔍"/>	<input type="button" value="Delete"/> UserConfig_BUSANNA_Sub0

- By searching for a specific string.

Search for

Contains text 
  Starts with text 
  By text id

Language

<u><a href="#">Id</a></u>	<u><a href="#">Text</a></u>	<u><a href="#">Configuration</a></u>	<u><a href="#">Text Id</a></u>
5917	Currency	UPDATE_DEFAULT	TXT_StatisticControllerText_17
1164	Base Currency	UPDATE_DEFAULT	TXT_Currency_0
1247	Base Currency	UPDATE_DEFAULT	TXT_net_core_texts_115
5883	Currency field	UPDATE_DEFAULT	TXT_StatisticPageText_20
5887	Currency-dependent	UPDATE_DEFAULT	TXT_StatisticPageText_24
2802	Display locked catalog values (except currency)	UPDATE_DEFAULT	TXT_net_configoptions_text_106

You can only edit texts that belong to the current configuration.

To edit a text:

1. Click on its **Id**. Click **Copy Text** if the text does not already belong to your configuration.

**U Edit text no. 9075 for all languages** BB (Business to business) UserConfig\_ **SUBANNA\_Sub0** (1033) ->UPDATE\_DEFAULT (0)

Startpage > Text Groups > Textgroup 'ProcessHelp'

**TextId:** 9075  
**Meaning:**  
**Default Text:** This process allows you to create new activities for companies. Change Delete

Used in the following Textgroup(s):

GroupId	Index	Name
694	0	ProcessHelp

LangId	Language	Text	Is Default
1	German	This process allows you to create new activities for companies.	<input type="checkbox"/>
2	English	This process allows you to create new activities for companies.	<input type="checkbox"/>
3	Czech	This process allows you to create new activities for companies.	<input type="checkbox"/>
4	Dutch	This process allows you to create new activities for companies.	<input type="checkbox"/>

You can modify the **Default Text** and all translations.

The **Default Text** is used for:

- displaying texts in CRM.Designer
- languages for which no translation exist (**Is Default** = checked)

---

**Note:** When creating a new text you must always define the **Default Text**.

---

2. Click the **Change** button beside the default text to modify the default text:

**TextId:** 9075

**Meaning:**

**Default Text:**

**Change Default Text:**  
 You can change the text, which is used when no language text is defined.  
**This change will be applied to all configurations and verticals!**

Used in the following Textgroup(s):

**Group** 694

**LangId**

1

**Meaning:**

2

3

Sealed

4

5 Save Cancel

3. Enter the desired text.
4. Use the **Meaning** field to provide e.g. context information for clarification when a translation agency has to work with these texts. For further details, see [Translating Texts](#) on page 183.
5. Enable **Sealed** to prevent a text from being translated; sealed texts are ignored by CRM.translation tool, i.e. they not exported and not updated on import. For further details, see [CRM.translation tool](#) in the *CRM.Core Administrator Guide*.

6. Click **Save**.
7. To edit language-specific texts (i.e. a translation), click **Change** beside the desired language.

**Change Text (German):**

**Default Text:**  
This process allows you to create new activities for companies.

**Info:**  
You can change the text for one language.  
**This change will be applied to all configurations and verticals!**

Mit diesem Prozess können Sie neue Aktivitäten für Firmen anlegen.

Save Cancel

8. To delete texts, click **Delete** beside the desired entry.

---

**Note:** For texts from a parent configuration you cannot delete the default text but only individual translations.

---

If you click the **Delete** button next to a specific language, the corresponding text is deleted and the default text is used instead.

Before a text is actually deleted, you get a confirmation dialog to avoid accidental deletions:

**Delete Text for Language German**

Text for this Language will be deleted. On further references the default text will be used.  
**This change will be applied to all configurations and verticals!**

**TextId:** 9075  
**Text:** Mit diesem Prozess können Sie neue Aktivitäten für Firmen anlegen.  
**Configuration:** UserConfig\_SUSANNA\_Sub0

Delete Cancel

## Placeholders in Texts

Learn to use placeholders in texts.

You can use placeholders in texts. For example: enter "Save {FI}" to get "Save Company".

For details, see the article "How to work with place holder in texts" at <https://support.aurea.com>.

## Translating Texts

Learn to manage the translation of customized texts.

Use **CRM.translation tool** to manage the translation of customized texts. You can export untranslated catalog values and texts from both the Aurea CRM and CRM.Designer databases. Once the texts have been translated, you can import them into the same or another database. For further information, see [CRM.translation tool](#) in the *CRM.Core Administrator Guide*.

## Logging the Usage of CRM.Designer Texts

Learn to log the usage of CRM.Designer Texts.

When translating CRM.Designer texts, it is helpful to know which texts are actually used in the customer's project and which texts are never displayed. Thus you can reduce translation effort and costs by only submitting used texts for translation.

To log the actual usage of CRM.Designer texts:

1. Add `<TextUsageLogger>true</TextUsageLogger>` to Aurea CRM.Web's `settings.xml` file (section `<update.web>`).

This logs each access to Aurea CRM.Web's text cache. The information is written to the `PORTAL_APPLICATION_TEXTUSAGE` table in the CRM.Designer database.

The `nr` column of `PORTAL_APPLICATION_TEXTUSAGE` contains the text ID. If `nr` contains 'NULL', the entire text group (`textgroupid` column) is used. Unused text groups are not listed.

**CRM.translation tool** exports this information along with the texts (CSV output only): The number of times the text was requested by Aurea CRM.Web is written to the `UsageWebInstallation` column of the output file.

---

**Note:** Turning on text logging slows down the application! Therefore, only use this setting until you have gathered enough data to be able to assume that all parts of your customized Aurea CRM.Web have been accessed at least once.

---

---

**Note:** Do not enable text logging on an Aurea CRM.Web offline client. Since no CRM.Designer database is present on Aurea CRM.Web offline clients, activating this setting makes Aurea CRM.Web offline unusable.

---

## Viewing all Strings for one Language

Learn to view all Strings for one Language.

Click **Languages** on the **Text Groups** page to view texts per language.

LangId	by Textgroup	Single Texts	Analysis	Untranslated
1	<a href="#">German by textgroup</a>	<a href="#">German single texts</a>	<a href="#">Analyse German</a>	<a href="#">Untranslated German texts</a>
2	<a href="#">English by textgroup</a>	<a href="#">English single texts</a>	<a href="#">Analyse English</a>	<a href="#">Untranslated English texts</a>
3	<a href="#">Czech by textgroup</a>	<a href="#">Czech single texts</a>	<a href="#">Analyse Czech</a>	<a href="#">Untranslated Czech texts</a>
4	<a href="#">Dutch by textgroup</a>	<a href="#">Dutch single texts</a>	<a href="#">Analyse Dutch</a>	<a href="#">Untranslated Dutch texts</a>
5	<a href="#">Spanish by textgroup</a>	<a href="#">Spanish single texts</a>	<a href="#">Analyse Spanish</a>	<a href="#">Untranslated Spanish texts</a>

All available languages are listed.

- **by Textgroup:** All text groups are listed displaying information on how many strings have already been translated to the selected language. Selecting a text group displays all texts:

Id	Nr	German (Make all active)	Default	Default Text / Meaning
9075	0	Mit diesem Prozess können Sie neue Aktivitäten für Firmen	<input type="checkbox"/>	<b>This process allows you to create new activities for companies.</b> Meaning:
9172	1	Help text for process.	<input checked="" type="checkbox"/>	<b>Help text for process.</b> Meaning:
9173	2	Help text for phase 1.	<input checked="" type="checkbox"/>	<b>Help text for phase 1.</b> Meaning:
9174	3	Help text for Phase 2.	<input checked="" type="checkbox"/>	<b>Help text for Phase 2.</b> Meaning:
9175	4	Help text for process metrics (Text).	<input checked="" type="checkbox"/>	<b>Help text for process metrics (Text).</b> Meaning:
9176	5	Help text for process metrics (TrafficLight).	<input checked="" type="checkbox"/>	<b>Help text for process metrics (TrafficLight).</b> Meaning:
9181	6	Help text for phase metrics (TrafficLight).	<input checked="" type="checkbox"/>	<b>Help text for phase metrics (TrafficLight).</b> Meaning:
9182	7	Help text for step.	<input checked="" type="checkbox"/>	<b>Help text for step.</b> Meaning:
9203	8	Create an opportunity and/or add opportunity participants.	<input checked="" type="checkbox"/>	<b>Create an opportunity and/or add opportunity participants.</b> Meaning:
9204	9	The name of the current rep.	<input checked="" type="checkbox"/>	<b>The name of the current rep.</b> Meaning:

Save changes

Strings that are already translated are editable. Untranslated strings are displayed read-only. Uncheck Default to enter a translation. Uncheck all **Default** check boxes by clicking the **Make all active** link and then enter the translations. Check Default to delete a translation.

- **Single Texts:** Lists all texts (from all text groups) for which a translation to the selected language exists. Click on a **TextId** to edit a text. For further details, see [Editing Texts](#) on page 180.
- **Analysis:** Displays a statistical overview of texts and words existing for that language.
- **Untranslated:** Lists all untranslated strings for the selected language.



## Uploading and Downloading Texts

Learn how to upload and download texts.

This section describes the bulk download and upload of texts. These concepts are useful to understand how to change texts and translations to ensure that they are correctly transferred using XML files.

There are two types of uploads:

- **Entire uploads of a configuration:** Text groups are added and modified, texts are added, modified and deleted, and also translations are added.

---

**Note:** It is strongly recommended that at least the two standard languages (English and German) are included in upload packs.

---

- **Language packs:** Only contain one language. Language packs only add, modify or delete translated texts. A text that is not included in the destination database but in the language pack is not added with the upload.

---

**Note:** update recommends using **CRM.translation tool** (instead of CRM.bulkloader) to manage the translation of texts in the CRM.Designer database. For further details, see [Translating Texts](#) on page 183.

---

The text owner defines which texts are up- or downloaded by supplying the configuration ID of the text group and the configuration ID of the text itself:

- All texts of a text group and the corresponding translations of a configuration is downloaded, even if another configuration has taken ownership of some texts and changed them.
- If you take ownership of a text belonging to the default configuration (UPDATE\_DEFAULT) and change it, downloading this configuration includes the changed text. When you upload this configuration to another CRM.Designer database, the changed text is owned by the default configuration.
- All texts owned by the configuration are downloaded and uploaded, even if it does not own the text group. When you upload such a configuration, it keeps ownership of the text in the destination database.
- A changed language translation is not owned by the changing configuration and therefore not downloaded and uploaded. But the translation is downloaded and uploaded by the configuration that owns the text.

Text group Ownership	Text Ownership	Lang. Trans Ownership	Transferred Configuration	Dest. Text Ownership	Dest. Transl. Ownership
A	A	A	A	A	A
A	B	B	A	A	A
A	A	B	A	A	A
A	A	A	B	-	-
A	B	B	B	B	B
A	A	B	B	-	-

(- = the text / translation is not transferred)

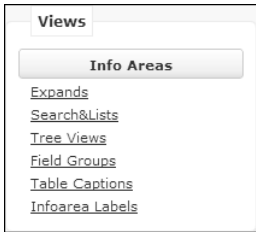
The ownership of a text in the destination database determines whether the text or language translation is transferred. A text or translation is not transferred when the text or translation is owned by a configuration with a higher ID in the destination database.

When you upload a configuration, language translations sometimes become invalid. They are deleted if the default text is changed. For example, you have a default text "Search Company" in your configuration with the German translation "Firma Suchen" assigned. Then you upload a configuration that changes the default text to "Company Search" but does not provide a German translation for this text, the German translation is deleted from the destination database. It is invalidated by the change of its default text.

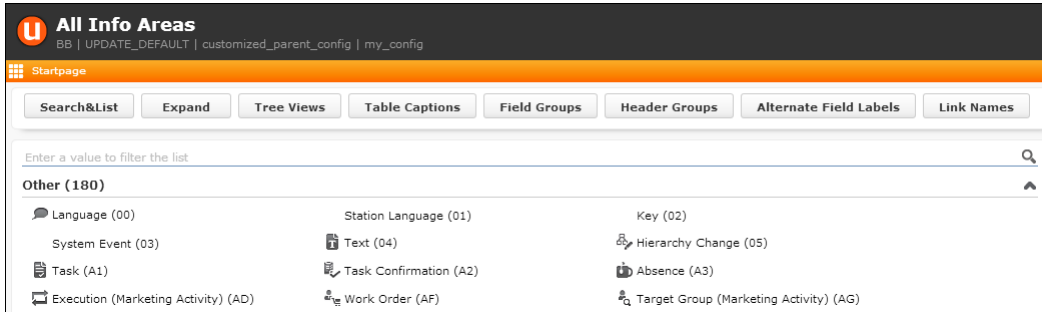
## Info Area Customizations

Learn to use the Views area of CRM.Designer to customize displays on Aurea CRM.Web.

Use the **Views** area of CRM.Designer to customize the way records are displayed in Aurea CRM.Web. Example - a record's Details control, a Search & List configuration etc.



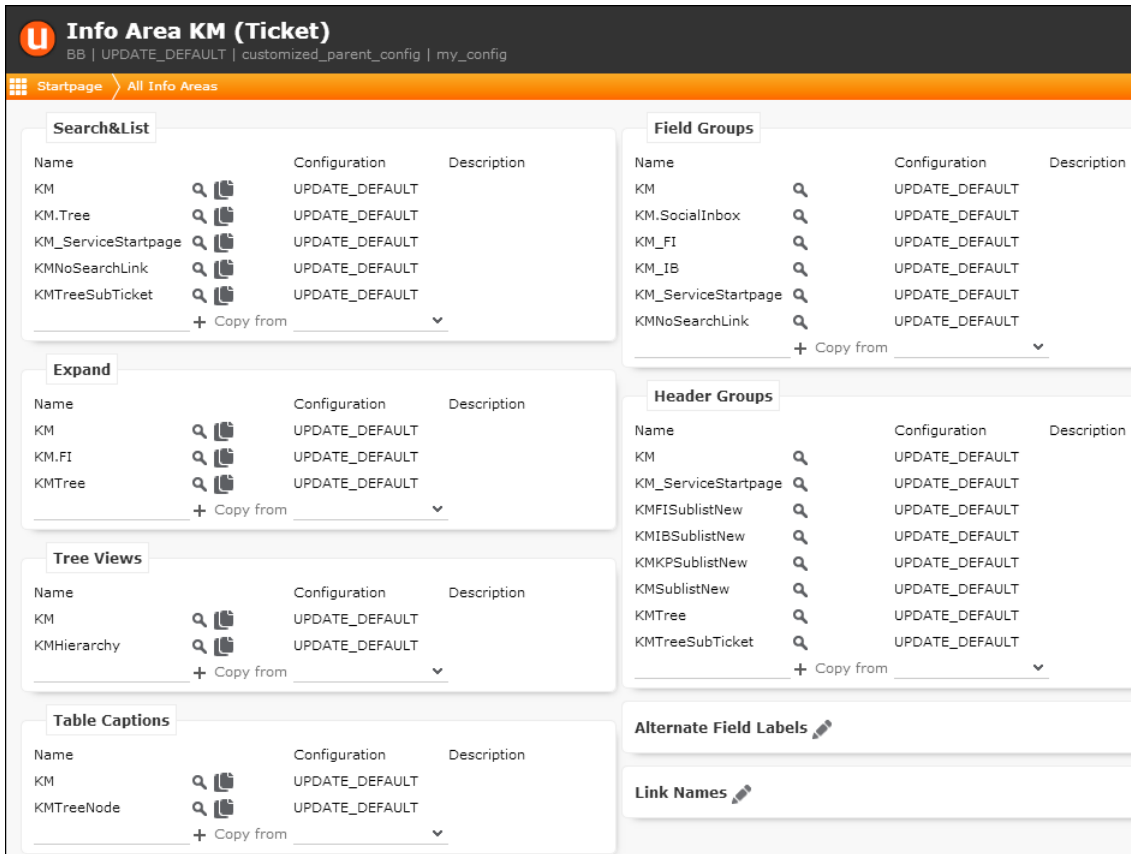
1. Click **Info Areas**.



2. Search for the info area you want to customize.

3. Click  (**Edit**).

All info area-specific configuration units are listed.



You can design the following configuration units:

- Search pages. For more information, see [Search&List Configuration](#) on page 365.
- Expand pages. For more information, see [Expand Configuration](#) on page 352.
- Tree view pages. For more information, see [TreeView Configuration](#) on page 381.
- Table captions used in headers, links, Tree views etc. For more information, see [Table Captions](#) on page 265.
- Field groups and controls defining list, search and details masks. For more information, see [Field Groups](#) on page 188.
- Headers. For more information, see [Headers](#) on page 321.
- Custom field labels. For more information, see [Alternate Field Labels](#) on page 270.
- Descriptive link names. For more information, see [Link Names](#) on page 274.
- Alternative info area names (only from the CRM.Designer main page and the **All Info Areas** page). For more information, see [Info Area Labels](#) on page 275.

For further details, see [Designing Configuration Units](#) on page 64

## Field Groups

The configuration of masks and lists is organized in **field groups**.

Field Groups		
Name		Configuration
EmailPhone	Q	UPDATE_DEFAULT
FI	Q	UPDATE_DEFAULT
FI.EmailPhone	Q	UPDATE_DEFAULT
FI.Report	Q	UPDATE_DEFAULT

A field group contains one or more **field controls**, i.e. the configurations of a details view, a list, a search mask etc. for the selected info area.

If you want to use your own field definitions, copy the control definition you want to modify to your configuration. For more information, see [Designing Configuration Units](#) on page 64.

For field groups you own, you can copy Details, List and Search controls from other configurations (**Copy from** drop-down list).

Field Controls for 'MyFI'			
Control	Table	Configuration	Description
Details	+ Company (FI)	Copy from	
List	+ Company (FI)	Copy empty control	
Search	+ Company (FI)	Copy EmailPhone in UPDATE_DEFAULT (0)	
Matchup	+ Company (FI)	FI in UPDATE_DEFAULT (0)	
MatchupList	+ Company (FI)	FI in update.tablet (15)	
MiniDetails	+ Company (FI)	FI in Sub_Emanuel (1001)	
Default	+ Company (FI)	FI in Sub_Joe (1004)	
		FI in Sub_David (1008)	
		FI in UserConfig_KV (1009)	
Description		FI in ArminE_SubConfig (1010)	
		FI in UNITTTEST_DEFAULT (1011)	
		FI in Sub_ChristK (1013)	
		FI in Sub_ChristK (1013)	
		FI in sv_config (1023)	
		FI in Marketing (1027)	
		FI in susi-config (1031)	

You can also extend Details and Default controls without the need to copy the parent definition to your configuration. For further details, see [Extending a Control](#) on page 206.

Field Controls for 'FI'			
Control	Table	Configuration	Description
Details	🔍 📄 Company (FI)	UPDATE_DEFAULT	📄 Extend
List	🔍 📄 Company (FI)	UPDATE_DEFAULT	
Search	🔍 📄 Company (FI)	UPDATE_DEFAULT	
Matchup	🔍 📄 Company (FI)	UPDATE_DEFAULT	
MatchupList	🔍 📄 Company (FI)	UPDATE_DEFAULT	
MiniDetails	🔍 📄 Company (FI)	UPDATE_DEFAULT	
Default	🔍 📄 Company (FI)	UPDATE_DEFAULT	📄 Extend
Description	Enter description here.		

Depending on the **field control type** a control definition may consist of the following: (For details, see [Field Control Types](#) on page 195)

- **fields**
- **sort fields** (e.g. in a List control)
- **panels** (e.g. in a Details control)
  - a panel may be displayed as a **tab** or **group**. For further details, see [Configuring Panels](#) on page 207.
  - panels have **panel attributes**. For further details, see [Panel Attributes](#) on page 208.
- general **attributes**. For further details, see [Control Attributes](#) on page 200.
- **column widths** for list controls. For further details, see [List control](#).
- **field attributes**. For further details, see [Field Attributes](#) on page 221.

## Special Field Groups Naming Conventions

Learn about the Special Field Groups Naming Conventions.

UPDATE\_DEFAULT contains the following special field groups:

- `D1.Upload/D2.Upload` (in the respective **Document** info area): Contains the default Details controls used for the **Upload Document** dialog for document fields. This is required in order to display the **Keyword** field on the **Upload Document** dialog. To change the **Upload Document** dialog either edit these controls or define new ones and use the Web Configuration parameters `Document.UploadFieldGroup.D1/D2`. For further details, see [Document.UploadFieldGroup.D1/D2](#).

These definitions are not used in the **Upload Document** dialog of the **Document** info areas themselves.

For best results, define `Columns=1` for the **Upload Document** dialog's Details control. For further details, see [columns](#) on page 201.

- `FI.Address/ID.Address/KP.Address` etc.: Contains the MiniDetails control used by the address popup in the Bing Map. For further details, see [Displaying Bing Maps in Related Areas and Forms](#) on page 826.
- `FI.EmailPhone/KP.EmailPhone/PE.EmailPhone`: The List control configures the results list of the **Phone & email Search**. The usage of these field groups is configured by the `Phone.EmailSearchListGroup` Web Configuration parameter. For further details, see [Phone.EmailSearchListGroups](#).
- `FI.MergeOverview/KP.MergeOverview/PE.MergeOverview/CP.MergeOverview`: Contains the Details control used for displaying the source and target records in the merge overview. For further details, see [Merging Companies and Persons](#) on page 852. These Details controls only support one column.
- `MA.EmailReps`: The Details control is used to determine the internal recipients when sending an appointment via email. For further details, see [email Reply](#) on page 839.
- `MA.ReplyFields / MB.ReplyFields`: The Details control is used to determine the copy fields for new MA and/or MB records created by the email Reply function. For further details, see [email Reply](#) on page 839.
- `OJ.MergeOverview`: Defines the merge overview for the property merge (BTB only). For further details, see [Merging Properties \(BTB only\)](#) on page 854.
- `Y2.BuyingCenter`: The Details control defines the items displayed on the Buying Center Wall. For further details, see [BuyingCenter](#) on page 399.
- `<InfoAreaId>.VisitPlanning`: The MiniDetails control defines the details displayed in the Visit Planning side bar (defined for FI/KP/MA/PE/SV). For further details, see [VisitPlanning](#) on page 479.

Some more field groups can be defined for special views or functions. Use the following naming conventions:

- `<InfoAreaId>.AddressCheck`: The Details and List controls are used for the External Address Check dialog. For further details, see [External Address Check](#) on page 840.

If undefined, the `upfields` defined in the MC entry (**External Address** category, **Parameters** option) are used as a fallback. If no `upfields` are defined, the fields actually updated in the record are displayed.

- `<InfoAreaId>.Cal`: The Details control is used for the popup view of calendar items (fallback: `<InfoAreaId>Cal`). `UPDATE_DEFAULT` contains `A1Cal`, `A3Cal`, `MACal` and `TDCal`.
- `<InfoAreaId>.NotifyReps`: The Details control is used to determine the internal recipients for the **Notify Rep** functionality. For further details, see [Notify Rep](#) on page 837.
- `<InfoAreaId>.Query`: Since it is not always possible to set a "deny access" right for fields that users do not need when defining lists, filters or queries, you can explicitly state which fields are available to users:

1. Define a field group called `<InfoAreaId>.Query`.
2. Add all desired fields to the List control of that field group.

In Aurea CRM.Web users see only those fields when defining lists, filters, query conditions and choosing query output fields for that info area.

- `<InfoAreaId>Questionnaire`: The Details control is displayed on top of the Survey page. For further details, see **Surveys** in the *Web Business Logic User Guide*. `UPDATE_DEFAULT` contains `F3Questionnaire`, `FIQuestionnaire` and `KPQuestionnaire`.
- `<InfoAreaId>.QuickView`: The Details control is used for the QuickView. For further details, see [QuickView](#).
- `<InfoAreaId>.Report`: Defines the fields to be output in exports and reports (Details and List control). You need to specify this alternative field group in the respective control definition. For further details, see [ExportFieldgroup](#) on page 201 and [ReportFieldgroup](#) on page 202.

**Example:** If you define `KP.Report` as `ExportFieldgroup` in the `KP` Details control, the fields defined in `KP.Report` are output when exporting from an Expand or Tree view of a person record.

---

**Note:** In lists this alternative field group is only used with the `Direct Export All` and `Direct XslReport All Excel` buttons. For further details, see [Export/Report Buttons](#) on page 310.


---

- `<InfoAreaId>.SerialEntry`: The List and MiniDetails controls define the items list and MiniDetails of a quick add page. For further details, see [SerialEntry](#) on page 454.
- `<InfoAreaId>.SerialEntry.Totals`: The List control defines the list of the totals popup (fallbacks: `<InfoAreaId>.SerialEntry` and `<ChildInfoAreaId>`).
- `<InfoAreaId>.SerialEntry.RelatedForm`: The Details control defines read-only details for the items of the quick add list.
- `<InfoAreaId>.Signature`: The List control defines the mapping for the signature parser and the contents of the **Copy from Signature or Website** dialog. For further details, see [Creating/Updating Records From email Signatures/Websites](#) on page 758.

## Customizing a Control

Learn how to customize controls.

To customize a control (e.g. the Details control from field group `FI`):

1. Navigate to the desired field group (info area `FI` > field group `FI`).
2. Open the definition of the desired control by clicking  (**Edit**).

Or, if the control belongs to a parent configuration:

- a. Click  (**View**) to view the control definition.

- b. Click  to copy the control definition to your configuration.



The screenshot displays the 'Field-Group 'FI', Control 'Details' editor. On the left is a search bar and a list of fields with their IDs and names. The central area shows several panels defined for the control, each with a list of fields and their IDs. The right-hand panel, titled 'Field "Company"', shows the configuration options for the selected field, including modes, attributes, and various field-specific settings like colspan, multi-line, and image options.

The Field Control Editor contains the following elements:

- **Search field:** Enter a field name or part of it or a field number to filter the field list or search for a specific field.
- **Field list:** All fields of the selected info area are listed. Fields that are currently included in the control are displayed **bold**. Clicking on such a field highlights all occurrences of the field in the control definition (i.e. the central part of the Field Control Editor).

Fields are displayed with their name, data type and unique ID. For further details, see [Field Names and IDs](#) on page 17 and [Data Types in Aurea CRM](#) on page 877.


At the end of the field list non-field elements (caption, separator etc.) and combined fields you defined are listed. For further details, see [Non-Field Elements](#) on page 210 and [Add Combined Field](#) on page 214.

- **Panels/Fields/Sort Fields:** Depending on the control (Details, MiniDetails, List etc.) all panels/fields/sort fields currently defined for this control are listed on

the central part of the screen. For further details, see [Configuring Panels](#) on page 207.


- **Attribute list:** Depending on the control and the selected element (panel, field, sort field) all available settings and attributes are displayed on the right part of the screen.

You can resize the three parts of the screen by dragging the border with the mouse.

3. Add, rearrange and remove fields using drag & drop. Use the Shift and Ctrl keys to select multiple fields.
  - Add fields by dragging them from the field list to the desired panel.
  - Rearranging/moving fields: Change the field order inside a panel or drag fields to another panel.
  - Use Ctrl+drag&drop to copy one or more fields from one panel to another. The field(s) are copied with all their attributes.
  - Remove fields from a panel by pressing Del or by dragging them to the field list on the left part of the screen.
  - You can modify the fields and sort fields defined for other controls (MiniDetails, List etc.) by drag & drop as well.
  - You can add additional elements like separators, free text etc. to your panels. For further details, see [Non-Field Elements](#) on page 210.
  - Click **Add Combined Field** to create "combined" fields e.g. displaying the address of a company or a person's full name (Details, MiniDetails only). For further details, see [Add Combined Field](#) on page 214.
  - Click **Add Linked Fields** to add fields from linked info areas. For further details, see [Add Linked Fields](#) on page 216.
4. Define additional settings for the control, the panel or field. For further details, see [Control Attributes](#) on page 200, [Configuring Panels](#) on page 207 and [Field Attributes](#) on page 221.
5. Click  **Save** (**Save**) to save your changes.

To discard your changes, click  **Delete** (**Delete**): If the parent configuration contains a control with the same name, your configuration reverts to this definition.

For Details controls changed by users inAurea CRM.Web, you can click

 **Delete Customization** to reset the control definition. For further details, see [Details Controls Extended by Users](#).

To view the differences between control definitions from different configurations on one screen, use the **Compare Field Controls** option in **Administration** area. For further details, see [Compare Field Controls](#) on page 746.

## Field Control Types

Learn about the field control types.

The following field control types are available:

### Details Control

Company: update software AG

Address: Operngasse 17-21  
1040 Wien, Austria

Country Code: 0043      Area Code: 01

Telephone: 87855      Fax:

E-mail 1:      E-mail 2:

Rep 1 ID: Lisa Simpson      Rep 2 ID: Simon Svendsen

Website:      Label:

Used to create the "mask" for the `ExpandNew`, the `Expand` and the `TreeView` page. For details on `Expand` and `TreeView` configurations and the related action templates. For further details, see [Expand](#) on page 578 and [Tree](#) on page 381.

For example - You can define for the panel layout which fields are displayed.

### List Control

Company	Lead Status	Country	Postal Code	City	Street
AkiComp	Interest	Austria	1040	Wien	Landgasse 7
BemComp	Customer	Austria	1100	Wien	Neustraße 1
Gabrieli	Customer	Italy		Venezia	
Gewo Baugenossenschaft	Lead	Germany	50935	Köln	Raumerstraße 20
Krist OY	Regular Custo...	Finland	00120	Helsinki	Mannerheimintie 1
Megabucks FC	Prospect	United King...	A92 Q2P	Bigsville	Harpers Lane 23
Merulo S.p.A.	Cold	Italy	43121	Parma	

The List control defines the fields used in the results list of a search. For details on the `Search&List` configuration and `Search` action template. For further details, see [Search](#).

- The **Fields** area contains the fields (columns) to be displayed in the list.

---

**Note:** A list control must always contain at least one field of the info area for which it is defined. Having only linked fields in a list results in an error in Aurea CRM.Web.

---

- Click on the area header to define the column widths:

Fields	Column Widths
Company Lead Status Country ZIP City Street	Column Width
	Company (C/2) 28
	Lead Status (X/38) 11
	Country (K/5) 14
	ZIP (C/7) 7
	City (C/8) 16
	Street (C/10) 24
	100%

Sort Fields
Company (C/2) ↑ Lead Status (X/38) ↓

If no column widths are entered, the percentages are calculated based on the field lengths in the data model.

Column widths are inherited from parent configurations. If fields are added to the child configuration, the required space is subtracted equally from all existing column widths.

- The **Sort Fields** area contains the fields used for sorting the records. List entries can be sorted either ascending or descending.

---

**Note:** When sorting by a time field, the corresponding date is always taken into account as well. You cannot sort by time independent of date.

---



---

**Note:** When a user sorts a list by clicking into a column header, the last sort criterion is appended as a secondary criterion to the new one. If more than one sort field is defined for a list, only the first one is appended as the secondary criterion.

---



---

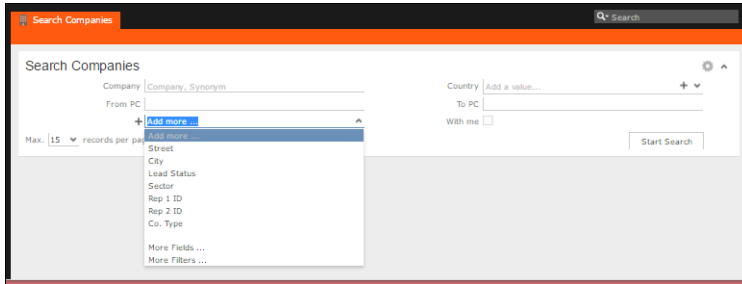
**Note:** If no sort criteria are applied to the data in the list (or sort criteria that produce more results than rows in the list), the order in which data is returned from the database is ambiguous. i.e. if a lot of records that are "equal concerning the sort criterion" exist, the database returns these in an arbitrary manner – thus, the results the database delivers may differ whether e.g. 10 or 50 results are displayed, and it is also possible that the same results are returned again when the user pages through the list. A feasible workaround when such problems occur is to define more specific sort fields.

---

## Search Control

The screenshot shows the 'Field-Group 'FI', Control 'Search'' configuration window. On the left, a search bar and a list of fields are visible. The list includes: 0/S - FI-StaNo, 1/L - FI-SerNo, 2/C - Company, 3/C - Synonym, 4/C - Division, 5/K - Country, 6/C - County, 7/C - Postal Code, and 8/C - City. Two panels, 'Panel-1' and 'Panel-2', are shown in the center. 'Panel-1' contains 'Company (2)', 'Country (5)', 'Postal Code (7)', and 'Postal Code (7)'. 'Panel-2' contains 'Street (10)', 'City (8)', 'Lead Status (38)', 'Sector (29)', 'Rep ID (24)', 'Rep 2 ID (26)', and 'Co. Type (28)'. On the right, the 'Field "Postal Code"' configuration is shown, including XML 'ZipCode', Field Nr. 7, Type C, Label: 'From postal code', Tool Tip, Function: '>=', and Field Attributes: 'Add Rep', 'Signature', 'Auto-Wildcard', 'All Columns', and 'Checkbox'.

The Search control defines the fields on a search mask. For details on the Search&List configuration and search action template. For further details, see [Search](#).



To define a Search control:

- Add the desired default search fields to panel No.1. These fields are shown as the default search mask.

---

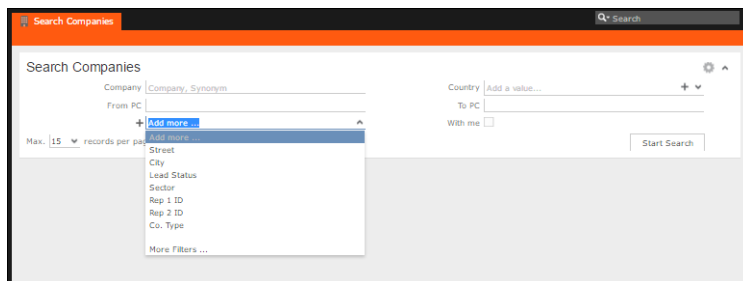
**Note:** Decoded fields (Z-fields) are not displayed on search masks.

---

- Any fields you add to a panel > 1 are displayed in the **Add more** drop-down list of the search mask.
- To search in several fields at once, use the `Enhanced Search Field` field attribute. For further details, see [Enhanced Search Field](#) on page 237.
- To hide the **With me** check box, add the `showOwnItems` control attribute and set it to `false`. For further details, see [Control Attributes](#) on page 200.

You can prevent specific users or groups from adding more fields to search from an info area. To do this:

- Define the search control as described in the steps above.
- Disable the **More Fields ...** option from the **Add more ...** drop down menu by setting the web configuration parameter `Search.HideMoreFields` to true. For more information, see `Search.HideMoreFieldsSearch.HideMoreFields`. The screenshot below shows the **More Fields ...** option is not longer available from the **Add more ...** drop down menu.



## Search Context

When linking records, users can select whether they want to search for records in the whole database or within the current context. For fields with a `Select-Function` this search context is determined as follows:

1. `SearchContext` as specified in the `Select-Function`. For further details, see [Syntax](#).
2. `SearchContext` as specified in the Search control's control attributes. For further details, see [SearchContext](#) on page 203.
3. If no `SearchContext` definition is found.
  - the source record is used
  - when linking records of the same info area, the search context is the source record's parent record (Example: linking a ticket to a ticket via the **Predecessor** field).

After deleting a link it is necessary to save the record to reset the search context.

### Matchup Control

If a match-up function is defined for an info area (i.e. a match-up button is configured in the respective `New` header, see [New](#)), data of a new record is compared with existing records to avoid duplicates. Examples in `UPDATE_DEFAULT: FI, KP, AU`.

The Matchup control defines which fields are compared when matching up records.

For information on further configuring the match-up functionality. For further details, see [Matchup Parameters](#) on page 134 and [PhoneticMatchupFieldnames](#) on page 202.

### MatchupList Control

The MatchupList control defines the list of similar records displayed on the Match-up dialog.

**Duplicates Found** □ ×

The following records already exist:

**Results (3 record(s))** << < 1 - 3 > >>

Company	Country	Postal Code	City	Street
ABC	Switzerland	1010	Wien	
ABC 123	Austria	2	Vienna	Refresh_Street
Abcde2	PVCS #70147	6546	Zürich	Abendstraße 12

<< < 1 - 3 > >>

**What would you like to do?**

- Go to existing record**  
The new record is not created.
- Complement existing record**  
Select this option to enter data from the new record in empty fields in the existing record. The existing record is retained and the new record is not added.
- Update existing record**  
Select this option to transfer all the data entered in the fields of the new record to the existing record; i.e. the existing fields are overwritten if new data is entered. The existing record is retained.
- Update new record with existing data**  
Select this option to copy all the fields in the existing record to the new record. The new record is stored.
- Complement new record with existing data**  
Select this option to enter data from the existing record in empty fields in the new record. The new record is stored.
- Create new record anyway**

Information on customizing the options displayed on the Match-up dialog, see the article at “How to Project Specific Match up Configuration” at <https://support.aurea.com>.

## MiniDetails Control

update software AG		Lead	Austria	1040	Wien	Operrgasse 17-21
Synonym update		Country Austria		Postal Code 1040		
City Wien		Co. Type		Sector		
E-mail 1		Telephone <a href="#">87855</a>		Fax		
Website		Rep 1 Lisa Simpson		Rep 2 Simon Svendsen		

The MiniDetails control provides a short read-only view without tabs, containing additional information for a record (e.g. for each list item, you can get information on additional fields by expanding the MiniDetails). If no MiniDetails are defined for an info area, the fields from the first tab of the Details control are displayed instead.

The MiniDetails view is available integrated into lists, in processes, in page headers etc.

**Note:** Link fields are automatically displayed as hyperlinks. Exception: If a link field cannot be resolved e.g. because it is not a direct link but one traversing multiple info areas, only the value but no link is displayed.

**Note:** Fields from linked info areas that are added to the MiniDetails using the **Add Linked Fields** function are not displayed as hyperlinks by default: Check the `Hyper-`

`link` field attribute to display such a field as a hyperlink. For further details, see [Add Linked Fields](#) on page 216 and [Hyperlink](#) on page 224.

## Default Control

In each field group you can create a Default control. The Default control contains all fields of an info area. This allows you to define default field attributes for all modes (Search, New, Update etc.) and for all fields of an info area, which are used in all controls that have no specific field attribute definitions for a field. For further details, see [Field Attributes](#) on page 221.

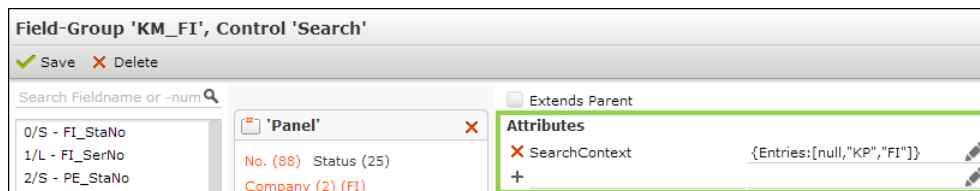
Even if a field is not displayed on a control, this can be helpful in case you change the content of the control in Aurea CRM.Web (e.g. the List control) and you want to provide some default behavior for a field.

## Control Attributes

Learn about Control Attributes.

To change the general attributes of a control, click on a blank part of the Field Control Editor or clear any previous selection by Ctrl+Click. The available settings are displayed on the right part of the screen:

- **Extends Parent:** You can extend Details and Default controls without the need to copy the parent definition to your configuration. For further details, see [Extending a Control](#) on page 206.
- **Attributes:** You can define additional properties for the entire control, e. g. the number columns and column widths.



To define a control attribute:

1. Enter the name (key) of the attribute.
2. Click + (**Add**).
3. Enter the value of the control attribute.

Or click (**Edit in Popup**), enter the value in the code box and click **Apply**.

4. Save the control definition.

To delete an attribute click (**Delete**).

## columnWidths

Defines the column widths.

Defines the column widths for this control. Enter the desired widths in px or as percentages separated by comma, e.g. "200px,15%,100px,150px".



---

You need to enter the widths for label and field separately, i.e. for a control with two columns you need to enter four values. For further details, see [columns](#) on page 201

Thus, "200px,15%,100px,150px" configures the following widths:

- 200px for the label of the first column
- 15% (of the total width) for the field of the first column
- 100px for the label of the second column
- 150px for the field of the second column

If you defined `columns = 3` and enter only 4 values in `columnWidths`, the widths for the last column are calculated automatically.

---

**Note:** Settings for individual fields (`Colspan`, `All Columns`, `No Label`) do not alter the width calculation, i.e. in the example above the first column would still have 200px + 15%.

---

## columns

Defines the default number of columns.

Defines the default number of columns for the current control (Details, Search and MiniDetails).

**Default = 2**

Label and text of a field count as one column.

You can define the `columns` attribute for the entire control and for specific panels. For further details, see [Panel Attributes](#) on page 208. The panel attribute overrules the control attribute.

## EditLinkedFields

Defines fields to make them editable.

In Details and List controls linked fields are read-only by default. For further details, see [Add Linked Fields](#) on page 216. Define `EditLinkedFields=true` to make them editable.

**Default = False**

---

**Note:** Use the `Readonly` field attribute to control which linked fields are actually editable/read-only, e.g. to make sure users cannot change a company name by editing the linked company field in a person list.

---

## ExportFieldgroup

Defined for exporting data.

You can specify an alternate field group to be used for exporting data from a list, tree or expand view of this field group.

## labelsAboveFields

Defined to make labels to be displayed above fields.

If set to `true`, field names are displayed above the fields in one column instead of displaying them in front of the fields. You can specify this attribute for an entire Details or Minidetails control or for specific panels of a Details control. For further details, see [Panel Attributes](#) on page 208.

---

**Note:** This attribute is mainly intended for single column layouts. However, the `All Columns`, `Colspan` and `Rowspan` field attributes are supported.

---

---

**Note:** If you set this for a widget, it is always displayed using the slim layout. For example, you can set it for the `ContactTimes` widget to get a slim layout display.

---

## PhoneticMatchupFieldnames

Defines fields used for phonetic match-up.

This attribute is available for the Matchup control only. For further details, see [Matchup control](#).

`PhoneticMatchupFieldnames` defines which fields are used for phonetic match-up, e.g. to match up O'Neal with O'Neill.

---

**Note:** The fields specified here must be part of the respective Matchup control.

---

---

**Note:** For the fields specified here a phonetic index must exist in the database. For further details, see [Index](#) in the *CRM.Core Administrator Guide* (index field property PHONETIC).

---

Specify the XML names of the fields to be used for the phonetic match-up (e.g. "LastName"), without quotes and separated by commas. If set to "\*", all fields that are part of a phonetic index are taken into account.

---

**Note:** The fields are searched in the order they are defined. The first field containing a value is used for the actual phonetic match-up.

---

## ReportFieldgroup

Defines an alternate fields group for creating reports.

You can specify an alternate field group to be used for creating reports for a list, tree or expand view of this field group. These results have more detailed information in a report than on screen where the additional information is usually only a click away.

## SearchContext

Defines a context-based search for linking records.

Use the `SearchContext` attribute to configure a context-based search for linking records. For further details, see [Search Context](#).

If a `SearchContext` is defined, users can select whether they want to search for records in the whole database or within the current context when linking records by using `↕`. Searching in the current context only returns results that are linked to a parent record of the current record (e.g. when linking a ticket to another ticket). Users get a drop-down list in the search mask where they can select the **Context** (the list entries using the default table caption of the corresponding record).

To configure the context-based search, define the `SearchContext` attribute for the Search control of the desired field group (i.e. the field group of the info area where you want to search for records). This definition is used as a default for all link fields to the respective info area. You can also define a search context within a specific `Select-Function` function. For further details, see [Syntax](#).

### Example

The search context defined for the **Ticket** info area (KM) `{items:[null,"KP","FI"]}` is used when linking a ticket to an activity (MA), to another ticket or an installed base (IB).

The following info areas are involved:

- `<sourceInfoArea>`: the info area of the details mask (e.g. MA,IB or KM)
- `<linkInfoArea>`: the info area used as the link record of the search (e.g. FI or KP)
- `<infoArea>`: the info area from which a record is selected (e.g. KM)

### Syntax

Define the `SearchContext` attribute using the JSON syntax (See <http://www.json.org>):

```
{items:[<item1>,<item2>,...,null]}
```

where

```
<itemX> = <linkInfoArea>
```

or (complex version offering more possibilities like link ID and alternate label)

```
<itemX> = {infoAreaId:<linkInfoArea>,  
linkId:<linkId>,sourceLinkId:<sourceLinkId>,  
sourceLinkName:<sourceLinkName>,label:<label>}
```

Specify `<itemX> = null` for searching the entire database.

The last entry provided in the `SearchContext` attribute is the one selected by default in the **Context** drop-down list of the Search mask.

## Available options:

<code>infoAreaId: &lt;linkInfoArea&gt;</code>	The ID of the info area used as the link record for the search (e.g. "FI").
<code>linkId: &lt;linkId&gt;</code> or <code>sourceLinkId: &lt;sourceLinkId&gt;</code>	The link ID used to read the link record (e.g. <code>{infoAreaId: "KP", linkId: 2}</code> ). You always need to specify the <code>infoAreaId</code> as well.
<code>sourceLinkName: &lt;sourceLinkName&gt;</code>	The link's name in the format <code>\$Link[&lt;infoAreaId&gt;:&lt;linkId&gt;]</code> , e.g. <code>"\$Link[FI:1]"</code>
<code>label: &lt;label&gt;</code>	Specifies an alternate label for the entry in the drop-down list, e.g. <code>{sourceLinkName: "\$Link[KP:2]", label: "Contact Person"}</code> for the second person link in KM. If not specified, the default table caption is used. You can reference (multilingual) texts from the CRM.Designer database by specifying <code>TXT_&lt;textgroup&gt;_&lt;number&gt;</code> , e.g. <code>{infoAreaId: "KP", label: "TXT_Admin_Text_5"}</code>
<code>parentInfoAreaId: &lt;parentInfoArea&gt;</code>	Use this option if there is no direct link in the context for the desired info area, but it can be retrieved as a parent link of the linked record. Example (s. below): Getting only marketing activities (AK) linked to the same campaign (CM).

**Note:** `sourceLinkId` and `sourceLinkName` can only be used if the argument `linkUid` (or `searchUid`) is passed to the search page.

Use the **SDK** page to test your definitions: **Search > Search Form**.

**Examples:**

```
{items: [null, "KP", "FI"]} ... entire database, person, and company context (default selection)
```

```
{items: ["FI", "KP", {infoAreaId: "KP", linkId: 2}, null]} ... company, person (default link), person (Link ID 2), and entire database (default selection)
```

```
{items: ["FI"]} ... the search context is fixed, i.e. no drop-down list is displayed: the search returns only records linked to the parent company
```

To show all companies/persons to which the ticket is linked (both **End Customer** and **Contact Company/Person**) in the search context:

```
{
items: [
null,
{sourceLinkName: "$Link[KP:1]},
{sourceLinkName: "$Link[KP:2]},
{sourceLinkName: "$Link[FI:1]},
{sourceLinkName: "$Link[FI:2]}
]
}
```

For the same result you can specify:

```
{
items: [
null,
{infoAreaId: "KP", sourceLinkId: 1},
{infoAreaId: "KP", sourceLinkId: 2},
{infoAreaId: "FI", sourceLinkId: 1},
{infoAreaId: "FI", sourceLinkId: 2}
]
}
```

To search only for marketing activities linked to the same campaign or the entire database (in UPDATE\_DEFAULT defined in the AN Details control, Select-Function of field **Follow-up Marketing Activity**):

```
{items: [{parentInfoAreaId: "AK", infoAreaId: "CM"}, null]}
```

**Search Marketing Activities**

Marketing Activity  + ▾ Level

Variant  + ▾ Resp. Rep ID

Camp. Name

+ Add more ... ▾ With me

max. 15 ▾ Records per page Context: **Fall 2013** ^

Search Results << < 1 - 3 of 3 > >>

Campaign	Marketing Activity	L...	Variant	Type	Date	Status
Fall 2013	Folder				25.06.2013	Scheduled
Fall 2013	Invitation				25.06.2013	Scheduled
Fall 2013	Promotional Activity BM...				25.06.2013	Scheduled

## showOwnItems

Defined to display 'With me' check box.

By default, Search controls for info areas that contain rep and/or participants fields display a **With me** check box, allowing users to quickly search for records where they are entered as rep or participant.

To hide the **With me** check box from a Search, define `showOwnItems = false`.

## SublistDefaultAction

Defines default action for sublist on double click.

If a list definition is used in a sub-list, you can specify the default action for a double-click on the elements of that list. For further details, see [Default Action](#) on page 289. This setting overrides the Web Configuration parameter `List.SubList.DefaultAction` for this control. For further details, see [List.SubList.DefaultAction](#).

## verticalTabbing

Defines tab order to set it to vertical.

If set to `true`, the tab order is set to vertical, i.e. top to bottom instead of the standard left to right tab order.

For Example - If a field contains a rep picker or record selector icon, tabbing activates these icons before proceeding to the next field below.

You can specify this attribute for an entire Details control or for specific panels. For further details, see [Panel Attributes](#) on page 208.

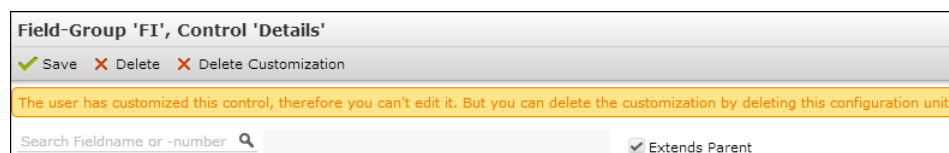
## Extending a Control

Learn about extending a control.

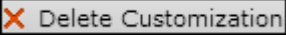
### Details Controls Extended by Users


Users can rearrange the tabs and groups of Details controls in Aurea CRM.Web using drag&drop. You can deactivate this behavior by unchecking the `Expand.PanelReorderingAllowed` Web Configuration parameter. For further details, see [Expand.PanelReorderingAllowed](#).

A Details control that has been changed by the user is **not** copied to the user's configuration: The control is only extended (**Extends Parent** checked plus message). Changes in the control's parent are therefore available to the user.



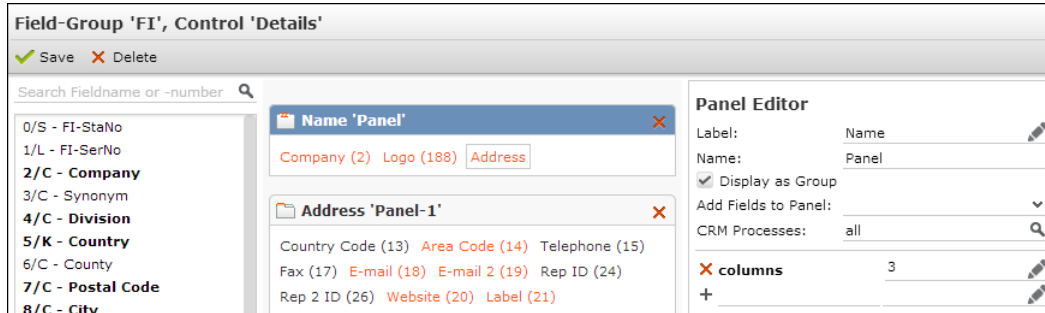
Such Details controls cannot be edited in CRM.Designer. Click [Delete Customization](#) to reset the control definition to its parent.

If the control was already customized before the user changed it, no message is displayed, but the  button is available.

To offer users the possibility to reset the layout of a Details control in Aurea CRM.Web, you can configure the `ResetFormLayout` button  into the respective `Expand` header. For further details, see [ResetFormLayout](#) on page 318.

## Configuring Panels

Learn to configure panels.



Fields displayed on Details controls are organized in panels displayed as tabs and groups. The Field Control Editor lists all panels defined for a control on the central part of the screen.

## Rearranging Panels

Learn to rearrange the order of panels.

Use drag&drop to change the order of panels in your control definition. Use the Ctrl key to move multiple panels.

To nest tabs/groups in a group:

1. Create a new (empty) panel.
2. Use drag&drop to add existing tabs/groups to this panel.

Nesting tabs/groups in tabs is not supported.

3. Use non-field elements (e.g. Captions, Texts) to further structure panels. For further details, see [Non-Field Elements](#) on page 210.

27.11.2012 17:40: Info for update Software AG

Menu | Edit | Save | Cancel | Recurrence... | Send as E-Mail | Copy | W | [Icons]

**Main data**

Activity Type **Info** Status **Scheduled**  
 Date 27.11.2012 End Date 27.11.2012  
 Time 17:40 End Time 19:40  
 Priority High Create To-Do   
 Result Private   
 Subject My Subject

**Subject**

Rep ID Test User Assigned by Rep ID  
 Participant ID Required Test User  
 Optional  
 For Information  
 Text

**Related Data**

**Details**

Duration 2  
 Satisfaction Indifferent  
 Unit Hours  
 Purpose Acquisition  
 Follow-up Date  
 Reminder Duration  
 Costs  
 Reminder Unit

**Links**

Property  
 Opportunity  
 Ticket  
 Installed Base  
 Previous/Main Activity

## Adding a New Panel

Learn to add a new panel to field control editor.

In the Field Control Editor click **Add Panel** to add a new panel. The new panel is added (as group) at the end of the panel list. Add fields using drag&drop and define the desired settings. For further details, see [Panel Attributes](#) on page 208.

## Deleting a Panel

Learn to delete a panel.

To delete a panel, click on **X (Delete)** in the header of the respective panel.

## Panel Attributes

Learn to edit the panel attributes.

**Panel Editor**

Label: Classification [Edit]

Name: Panel-3

Display as Group

Add Fields to Panel: [Dropdown]

CRM Processes: all [Search]

Modes:  Details (All)  New  Update

**X collapsed** true [Edit]

+ [Edit]



To change the settings of a panel, click on the panel's header. Use the Panel Editor on the right part of the screen to edit the panel's attributes:

- **Label:** The text displayed on the tab/as header of the group. The texts are saved in the `net_configtab_text` text group. For further details, see [Texts](#) on page 50. Group names are only visible if more than one group exists.
- **Name:** The name of the panel (i.e. the panel's ID).

In the panel list **Label** and **Name** are displayed in each panel's header.

- **Display as Group:** Checked for groups, unchecked for tabs. Check this option to create a group.

Nested panels are always displayed as groups.

In the panel list tabs are indicated by the , groups by the  icon.

- **Add Fields to Panel:** When creating a group from a tab, enter the panel to which the content of this tab is to be appended. The fields of the tab are appended at the end of the tab/group.

If you don't specify a panel, the fields are appended at the end of the first panel.

- **CRM Processes:** Assign one or more CRM processes to the panel. For further details, see [Roles & CRM Processes](#) on page 727.

You can specify additional attributes by clicking + (**Add**) and entering a name and value. Available attributes:

### collapsed

Defines the initial state of group panels in a Details control. If set to `true` the group is initially displayed in collapsed state. This setting is overruled by the user's settings.

This setting is ignored for tab panels but is applied as soon as the user changes a tab panel to a group panel via drag&drop.

### columns

Defines the default number of columns for the selected panel. Label and text of a field count as one column.

**Default = 2**

### labelsAboveFields

If set to `true`, field names are displayed above the fields in one column instead of displaying them in front of the fields.

### verticalTabbing

If set to `true`, the tab order is set to vertical (top to bottom) instead of the standard left to right tab order.

**Note:** You can define the `columns`, `labelsAboveFields` and `verticalTabbing` attributes for the entire control and for specific panels. The panel attribute overrules the control attribute. For further details, see [Control Attributes](#) on page 200

## Non-Field Elements

Learn to add non-fields elements to control definition.

You can structure the information displayed on Details, MiniDetails and Search controls by adding the following "non-field" elements to your control definitions:

- **Line Feed:** Inserts a line break as follows:
  - If the **Line Feed** element is inserted in another than the first column, the field after the line break is moved to the next row.
  - If inserted in the first column (i.e. at the beginning of a row), it is ignored.
  - If inserted with the `All columns` field attribute checked, an empty row (blank line) is created at this position. This functionality is not supported if the **Line feed** element is part of a combined field.
  - If used inside a combined field without additional attributes the line feed spans the same number of columns as the combined field itself **Default:** 1 column
- **Caption:** Inserts a caption text followed by a line break. The caption text is defined by the **Label** attribute. To display multiple captions in one row, use the `One Column` field attribute.
- **Text:** Inserts any kind of text. The text itself is defined by the **Label** attribute. If a text requires more than the available space, the entire text is displayed in a tool tip.
- **Separator:** Inserts a horizontal separation line across all columns.

---

Non-field elements are listed at the end of the Field list. You can add them to a panel just like fields (using drag&drop).

## Attributes

Learn about the attributes supported by non-fields element.

Non-field elements support the following attributes:

- [Hide Field](#) on page 224
- **Hide if predecessor/successor is empty:** These attributes hide a non-field element if the field before or after it is empty. Use this for combined fields to only show the comma in "1040 Vienna, Austria" if the **Country** field contains a value. For further details, see [Add Combined Field](#) on page 214.

---

**Note:** The **Hide if predecessor/successor is empty** attributes are only supported for combined fields (in read-only mode).

---

The **Line Feed** element also supports:

- [All Columns](#) on page 223 (details above)

The **Caption** and **Text** elements also support:

- **Label:** Define the text for the **Caption** or **Text** element here. The texts are saved in the `net_alternate_field_labels` text group. For further details, see [Texts](#) on page 50.
- [All Columns](#) on page 223
- [One Column](#) on page 226
- [Colspan](#) on page 228
- [Rowspan](#) on page 234

## Modes

Learn about edit and update modes.

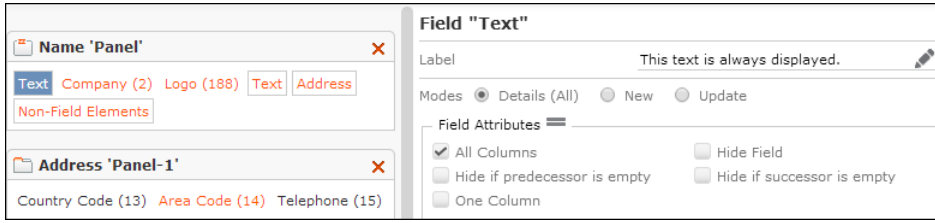
You can display non-field elements in Details controls depending on the mode (**Details (All)**, **New** or **Update**). For further details, see [Field Attributes](#) on page 221.

### Example New/Update Mode

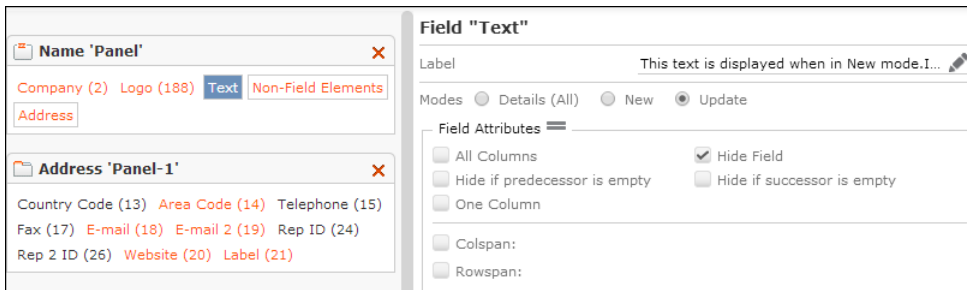
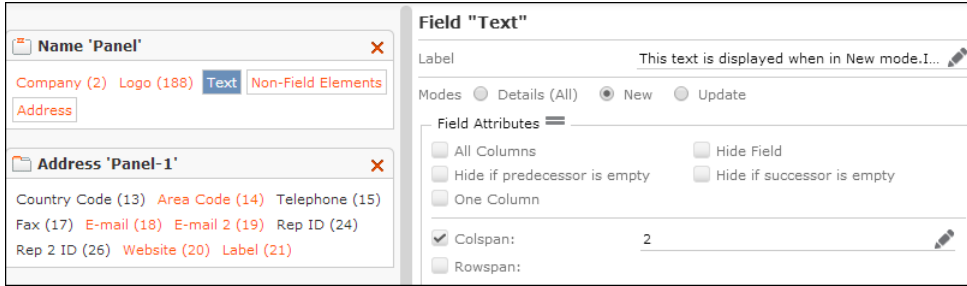
This allows you to provide different help texts for creating a new record and for updating a record.

For this example two text elements were added to an FI Details control.

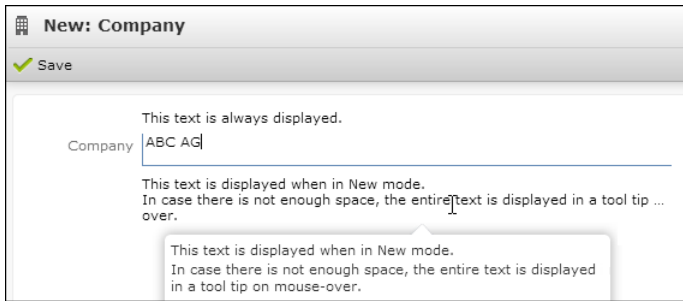
The first text is displayed in all modes:



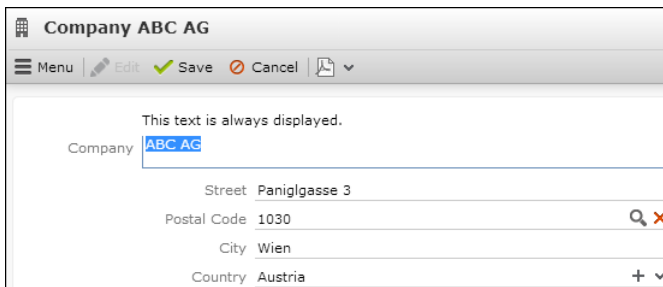
The second text is displayed in New mode and hidden in Update mode:



Result in New mode:



Result in Update mode:



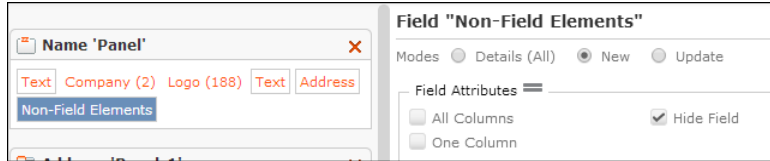
### Example Edit/View Mode

In addition, you can also display different non-field elements for Edit and View mode by using combined fields.

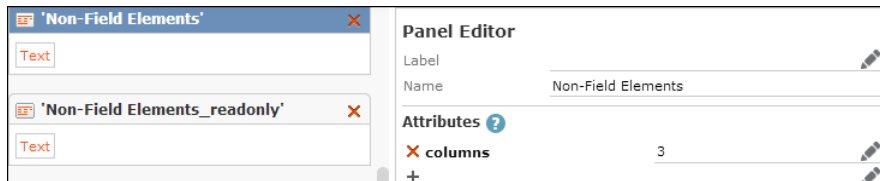
To display different non-field elements for Edit and View mode you can use a combined field and add different non-field elements to the Edit and to the ReadOnly panel. For further details, see [Add Combined Field](#) on page 214.

This example extends the FI Details control from the above example.

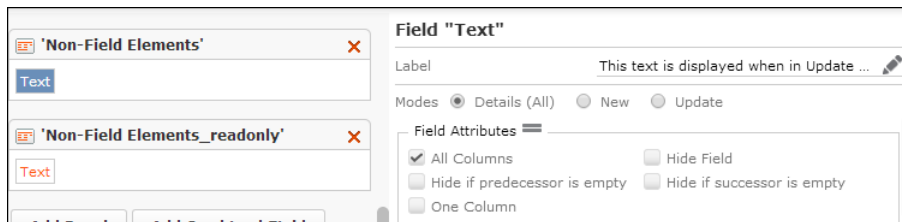
The combined field is displayed in Update mode and hidden in New mode:



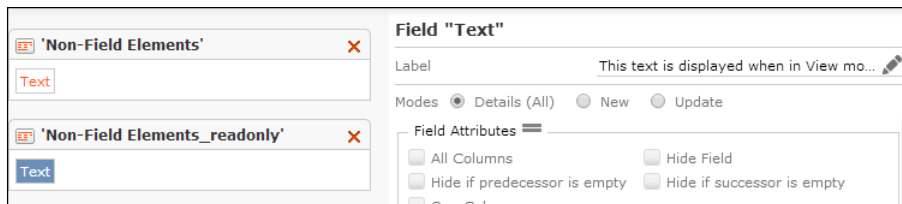
The Edit panel:



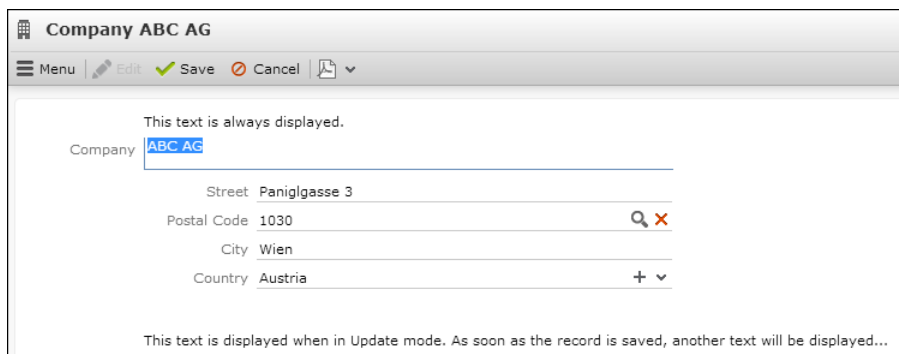
The text to be displayed when in Edit mode:



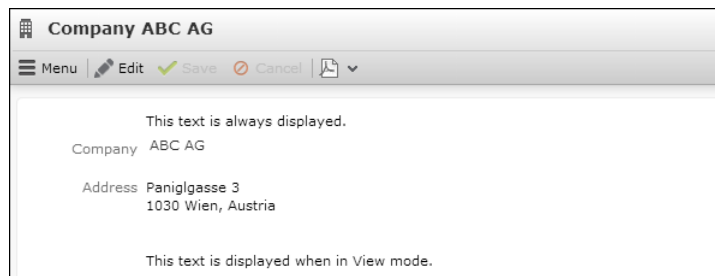
The ReadOnly panel with the text to be displayed when in View mode:



Result in Edit mode:



Result in View mode:



## Add Combined Field

Learn how to display content of multiple field as one.

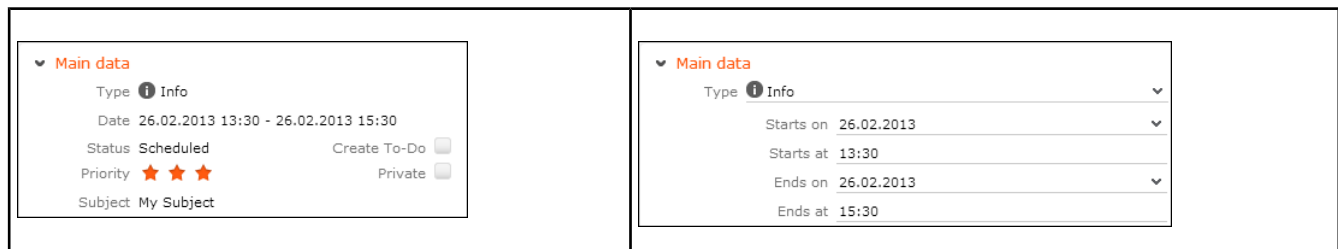
You can display the content of multiple fields in one "combined" field in order to display them together. For Example - a person's address or phone number displayed on a business card.

---

**Note:** Combined fields are available for Details and MiniDetails controls only.


---

Example: a customized MA Details control



To define combined fields:

1. In the Field Control Editor click **Add Combined Field**.
2. Enter a name (max. length: 23 characters).
3. Click **OK**.

This adds two new panels to the panel list: Panel "<name>" is used for the combined field in Edit mode, "<name>\_readonly" for the combined field in View mode. (Panels for combined fields are indicated by the  icon.) The combined field "<name>" is added to the fields list.

4. You can define a **Label** to be displayed before the combined field when in View mode. (In Edit mode the separate field labels are displayed instead.)

5. Add the fields you want to combine (e.g. **Starts on**, **Starts at**, **Ends on** and **Ends at**):

- For Details controls add fields to both the Edit and the Read-only versions of the combined field.
- For MiniDetails controls only the Read-only definition is used. (Panel "<name>" is ignored.)
- Add non-field elements to the Read-only definition to further structure the display when in View mode. For further details, see [Non-Field Elements](#) on page 210. Define **Text** elements containing a space, a slash, dash etc. to control the spacing of the combined field.



Use panel attributes like **Hide if successor is empty** to e.g. hide the dash if the activity record (in the example above) contains no **End Date/Time**:

Date 27.11.2012 17:40

---

**Note:** In Edit mode the Line Feed elements are removed; all fields from the combined field definition are listed in one column.

---



---

**Note:** If you add non-field elements that are not supported for combined fields (e.g. **Caption**) a warning is written to the client log.

---

- You can set field attributes for individual fields of a combined field definition. For further details, see [Field Attributes](#) on page 221. Please note that not all field attributes make sense in View mode and that combined fields in Edit mode are always displayed in one column.

---

**Note:** If all fields contained in a combined field are defined as read-only (via access rights definition or field attribute), the combined field is displayed as in View mode.

---

- For a combined field itself only a sub-set of field attributes is available (`All columns`, `Colspan` etc.)

6. The combined field is listed at the end of the field list. Add it to the desired panel(s) by using drag&drop.

---

**Note:** When creating a report for records from an info area where combined fields have been defined, the fields are output individually with their label and value, i.e. not in the formatting defined in the Details control.

---

## Add Linked Fields

Learn to add linked fields from related info areas.

You can add fields from related info areas (N:1- and 1:1-links) to a control (Example in UPDATE\_DEFAULT: the `KP` List control contains the **Company** field from FI).

---

**Note:** **Add Linked Fields** is available for the Details, List and Search controls, but not for Matchup or Default.

---

---

**Note:** Using linked fields may impact the performance. If applicable, use Z-fields instead.

---

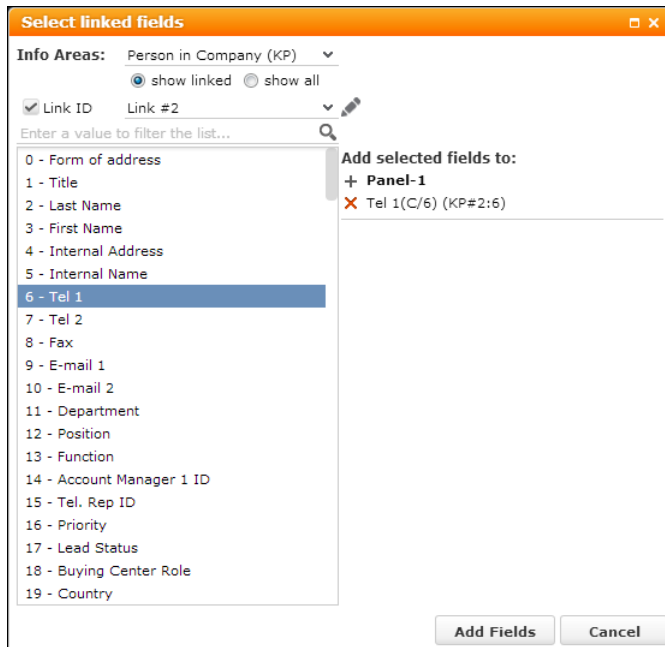
To prevent changes to a parent record, linked fields are read-only by default in List and Details controls. Define the control attribute `EditLinkedFields=true` to make linked fields editable. For further details, see [EditLinkedFields](#) on page 201. If you use this setting you can still use the `ReadOnly` field attribute to control which linked fields are actually editable/read-only.

Linked fields display "N/A" if they are empty due to a non-existent linked record (e.g. KP fields on an MA Details control if the MA record is linked to an FI record only). You can change this text by editing text no. 8 in text group `common_text`.

To add linked fields:

1. In the Field Control Editor click **Add Linked Fields**.





2. Select the info area from which you want to add a field.
  - **show linked:** All info areas are listed to which the current info area is linked via N:1- or 1:1-link.
  - **show all:** All available info areas are displayed in the **Info Areas** list. All fields of the selected info area are listed.
3. Check the **Link ID** check box, if you want to use another than the default link. Select the desired link. For further details, see [Link IDs](#) on page 19.
4. Enter a search string to filter the list of available fields.
5. Select one or more fields (Ctrl) from the list.
6. Under **Add selected fields to** click on + (**Add**) next to the panel to which you want to add the field(s).
 

Click **X (Delete)** to remove a field.
7. Click **Add Fields** to add the fields to the control definition, otherwise click **Cancel**.
8. Save the control definition.

### Syntax:

```
<Fieldname>(<type>/<fieldnumber>) (<InfoAreaId>#<link id if != default link>:<fieldnumber>)
```

### Examples:

Company (C/2) (FI) ...to show a **Person's Company** on the **KP** List control

Tel 1(C/6) (KP#2:6) ...to show the **Contact Person's** phone number on the **KM** Details control

## Edit Linked Fields

Learn to edit linked fields.

You can add fields from related info areas (N:1- and 1:1-links) to a control (Example in UPDATE\_DEFAULT: the `KP` List control contains the **Company** field from FI).

Linked Fields can be edited, but only under certain circumstances:

- The Linked field must be placed on the root panel. It does not work if it is created in some other panel.

The screenshot shows the 'FieldControl' configuration page for 'Field-Group 'FI', Control 'Details''. The interface includes a search bar for field names or numbers. On the left, a list of available fields is shown, including '2/C - Company', '4/C - Division', '5/K - Country', '7/C - Postal Code', and '8/C - City'. The main area displays three panels: 'Name 'Panel'' with 'Company (2)', 'Logo (188)', and 'Address'; 'Address 'Panel-1'' with 'Country Code (13)', 'Area Code (14)', 'Telephone (15)', 'Fax (17)', 'E-mail (18)', 'E-mail 2 (19)', 'Rep (24)', and 'Rep 2'; and 'Type 'Panel-2'' with 'Co. Type (28)', 'Division (4)', 'Line Feed', 'Sector (29)', and 'SIC1 (32)'.

- Ensure that the current info area already has value saved for it to be able to edit the linked fields. For example, if you want to edit linked fields for a company and you added linked fields to City, you must first ensure that you have created a company with a city and saved it.

## Input Widget for Array Fields

Learn to configure array fields defined in the data model.

In Details, List and MiniDetails controls you can configure array fields defined in the data model to be displayed using a `ListInput` widget. To access the widget, go to >> **Inputs** > **ListInput** on the **SDK** page.

This enables users to quickly enter data using the keyboard to navigate between the fields of the array.

The screenshot shows a configuration window for a ListInput widget. It has tabs for 'Address', 'Type', 'Classification', and 'Event'. Under the 'Type' tab, there are two array fields: 'Co. Type' with 'Head Office' selected and 'Add 4 more...' button, and 'Sector' with 'Services' and 'CRM' selected and 'Add 2 more...' button. Each field has a '+' and '-' icon for adding or removing items.

To configure the `ListInput` widget:

1. Add the first field of an array to the desired control (e.g. **Co. Type** and **Sector** in the `FI` Details control).

The array is displayed as one field with the label of the first field of the array. Each value is displayed in a small rectangle. A ghost text indicates the number of available fields in the array. In View mode the values are listed separated by comma.

---

**Note:** If you add another field or more field of the array, the default field widget is used (drop-down list etc.).

---

Supported field types: variable catalog, fixed catalog, text, rep input for **Participant** fields only - see below.

For fixed catalogs the image is displayed as well (if defined). The `Small Image Template` field attribute must be defined. For further details, see [Small Image Template](#) on page 234.

The **Participant** fields in the **Activity** info area use the `ListInput` widget by default. In this case, the number of possible values is not restricted. The separator is determined by the `Email.Separator` Web Configuration parameter. For further details, see [Email.Separator](#).

## Restricting the Number of Items in an Array Field

Learn how to restrict the number of items that can be defined in the field.

In some cases, you may want to restrict the number of items that can be defined in the field details mask of an array field. To define the maximum number of items that can be selected from an array field set the `arrayFieldMaxItems` value in the extended options for a `ListInput` widget.

The following sample configuration below restricts the number of fields in the array field to one item:

```
{ 'widgetOptions': { 'arrayFieldMaxItems': 1 } }
```

## HTML Fields

Learn about HTML fields in Aurea CRM.Web.

You can create HTML fields to store HTML-formatted text in the Aurea CRM database. This may be used to send HTML-formatted emails out of Aurea CRM.Web. For further details, see [Send As email](#) on page 835.

Create a field of type "html-text" e.g. in the activity info area (MA). For further details, see [Data Model](#) and [Aurea CRM Field Types](#) in the *CRM.Core Administrator Guide*.

HTML fields are supported for Details and MiniDetails controls. In MiniDetails the HTML-formatted text is displayed as in View mode. HTML fields are not supported in lists. Use the linked plain text field in List controls instead.

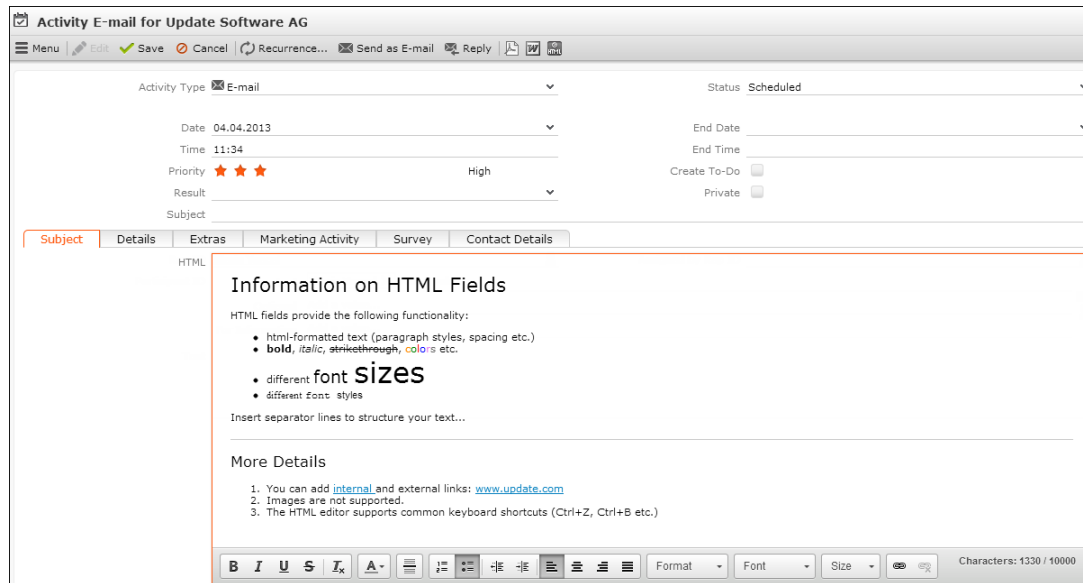
---

**Note:** Make sure you use the plain text field in search forms and as query parameters. Otherwise users may not be able to find their search terms due to HTML tags in the text.

---

**Note: Restriction:** Entering text into an HTML field via copy/paste from MS Word is not supported when using Firefox. In this case all formatting tags are copied as plain text.

In Edit mode the HTML field offers a WYSIWYG HTML editor for entering formatted text.




HTML fields support the `Multi Line` field attribute (min. 6 lines). For further details, see [Multi Line](#) on page 232.

If you specify an HTML field configured in MA in the `EMail.BodyFieldID` Web Configuration parameter, **Send As email** sends HTML-formatted emails. For further details, see [EMail.BodyFieldID](#). This is only supported for sending server-side emails. For further details, see [EMail.SendingMethod](#).

Using HTML fields in reports is only supported for HTML reports. For other output formats you need to either use the linked plain text field or create an appropriate XSL transformation.

To configure HTML-formatted signatures:

1. In the **Rep** info area (ID) create a new field of type "html-text" linked to the existing **Signature** field. For further details, see [Data Model](#) and [Aurea CRM Field Types](#) in the *CRM.Core Administrator Guide*.
2. Enter each rep's signature as HTML-formatted text. Or add the HTML field to the ID Expand view, thus enabling users to define their signatures on their own.
3. Check the `Add Rep Signature` field attribute e.g. for the HTML field in the activity info area (from the above example). For further details, see [Add Rep Signature](#) on page 223.
4. The  (**Add Signature**) icon is displayed in the HTML Editor's toolbar when in Edit mode, allowing the user to insert his signature as HTML-formatted text.

## Multi-Language Fields

Learn to add multi-language fields to info areas.

You can add multi-language fields to info areas in order to display and enter data in multiple languages. Multi-language fields are implemented by default for the following info areas: **Answer** (F3), **Field Help** (FH), **Question** (F2) and **Questionnaire/Portfolio** (F1).

For details on defining and using multi-language fields. For further details, see [Multi-language Fields](#) in the *CRM.Core Administrator Guide*.

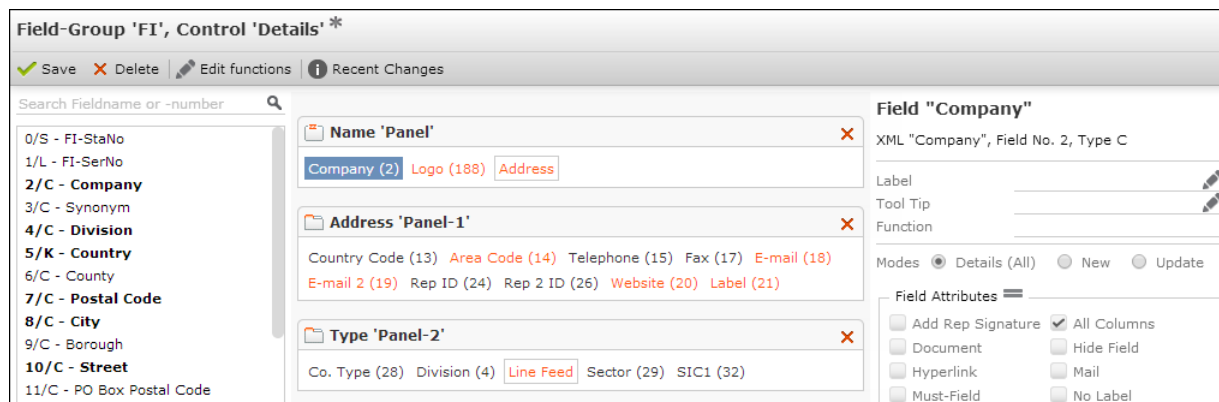
## Field Attributes

Learn about field attributes.

Field attributes define additional properties for fields displayed in masks and lists. You can define field attributes to display a field without label or as hyperlink etc., to perform special checks on a field, or to apply special functions.

You can specify field attributes for all controls except the Matchup control.

**Note:** Field attributes are applied on top of the attributes determined by the user's access rights (e.g., you can set a field to read-only even when a user is allowed to edit it according to his access rights) and on top of style sheet settings (e.g. you can specify `Colspan` and `Rowspan` per field).



Fields for which attributes are defined, are displayed in orange (except in the Default control).

If a field control definition contains multiple occurrences of the same field, you can define different field attributes for each occurrence.

**Note:** Defining different field attributes for multiple occurrences of the same field is supported for all controls except Default and Matchup.

---

**Note:** Defining different field attributes is not supported for linked fields.

---


To customize a field:

1. Select a field in the central part of the Field Control Editor.

The current settings for the selected field are displayed on the right part of the Field Control Editor. (The field's name, XML name, ID and type are output at the top. For further details, see [Field Names and IDs](#) on page 17 and [Data Types in Aurea CRM](#) on page 877.)

The attributes list contains only attributes available for the current context, i.e. depending on control type and field type. Example: The `Document` field attribute is only available for text fields (type C), the `Enhanced Search Field` attribute is only available for Search controls, etc. For the Default control field attributes are filtered by field type only. Attributes are **not** filtered by mode.

For a complete list, see the article “Relevant Field Attributes for Aurea CRM” at <https://support.aurea.com>.

Use the  (Show all attributes / Hide empty attributes) buttons to toggle between displaying all available and the currently defined attributes.

2. Define the following:

- **Label:** You can define alternate field labels here, e.g. when adding a field twice for entering ranges (**From/To postal code**). The texts are saved in the `net_alternate_field_labels` text group. For further details, see [Texts](#) on page 176. This label overrules field names defined globally on the **Alternate Field Labels** page. For further details, see [Alternate Field Labels](#) on page 270.
- **Tool Tip:** currently no function - on mouse-over field labels tooltips display the field help. If no field help exists for a field, the tooltip displays the field label and the max. number of characters. For further details, see [Field Help](#) on page 50.
- **Function:** You can enter e.g. a comparison function for a field. The `FI Search` control in e.g. uses operators for entering a **Postal Code** range:



Valid operator combinations:


">" and "<"

">=" and "<"

">" and "<="

">=" and "<="

**Default** is "="

Click on  **Edit functions** (**Edit functions**) to view and edit all functions defined for the fields of a control in one popup. The popup lists all fields plus their functions, if defined.

- **Modes:** Select the mode for which a field attribute setting applies:
  - For Details controls you can define different sets of field attributes for **New**, **Update** or generally (**Details (All)**).
  - For Default controls you can define different sets of field attributes for all "modes": **All Controls**, **Details (All)**, **New**, **Update**, **View** (for MiniDetails controls), **Search** (for Search controls) and **List** (for List and MatchupList controls).
  - For the other controls the mode is fixed.
- Set **Field Attributes** as described in the following chapters.


### 3. Save the control definition.

Use the Default control to define default field attributes for all fields of an info area: These definitions are used by all controls that have no specific field attributes definition for a field. For further details, see [Default control](#).

## Add Rep Signature

Learn about add rep signature attribute.

Use this attribute to allow users to insert their signature as defined in the **Rep** info area into the field (e.g. for email activities). Example in UPDATE\_DEFAULT: **Text** field in field group `MA`.

If checked, an  (**Add Signature**) icon is displayed beside the text (or HTML) field when in Edit mode (Expand view only).

---

**Note:** This attribute is only supported for text and html fields.

---

For information on adding HTML-formatted signatures. For further details, see [HTML Fields](#) on page 219.

## All Columns

Learn about All Columns attribute.

If checked, the field value is displayed across all columns (i.e. the entire row) independently of its actual length.

---

**Note:** Fields defined as more than 80 and less than 256 characters in the data model are automatically displayed across all columns unless a contrary field attribute (e.g. `Colspan`) is defined.

---

## Auto-Wildcard Search

Learn about Auto-Wildcard Search attribute.


If checked, a wildcard character (\*) is automatically added at the end of the search string for this field. This parameter applies to character fields used in searches.

For the `wildCardEnclose` option. For further details, see [Extended Options](#) on page 228.

## Document

Learn about Document attribute.

Use this attribute to turn a text field into a document field. For further details, see [Document fields](#) on page 831.

In Edit mode a record selector icon  is displayed next to the field for users to upload documents or images.

In View mode and in MiniDetails view the document is displayed as a link and can directly be opened by clicking on the link.

---

**Note:** Document fields defined in the data model ("Document" property checked. For further details, see **Data Model** in the *CRM.Core Administrator Guide*) contain the described functionality without the `Document` field attribute being set.

---

For more information on documents in Aurea CRM.Web. For further details, see [Documents](#) on page 829.

## Hide Field


Learn about Hide Field attribute.

The field is hidden (no field label and field value is displayed).

List controls: Fields with `Hide Field` enabled are also hidden from the **Modify List Fields** dialog.

## Hyperlink

Learn about Hyperlink attribute.

If checked, the field value is displayed as a hyperlink in View mode. In Edit mode, the **Open Web Site** icon  is displayed beside the field. Clicking the link or the icon opens the website entered in the field in a new browser window or tab (depending on the website).

---

**Note:** For fields for which "Hyperlink" property is set in the data model, you do not need to check the **Hyperlink** field attribute.

---


You can define icons as hyperlinks (e.g. to link to a company's social media account). For further details, see [Large Image Template](#) on page 231.



---

## Mail

Learn about Mail attribute.

The field value is displayed as a "MailTo:"-hyperlink in View mode. In Edit mode, the **email** icon  is displayed beside the field. When the link or icon is clicked, a new email message is created for the address in the field.

---

**Note:** For fields for which "email" property is set in the data model, you do not need to check the `Mail` field attribute.

---

## Must-Field

Learn about Must-Field attribute.

Use this attribute for Search and Details controls to determine that a field is mandatory (and must not be empty). For Details controls in Edit mode, the `Must-Field` attribute can be set separately for New and Update mode.

For fields defined as mandatory in the data model (field property "must") or via the access rights definition you do not need to set the `Must-Field` field attribute.

---

**Note:** To ensure consistent behavior across all clients, update recommends defining mandatory fields via the access rights definition in the Aurea CRM win **Rights** module.

---

When defining mandatory fields in the Aurea CRM win **Rights** module (**Fields** column):

- Check **Deny Deletion** to define a field as mandatory when in Edit/Update mode.

---

**Note:** If the value of a field (on a Details control) with Deny Deletion is deleted, a message is displayed. In List controls the original value is restored upon saving the record and no message is displayed.

---

- Check **Mand. Field** to define a field as mandatory when in Edit/New mode.

Concerning mandatory fields on Details controls:

- If the mandatory field is part of a group, the group is expanded on save.
- Mandatory fields are not highlighted when in View mode.
- The names of tabs in Details controls containing mandatory fields are highlighted as **bold** \*. The tab is activated on save if no value was entered.

To customize the formatting of tab labels, change the settings for class `.u8-tabs-item-must.u8-tabs-item` in the `tabs.css` (located in `..\styles\update\base`).

- Per default, an asterisk (\*) is displayed in front of the label of a mandatory field, and an exclamation mark (!) is displayed in front of the label of a field that must not be deleted.

The characters used for indicating mandatory field and fields with deny deletion (\* and !) are defined in texts number 4 and 5 of the text group `common_text`. For further details, see [Texts](#) on page 176.

If you want to display another character or string for mandatory fields, you just need to modify these texts in your configuration.

To customize the formatting of must field labels, change the settings for class `.u8-fields-must` in the `expand.css` file (located in `..\styles\update\crm\widgets`).

## No Label

Learn about No Label attribute.

If checked, the field's label is not displayed.

## No Multi Line

Learn about No Multi Line attribute.

If checked, the field is not displayed as a multi-line field, even if the width exceeds the limit for multi-line fields (256 characters).

## One Column

Learn about One Column attribute.

If checked, the field value is displayed in one column even if its width exceeds the limit for one column field.

---

**Note:** Fields defined as less than 80 characters in the data model are automatically displayed as one column fields unless a contrary field detail (e.g. `All Columns`) is defined.

---

## Password Field

Learn about Password Field attribute.

If enabled, the field value is displayed in password style (\*\*\*\*).

---

**Note:** This field attribute is not supported by multi-line fields.

---

## Phonetic Search Checked

Learn about Phonetic Search Checked attribute.


Defines that the phonetic search is active (🔊) when the search is opened.

---

## Phonetic Search Enabled

Learn about Phonetic Search Enabled attribute.

The field can be searched using a phonetic search algorithm.

An additional icon  is displayed beside the field, allowing the user to select whether he wants to search for this field's value phonetically. For a description of the phonetic search algorithm. For further details, see **Phonetic Search** in the *Web Business Logic User Guide*.

Make sure you check this attribute for only one field of the Search control.

---

**Note:** For the field for which you check this attribute a phonetic index for must be defined in the data model, see [Index](#) in the *CRM.Core Administrator Guide* (index field property PHONETIC). Do not enable the phonetic search for enhanced search fields. For further details, see [Enhanced Search Field](#) on page 237.

---

## Placeholder

Learn about Placeholder attribute.

Unused fields can be defined as placeholders for more freedom in the design of Details controls.

If a field is defined as `Placeholder`, its label and content are hidden: This space is empty and the next field does not move up. For placeholder fields, you can also specify a `RowSpan`. For further details, see [Rowspan](#) on page 234.

---

**Note:** To add an empty space at the end of row in a Details control, use the non-field element **Line Feed**. For further details, see [Non-Field Elements](#) on page 210.

---

---

**Note:** Fields flagged as `Placeholder` are not exported (exception: export as XML - `fieldplaceholder` attribute).

---

## Readonly

Learn about Readonly attribute.

If checked, the field is read-only and cannot be changed by the user.

For fields defined as read-only in the access rights definition (**Fields** column > **Deny Update**) you do not need to set the `Readonly` field attribute.

Fields with **Deny New** in the access rights definition are displayed as read-only in Edit/New mode. In Edit/Update mode users may enter a value but the field is reset to empty on save.

## Select Field for Applying Last Value

Learn about Select Field for Applying Last Value attribute.

If a new record is created and the Web Configuration Parameter `Expand.InitialValuesFromLastInput` is set to "Use admin-defined values from last data entry", the values of fields with this attribute checked are used for the `ExpandNew` action. For further details, see [Expand.InitialValuesFromLastInput](#).

## Colspan

Learn about Colspan attribute.

Defines the number of columns that the field value uses - Details, Search and MiniDetails controls.

Label and text of a field count as one column.

To use `Colspan` for fields that would by default be displayed across the whole mask due to their field length, you need to check the `One Column` field attribute as well. For further details, see [One Column](#) on page 226.

## Extended Options

Learn about Extended Options attribute.

The following options are available:

- `showOnlyForAffiliations`: Use this option to provide additional information when the Affiliation View is active. For further details, see [Affiliation View](#) on page 752.

Example: Add **Company** as a linked field to an **Activity** (MA) sub-list and define `showOnlyForAffiliations=true` for this field. Thus **Company** is only displayed when the Affiliation View is active and MA records for various affiliated companies are listed.

In the **Modify List Fields** dialog fields with `showOnlyForAffiliations=true` are displayed with status "hidden" when the Affiliation View is not active.

---

**Note:** The `Hide Field` attribute overrules `showOnlyForAffiliations=true`.

---

- `UseRootRecord`: To display the titles of the D3 SubList entries as links, define `UseRootRecord=true` under `Extended Options`. Use this in order to be able to open a reconstructed serial letter document by clicking the field "Title" from D1 in a list of document links (D3).
- `wildCardEnclose`: If the `Auto-Wildcard Search` field attribute is enabled a wildcard character `*` is added at the end of the search string. Specifying `wildCardEnclose=true` as extended option adds the wildcard character before and after the search string. For further details, see [Auto-Wildcard Search](#) on page 224.

---

**Note:** Using a wildcard character before and after the search string may result in a severe performance decrease, especially for large amounts of data, and should thus only be enabled if absolutely necessary.

---

- `autoComplete`: Auto-completion of field attributes is disabled if you set this value to `autoComplete=false`.
- `actionNoFocus`: Set this extension option to `true` to ensure that on pressing the TAB key the input focus moves to the next field in the tab order. If this option is not set on a field then the focus moves to the magnifier and clear selection action icons before moving on to the next field. This requires multiple TAB key presses for the user to move focus to the next field.
- `useWildcardForOrgGroup` - Use this option for **Participant Id** fields to control organizational group rep addition to a filter. If it is set to `true`, organizational group Rep Id is added as a wildcard to enable search by Rep Id only. The following is an example extended option setting:

```
{
  widgetOptions : {
    useWildcardForOrgGroup : true
  }
}
```

Use semicolons to separate multiple `Extended Options`.

## Image

Learn about Image attribute.

If a control contains a document field with the `Image` attribute checked, users can upload image files and display a preview (e.g. to show pictures of products in the **Item Master** info area (AR)). For further details, see [Uploading Documents](#) on page 830.

---

**Note:** The `Image` attribute only works for document fields defined in the data model ("Document" property checked). For further details, see [Data Model](#) in the *CRM.Core Administrator Guide*.

---



---

**Note:** For uploading and previewing images, Aurea CRM.Web requires local access to these images - therefore, Aurea CRM.Web must be added to the browser's "Trusted Sites".

---



---

**Note:** Using large images and displaying images in lists with many results may lead to performance problems.

---

Valid parameters for the `Image` attribute:

Parameter	Default	Description
<code>maxWidth</code>	2048	maximum allowed width of a picture in pixel (checked before upload)
<code>maxHeight</code>	2048	maximum allowed height of a picture (checked before upload)
<code>previewWidth</code>	n/a (see below)	maximum width of the picture preview in the document field
<code>previewHeight</code>	Expand: 256 MiniDetails: 128 other views: 96	maximum height of the picture preview in the document field
<code>allowedExtensions</code>	<code>.gif .jpg .png .mp</code>	list of allowed extensions (checked before upload)

Use JSON syntax for specifying the parameters.

**Example:** `{maxHeight:200, maxWidth:300, allowedExtensions:".jpg|.png"}`

The document upload dialog shows the defined defaults.

Calculation of `previewWidth` and `previewHeight`:

- `previewWidth` and `previewHeight` not set: The default `previewHeight` is used. `previewWidth` is the maximum available width. The aspect ratio of the image is taken into account.
- Both `previewWidth` and `previewHeight` set: Both parameters are applied. Images with another aspect ratio is distorted.
- `previewWidth` set: `previewHeight` is calculated based on the aspect ratio of the image.
- `previewHeight` set: `previewWidth` is calculated based on the aspect ratio of the image.

---

**Note:** If the calculated `previewWidth` exceeds the maximum available width, the maximum available width is used instead. `previewHeight` is re-calculated based on the aspect ratio of the image.

---

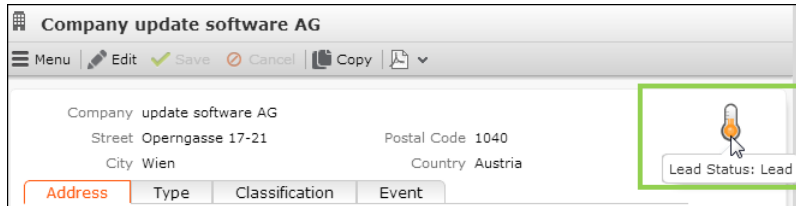
For Details and MiniDetails controls you can use the `Colspan` and `Rowspan` field attributes to control the position of the picture in relation to other fields. For further details, see [Colspan](#) on page 228 and [Rowspan](#) on page 234.

**Note:** When using `Rowspan` for an image field on a multi-column mask, the vertical alignment of the elements (without `Rowspan` defined) next to the picture is handled differently by different browsers.

## Large Image Template

Learn about Large Image Template attribute.

Displays an image depending on a field value in a separate area beside Details and MiniDetails controls. To display images in lists, use the `Render Hook` field attribute. For further details, see [Render Hook](#) on page 239.



### Syntax:

Specify the name of an image registered in CRM.Designer plus `{VALUE}` as placeholder for the field's value. For further details, see [Images](#) on page 717

```
<image_name>_{VALUE}
```

Use JavaScript expressions to get and transform the value.

Large Image Template (and Small Image Template) support character, catalog and boolean values.

If you define `Large Image Template` for a hyperlink field defined in the data model or via field attribute, users can click on the icon to open the linked page (e.g. to access a company's social media pages). For further details, see [Hyperlink](#) on page 224.



### Examples:

```
Small Image Template = CAT_CONTACT_BTBT_{VALUE}
```

Displays the images associated with the different values of the **Activity Type** catalog. The images are registered as `CAT_CONTACT_BTBT_0` to `CAT_CONTACT_BTBT_7` for the first eight catalog entries.

```
Large Image Template = CAT_AQ_{VALUE}
```

Displays the image for a company's current **Lead Status**. The images are registered as `CAT_AQ_0` to `CAT_AQ_6` for the first seven catalog entries.

```
Large Image Template = CAT_ABC_{javascript:($['VALUE'] || '').toUpperCase();}
```

Displays the image for a company's current **ABC** rating. The images are registered as `CAT_ABC_A`, `CAT_ABC_B` and `CAT_ABC_C`. For letters other than A, B or C no image is displayed.

```
Large Image Template = private_{javascript:u8.services.typeConverter.to-Boolean($['VALUE'])}_value
```

Displays different images depending on the state of the Boolean field **Private** in MA. You need to create and register the images (`private_true_value` and `private_false_value`) for this example, as they are not part of `UPDATE_DEFAULT`.

```
Large Image Template = SocialNetwork_Facebook{javascript:$['VALUE']?"":"_doNothing";}
```

Displays the Facebook icon (registered as `SocialNetwork_Facebook`), if the **Facebook** field contains an address. If the **Facebook** field is empty no icon is displayed, unless you registered an image called `SocialNetwork_Facebook_doNothing` indicating that no address is available.

## Multi Line

Learn about Multi Line attribute.

Use this attribute to display a field as a multi-line field.


---

**Note:** Fields defined as more than 256 characters in the data model are automatically displayed as multi-line fields unless a contrasting field attribute is set.

---

Enter the number of lines as parameter, e.g. `Multi Line = 3`. To allow users to comfortably view the entire field contents even if the number of visible lines is exceeded, you can specify additional parameters as JSON, e.g. `Multi Line = {lines: 3, allowExpand: true, expanded: true}`.



Name	Type	Description
lines	int	The number of visible lines.
allowExpand	Boolean	If set to <code>true</code> , an <b>Expand/Collapse</b> toggle button  is displayed beside the field, allowing the user to display the entire field content.
expanded	Boolean	If set to <code>true</code> , the field is displayed expanded, i.e. all lines are visible. <code>expanded</code> only works if <code>allowExpand</code> is set to <code>true</code> .

**Note:** The JSON is supported for Details and MiniDetails controls for text and HTML fields.

**Note:** All multi-line text fields configured in MiniDetails controls are expanded by default.

## Rep Filter

Learn about Rep Filter attribute.

Use this field attribute to implement a fixed filter for a specific rep field. The filter is applied in the rep picker (all views) and when using auto-completion.

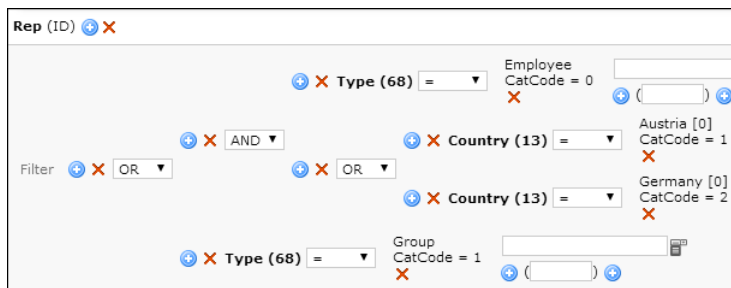
You can specify the filter with or without prefix:

```
Rep Filter = myEmployees
```

```
Rep Filter = ID.myEmployees
```

Only filters defined for the **Rep** info area (ID) are taken into account. Filters defined for other info areas are ignored. However you can define a filter with conditions on a linked info area, e.g. GZ. For further details, see [Defining Filters](#) on page 489.

The filter is applied to ID records of all types (rep, group, resources). Add a condition on the **Type** field (ID:68), to determine which types are taken into account. Example: To see all available groups and filter only employees by **Country**:



You can use the `Rep Filter` attribute for rep and participant fields on Details, List and Search controls.

The filter name (without prefix) is displayed in the rep picker's title bar. The **Favorites** node in the rep picker is hidden.

For participant fields (e.g. **Participant ID** in MA) the record's main rep is listed as well, regardless of whether this rep meets the filter conditions or not.

For **Participant ID** fields you can control the addition of the organizational group's Rep to the filter using the extended option `useWildcardForOrgGroup`. For further details, see [Extended Options](#).

### Rowspan

Learn about Rowspan attribute.

Defines the number of rows used by the field value - Details, Search and MiniDetails controls.

---

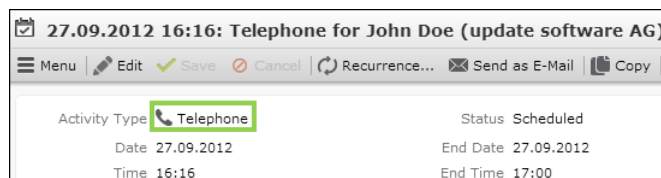
**Note:** When using `Rowspan` for an image field on a multi-column mask, the vertical alignment of the elements (without `Rowspan` defined) next to the picture is handled differently between browsers. For further details, see [Image](#) on page 229.

---

### Small Image Template

Learn about Small Image Template attribute.

Displays an image in front of a field depending on the field value.



For information on syntax and examples. For further details, see [Large Image Template](#) on page 231.

### Delete Link

Learn about Delete Link attribute.

To clear a link field, users need to click the **Remove Link** button **X** (beside the record selector button **↻**). For details on when **X (Remove Link)** is displayed, see [Select-Function](#) on page 240.

Field contents that were copied via the `CopyFields` property of a `Select-Function`, are automatically cleared/changed when the respective link is removed or changed.

Use the `Delete Link` attribute to define a specific Delete function to be called by the **Remove Link** button , e.g. to

- clear/reset field values when deleting/changing a link (no matter if a `Select-Function` was defined or not)
- change the content of fields previously filled by a `Select-Function`
- provide a custom JavaScript function.

---

**Note:** After deleting a link it is necessary to save the record to reset the search context. For further details, see [Search Context](#)

---

**Note:** For non-link fields the `Delete Link` attribute is only supported if a `Select-Function` is defined as well, i.e. you cannot use `Delete Link` to clear e.g. **Street** when deleting **Company** in FI: You would have to define a trigger for that instead.

---

**Note:** If you use the `Delete Function` to clear/reset a field that was not copied via `copyFields`, you also need to explicitly specify the fields copied with `copyFields` in the `Delete Function`.

---

### Syntax:

Name	Type	Description
<code>clearFields</code>	function	delete function
<code>fieldIds</code>	int or string	The ID or XML name of the field(s) to be cleared/reset.
<code>value</code>	string	The new value of the field. Use this property to reset a field to some default value.

### Examples:

To clear the contents of the **Info Area Code** and **Field Number** fields in Z5:

Definition for **Info Area Code** (Z5/8):

```
clearFields({
  fieldIds: ["TableId", "FieldId"],
  value: {"TableId": "", "FieldId": "-1"}
});
```

Definition for **Field Number** (Z5/9):

```
clearFields({  
  fieldIds: ["FieldId"],  
  value: "-1"  
});
```

If **Info Area Code** is cleared, both the **Info Area Code** and **Field Number** fields are reset; if **Field Number** is cleared, only **Field Number** is reset.

To clear a ticket's **Priority** if another **End Customer Company** is selected:

Definition for **End Customer Company** (KM/85):

```
clearFields({  
  fieldIds: ["Priority"]  
});
```

To reset **FreeC1** in KM to "D" (copied from **ABC** (FI) via a **Select-Function**):

**Select-Function** defined for **End Customer Company** (KM/85):

```
selectRecord({  
  from: {  
    infoAreald: "FI"  
  },  
  target: {  
    addLink: true,  
    copyFields: {  
      "FreeC1": "ABC"  
    }  
  }  
});
```

**Delete Link** defined for **End Customer Company** (KM/85):

```
clearFields({  
  fieldIds: ["FreeC1", "CoGrp2", "CoNo2", "Company2"],  
  value: {"FreeC1": "D"}  
});
```

The link and respective key fields ("CoGrp2", "CoNo2", "Company2") are cleared, **FreeC1** ("FreeC1") (containing the company's ABC-Classification) is reset to "D".

## Enhanced Search Field

Learn about Enhanced Search Field attribute.

Use this attribute in Search controls to define a custom search field that searches the contents of several fields of an info area (e.g. the three email fields or the four telephone fields in KP). The search criteria can be combined by logical AND or OR. Additional search criteria added by the user (fields and filters under **Add more ...**) are AND-combined with the enhanced search field.

You can define any field of the current info area as an enhanced search field. (You may want to specify a **Label** indicating the fields functionality.) The field itself is not used in the search: You need to specify all search fields in the attribute.

### Syntax

```
{searchFields: [comma-separated list of fields], type: ["multi" (default)|"combined"]}
```

You can specify either unique field IDs or XML names.

You can search numeric fields, text fields and variable catalog fields.

Depending on the value of `type`, the search criteria are combined by logical AND or OR:

```
{searchFields:["FreeC1","FreeC2","FreeC3"], type: "multi" }
```

When entering the search term "Test" the resulting filter is `F74='Test' OR F75='Test' OR F76='Test'`.

```
{searchFields:["FirstName","LastName"], type: "combined" }
```

Depending on the user input the resulting filter is:

- search term "John Doe": `(F3='John' AND F2='Doe') OR (F2='John' AND F3='Doe')`
- search term "John,Doe": `(F3='John' AND F2='Doe')`
- search term "John": `F3='John' OR F2='John'`

Superfluous whitespace is trimmed, i.e. searching for "John , Doe" gets the same result as searching for "John,Doe".

---

**Note:** A field defined as an enhanced search field may not be used as a "normal" search field as well: If you define e.g. the **Company** field in FI as an enhanced search field, you can not add it again as a normal field.

---

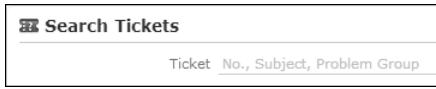
### Example

To search for tickets by **No.**, **Subject** and **Problem Group**, in the `KM` Search control define the following for the **No.** field:

1. Check `Auto-Wildcard Search`.
2. Check Enhanced Search Field and enter

```
{searchFields:["No","Title","ProblemGroup"], type:"multi"}
```

### 3. Change the field's **Label** to "Ticket".



## Hook Function

Learn about Hook Function attribute.

Use the `Hook Function` field attribute to define "field hooks" that perform calculations based on one or more field values.

Field hooks are small JavaScript functions that are executed whenever values of fields change. Field hooks are executed **after** the value entered by the user is written to the model (i.e. after validation and conversion).

### Syntax

For information on the syntax, on handling different field types, on implementing asynchronous `Hook Functions` etc, see the article "How to Define Field Hooks" at <https://support.aurea.com>.

### Examples

Aurea CRM.Web contains an out-of-the-box function calculating the correct duration and unit depending on start date/time and end date/time. (You can use this function for any info area containing the respective fields). The following field hook is configured for the `MA Details` control in `UPDATE_DEFAULT` as an example:

```
var fieldNames = {
  startDate: "Date",
  startTime: "Time",
  endDate: "EndDate",
  endTime: "EndTime",
  duration: "Duration",
  unit: "Unit"
};
u8.services.collaboration.updateDurationUnit($, fieldNames);
```

In the **Property** info area (OJ), you can define field hooks to set the **ABC** field to "A" whenever the **Reference** field is checked and to uncheck **Reference** if the **Project Status** field is set to "Lost".

To implement this you need to define two field hooks (in the OJ Details control):

**Reference** field:

```
var valuesAsBoolean = $.getNativeValue("Reference"); // true or false
```

```
if (valuesAsBoolean === true)
```

```
$.setValue("ABC", "A");
```

**Project Status** field:

```
var catCode = $.getValue("ProjectStatus");
```

```
if (catCode == 2) // lost
```

```
$.setValue("Reference", false);
```

## Render Hook

Learn about Render Hook attribute.

Use the `Render Hook` attribute to define conditional formatting (colors, icons, styles) for fields depending on a field's value.

<input checked="" type="checkbox"/> Render Hook: <code>\$.setColor("white");\$.setBackground("black");</code>	Company <b>update Software AG</b> Synonym <b>update</b>
--	--

**Note:** Render hooks defined for a field that is part of a combined field always apply to the entire combined field. For further details, see [Add Combined Field](#) on page 214. In forms render hooks apply only to the respective cell (not to an entire widget). For further details, see [Forms](#) on page 621.

The provided `Render Hook` is executed as JavaScript code (without any transformation) – thus, you can use standard JavaScript syntax.

The following functions are available:


Function	Description
<code>setBackground</code>	Changes a field's background color.
<code>setColor</code>	Changes a field's font color.
<code>setContextBackground</code>	Defines the background color of the element hosting the widget, e.g. the list row.
<code>imageAdd</code>	Adds an image in front of the field value.
<code>imageReplace</code>	Replaces the field value by an image.
<code>addClass</code>	Changes the field attributes corresponding to the CSS style.

For detailed information on the syntax and some examples. For further details, see [Defining Render Hooks](#) on page 244.

## Select-Function

Learn about Select-Function attribute.

The following functions are available:


- `selectRecord`: To link a record to the current record, users click on the record selector icon  beside the link field and search for records according to the settings you specified in the `Select-Function`. For further details, see [Defining selectRecord Functions](#) on page 249.
- `findAsYouType`: To link a record to the current record, users type a search string into the link field and get all matches in a drop-down list according to the settings you specified in the `Select-Function`. For further details, see [Defining findAsYouType Functions](#) on page 253.

---

**Note:** `selectRecord` and `findAsYouType` functions are supported for Details and List controls.

---

- `selectFormat`: Users can select a format. For further details, see [Defining selectFormat Functions](#) on page 262.
- `selectInfoAreaId/selectFieldId`: Users can pick an info area and field. For further details, see [Defining selectInfoAreaId and selectFieldId Functions](#) on page 264.
- In addition, you can call any custom JavaScript select/deselect function via the `Select-Function` attribute. For more information, see the article “How to create a custom selector” at <https://support.aurea.com>.

Standard link fields (Z-fields) that are configured into a Details or List control automatically display the record selector icon . This provides users with the possibility to search for and select a record that is then linked to the current record (Example in UPDATE\_DEFAULT: the **End Customer Company** field in the **Ticket** info area).

The `Select-Function` attribute extends the behavior and functionality of link fields: It defines which records from which info area can be selected and how the relation is set after a record has been selected. It also allows fields to be filled with values from a linked record. (Examples in UPDATE\_DEFAULT: the **Postal Code** field in the **Company** info area, the **Predecessor** field in the **Ticket** info area etc.)

**Example:** The **Predecessor** field in the **Ticket** info area has the following `Select-Function` defined (KM Details control):

```
selectRecord(  
{  
  from:  
  {  
    infoAreaId: "KM",  
    searchContext:  
    {  
      items:
```



```

    [
      null,
      "KP",
      "FI"
    ]
  },
  target:
  {
    71: 6,
    72: 7,
    addLink: true
  }
})

```

Predecessor




The selected record is then linked to the current record. The record's reference text is written to the link field. You can define custom reference texts in the **Configuration** info area (MC) (**Data Model** category, **Reference Text** option). For further details, see **Configuration Info Area** in the *CRM.Core Administrator Guide*.

Predecessor Annual Service

When in View mode, clicking on the link opens the default view of the linked record, click and hold opens the QuickView.

If a link is changed (because the user selects a different record), data is reset.



To clear a link field, users need to click the **Remove Link** button  (beside the record selector button). The **Remove Link** button is displayed if the following conditions are met:

To clear/reset field values when deleting/changing a link, you can define a custom Delete function. For further details, see [Delete Link](#) on page 234.

---

**Note:** Both the record selector and remove link buttons are displayed only when in Edit mode.

---

**Note:** By default, reference fields that are marked as read-only (either in the Aurea CRM data model or in CRM.Designer via the `ReadOnly` field attribute, see [ReadOnly](#) on page 227) never display a record selector button . You can override this behavior by setting the Web Configuration parameter `RecordSelector.ShowSelectorIfDefined` to `true` and explicitly specifying a `Select-Function` for the desired reference fields, see [RecordSelector.ShowSelectorIfDefined](#). The record selector icon  is then displayed regardless of the above mentioned read-only settings.

---

**Note:** In this case, it is the administrator's responsibility to ensure that the `Select-Function` "makes sense" and fills the required (key) fields. This possibility increases Aurea CRM.Web's flexibility and may e.g. be used to let users reassign parent records if required.

---

---

**Note:** User's access rights settings are only checked for the field where the `SelectFunction` is defined (not for the fields contained in the function).

---

**Note:** If a user has no update rights for an info area, no record selectors `q` are displayed.

---

## Widget

Learn about Widget attribute.

Use this attribute to assign another than the default field widget to a specific field.

The following built-in special field input widgets are available:

- `RangeInput`: Displays a slider widget for percentage fields.



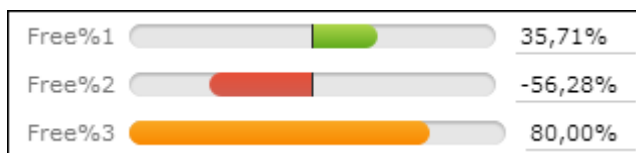
The slider widget is used for all percentage fields.

To disable the percentage slider for a field, define `widget = TextField`.

Free%1 35,71%

Define this in the Default control for percentage fields where users are supposed to enter floating point values.

- `PercentField`: All predefined percentage fields in the data model are decimal fields and automatically signed. In Aurea CRM.Web they use the `RangeInput` widget by default. To support the input of negative percentage values via the slider widget, use the `PercentField` factory (for those fields where negative values are actually used).

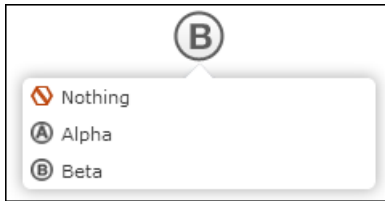


For information on the available options for `PercentField`, see the article “HOWTO configure percentage fields” at <https://support.aurea.com>.

- `RatingInput`: Graphically displays a row of values. Users can select a value by clicking on an icon.
- `CatalogRatingInput`: Graphically displays catalog values. Users can select a catalog value by clicking on the respective icon.



- `ValuePicker` and `FieldValuePicker`: Use these widgets to display an image for a field value and allow users to pick another value by clicking on the image.



You can implement this for text fields (`ValuePicker`) and catalog fields (`FieldValuePicker`). For a complete description of all available settings, see the article “How to Configure the Field ValuePicker” at <https://support.aurea.com>.

**Note:** The factories for these widgets are called `ArrayPickerField` for the `ValuePicker` widget and `CatalogPickerField` for the `FieldValuePicker` widget.

**Restriction:** `ValuePicker` and `FieldValuePicker` widgets are not supported in List controls. The (textual) field values are displayed as a fallback.

## Syntax

Enter one of the following:

- the widget type (i.e. widget's name), e.g. `RangeInput`
- the name of the widget factory, e.g. `UserWidgetFactory`
- the widget type plus additional options, e.g. `{type: "UserWidget", options: {text: "City"}}`
- the factory plus additional options, e.g. `{factory: "UserWidgetFactory", options: {widgetText: "PostalCode"}}`

For more details, see the article “How to Configure a different widget for a field” at <https://support.aurea.com>.

If the widget factory definition contains definitions for specific fields, the widget is automatically used (i.e. in that case you do not need to set the field attribute).

For more information on defining custom widgets, see the articles “How to create a field widget” and “How to Register a field widget” at <https://support.aurea.com>.

## Examples

To configure the `CatalogRatingInput` widget for the **Activity Type** field in the MA Details control:



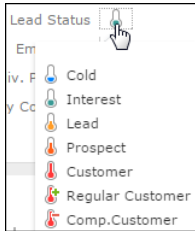
```
{
  type: "CatalogRatingInput", // widget name - mandatory
  options: {
    infoAreaId: "MA", // info area - mandatory
    fieldId: "Contact", // field name or id - mandatory
    showValue: true, // if value text is visible
    imageTemplate: "CAT_CONTACT_BT_{VALUE}" // image
  }
}
```

To enter the **Service Level** of a ticket (KM) using a star rating:

Service Level ★ ★ ★ ☆ ☆ Gold

```
{ type: "CatalogRatingInput", // widget name
  options: {
    infoAreaId: "KM",
    fieldId: "ServiceLevel",
    showValue: true,
    style: "stars" // "plain", "boxes", "stars", "stars-in-boxes"
  }
}
```

To use the `FieldValuePicker` widget for the **Lead Status** field (FI:38):



Since the `FieldValuePicker` and `ValuePicker` widgets are using the `Large Image Template` field attribute as a fallback (for their own `imageTemplate` attribute), you can define the following for FI:38:

Set the `Large Image Template` field attribute to `CAT_AQ_{VALUE}` (already defined in `UPDATE_DEFAULT`).

Define `Widget` as

```
{
  factory: "CatalogPickerField"
}
```

---

**Note:** Make sure you define your widgets for both View and Edit modes.

---

Refer to the **SDK** page for more sample configurations and information on valid options for all built-in widgets: **Inputs > RangeInput, RatingInput, CatalogRating-Input**. Since the **SDK** page already "knows" which widget is used, make sure you only enter the `{}` expression after `options:.` The value picker widgets can be found at **ValuePicker > ValuePicker, FieldValuePicker**.

All field input widgets are also available when defining forms.

## Defining Render Hooks

Learn about `Render Hook` field attribute.

The `Render Hook` field attribute is used to style the current cell (list cell or mask field) and/or the current context.

## Syntax

<code>\$.getValue(["&lt;XmlFieldName&gt;" [, "&lt;Alias&gt;"]])</code>	Returns the value of a field with the given XML name and from the record with the given alias, if present.  <code>\$.getValue()</code> without any parameters returns the value of the current field, i.e. the field for which the render hook is defined.
<code>&lt;XmlFieldName&gt;</code>	XML name of the field
<code>&lt;Alias&gt;</code>	Used to get values from parent records: If the default link is used, specify the info area ID (e.g. "MA").  If another link is used, specify info area ID + link ID (e.g. "FI1", "FI2"), see <a href="#">Link IDs</a> on page 19.

**Note:** `$.getValue()` always returns the field content as provided by the core. Therefore, when using other than text fields, you need to use a conversion function or escape e.g. number formatting by a `replace` statement. When using catalog fields you need to specify the catalog code instead of the catalog value.

**Note:** To get a field's value as provided by the core, use `alert ($.getValue("<XML-FieldName>"))`;

<code>\$.&lt;StyleFunction&gt;(&lt;StyleParameter&gt;);</code>	Sets the formatting of a field.
<code>\$.setColor("&lt;color&gt;")</code>	Sets the font color of the field's value.
<code>\$.setBackgroundColor("&lt;color&gt;")</code>	Sets the background color of the field.
<code>\$.setContextBackgroundColor("&lt;color&gt;")</code>	Sets the background color of the field's context.
<code>&lt;color&gt;</code>	JavaScript color name (e.g. "red") or hexadecimal color code e.g. ("#ff0000").
<code>\$.imageAdd("&lt;imageName&gt;", "&lt;size&gt;")</code>	Adds an image in front of the field value.
<code>\$.imageReplace("&lt;imageName&gt;")</code>	Replaces the field value by an image.

<code>&lt;imageName&gt;</code>	Name of an image file in <code>..\images\&lt;&lt;theme e.g. update&gt;\</code> , e.g. <code>"image.png"</code>
<code>&lt;size&gt;</code>	This optional parameter for <code>\$.imageAdd</code> allows you to specify an image size (e.g. <code>\$.imageAdd("CAT_AQ_" + \$.getValue(), "xs");</code> ).  Possible values:  <code>"xs" ... 14x14"s" ... 16x16nothing ... 32x32"lg" ... 48x48"xl" ... 64x64</code>

**Note:** A PNG file with the specified suffix (`*-xs.png`) must be available in `..\images\<<theme>`, otherwise nothing is displayed. For more information, see the article “Image Naming Guidelines” at <https://support.aurea.com>.

<code>\$.addClass("&lt;ClassName&gt;")</code>	Specifies a CSS class to be applied to the field.
<code>&lt;ClassName&gt;</code>	Name of the CSS style and path to the CSS file relative to the application path separated by comma, e.g. <code>"myBoldStyle", "../styles/custom.theme/custom.css"</code>

`Render Hook` definitions containing syntax errors are ignored and a message is written to the client log.

You can define `Render Hook` functions for all field types.

**Note: Highlighting Boolean fields:** Use `$.setBackgroundColor` (since the check box itself is an image and its color can therefore not be changed using `$.setColor`).

`Render Hook` functions work for List, Expand/Quick and Minidetails views in both View and Edit mode. **Exceptions:** `imageAdd` and `imageReplace` only work for List and Minidetails views and only in View mode.

You can only set conditions for fields that are part of the control's configuration: The rendering function is not activated if either the field where the `Render Hook` is defined or the comparison field(s) used in the JavaScript code are not configured in the current view. Therefore, if a condition uses a field from a linked info area, you need to configure this field into the control as a (hidden) linked field. For further details, see [Add Linked Fields](#) on page 216.

## Example

To set the formatting of a field:

```
$.setColor("white");$.setBackgroundColor("black");
```

To define conditional formatting for a field:

- If the **Postal Code** field of a company (XML name: "ZipCode") is empty or undefined, set the background color to red:

```
if (!$.getValue("ZipCode"))
    {$.setBackgroundColor("red");}
```

- For person records, if the **Postal Code** of the parent company has a value between 1000 and 1999, set the context background color to pale green:

```
if ($.getValue("ZipCode","FI")>=1000 &&
    $.getValue("ZipCode","FI")<2000)
    {$.setContextBackgroundColor("#CCFACC");}
```

- If a company's **Postal Code** has the value "1040", set the font color to red, in all other cases set it to blue:

```
if ($.getValue("ZipCode")==1040)
    {$.setColor("green");}
else
    {$.setColor("blue");}
```

- If a company's **Revenue** field has exactly the value "1.000.000", set the font color to red; if it has a value below 1.000.000, set the background color to grey; if it has a value above 1.000.000, display an image in front of the value:

```
var _strText = $.getValue("Revenue").replace(/\./g, "" );
var _revenue = parseInt(_strText);

if (_revenue == 1000000) {$.setColor("red");};
if (_revenue < 1000000) {$.setBackgroundColor("grey");};
if (_revenue > 1000000) {$.imageAdd("ico_Sales.png");};
```

- Set the font color, if the **Subject** of an activity contains a certain string:

```
var _searchFor = "update"
var _string = $.getValue().toLowerCase();
StringContains = _string.indexOf(_searchFor) != -1;

if (StringContains) {$.setColor("green");}
```

- Set the background color of a ticket's **Priority** field for open tickets only (i.e. field **Status** (XML name "AccSwitch") != "Closed"):

```
if ($.getValue("AccSwitch")!=2)
    {
        if ($.getValue()==1)
            $.setBackgroundColor("#dd0000");
        else if ($.getValue()==3)
            $.setColor("#00dd00");
    }
```

To define conditional formatting based on **catalog values** you need to specify the value's code/external key/ID:

- Set the font color depending on a company's country:

```
if ($.getValue("Country")==2)
    {$.setColor("blue");}
    Since the codes of variable catalogs may not be unique in your
    environment, you can define a render hook using external keys instead:
    var code = $.getValue();
    var catalogField = u8.services.schema.getField("FI", "Country");

    var externalKey =
u8.services.catalogs.codeToExternalKey(catalogField, code);
```

```
if (externalKey == "100_2_42") {$.setColor("blue");}
```

- Add icons to the **Activity Type** field e.g. in MA's MiniDetails view:

```
var _cnt = $.getValue("Contact");

switch (_cnt) {
case "0":
{$.imageAdd("u8-contacttype-phone-xs.png");}
break;
case "1":
{$.imageAdd("u8-contacttype-visit-xs.png");}
break;
case "5":
{$.imageAdd("u8-contacttype-email-xs.png");}
break;
default:
{$.imageAdd("u8-contacttype-event-xs.png");}
break;
}
```

- Add an exclamation mark to the **Subject** field for tickets with **Priority "high"**:

```
if($.getValue("Priority")==1)
$.imageAdd("u8-participationtype-required-xs.png");
else $.imageAdd("u8-shim.png");
```

To define conditional formatting based on a **Boolean field**:

- Different coloring for true/false:

```
if ($.getValue("FreeL1") == u8.session.localeSettings.booleanTrue)
{$.setColor("blue");}
else
{$.setColor("grey");}
```

- Display an icon for recurring activities (as configured in the MA List control in UPDATE\_DEFAULT):

```
var isSerialContact =
u8.services.typeConverter.toBoolean($.getValue("IsSerialContact"));
$.imageAdd(isSerialContact ? "ico_recurring.png" : "shim.png");
```

To define conditional formatting per **rep** (e.g. for the activity list):

```
var _rep=$.getValue("Rep");

if (_rep == u8.session.identity.repName)
{$.setContextBackgroundColor("yellow");}
else
{$.setContextBackgroundColor("white");}
if (_rep == "Administrator")
{$.setContextBackgroundColor("red");}
```

Available functions for identifying the current rep. For further details, see [u8.session.identity](#).

Render Hooks with **date/time**:

- Display scheduled activities for today and older in another color:

```
var dtToday = new Date();
var u8date =
u8.services.typeConverter.toDate($.getValue("Date"));
```



```
if (u8date <= dtToday && $.getValue("Status")==="0")
{$.setContextBackgroundColor("#ffc000");}
```

- Comparing dates: To set a different row color for To-Dos linked to Activities with Date (MA) = StartDate (TD) define the following for the **Starts on** field in TD. (Don't forget to configure the **Starts on** field from MA into the TD List control.)

```
var _getMAStartDt = $.getValue("Date","MA");
var _TDStartDt = $.getValue();
if (_getMAStartDt == _TDStartDt)
{$.setContextBackgroundColor("yellow");}
```

- Comparing times (by extracting the time from a date field):

```
var _todayDt = new Date().stripTime();
var _freeD1 =
u8.services.typeConverter.toDate($.getValue()).stripTime();
if (_todayDt.getTime() == _freeD1.getTime())
{$.setBackgroundColor("blue");$.setColor("white");}
```

To manipulate the contents of a **text field**, e.g to truncate the document key information when listing checked out documents define the following for the **Text** field in the T3.Documents List and Details controls:

```
var text = $.getText();
text = text.substring(0, text.indexOf("<"));
$.setText(text);
```

Using customized **CSS definitions**: Create a CSS file with custom CSS classes. Import your CSS file into your main CSS file. For further details, see [Customizing Styles](#) on page 15. Use your CSS classes in your Render Hook definitions, e.g. in FI:

```
if ($.getValue("ZipCode") == "1040")
{$.addClass("myCustomCSSClass","../styles/custom.theme/custom.css");}
```

---

**Note:** Border styles are not displayed correctly in Google Chrome.

---

## Defining selectRecord Functions

Learn to define selectRecord function.

To define a `selectRecord` function:

1. In the Field Control Editor select a standard or generic link field. For further details, see [Customizing a Control](#) on page 192.
2. Select the desired **Mode**. For further details, see [Field Attributes](#) on page 221.
3. Define a `Select-Function` `selectRecord` for the selected field.

The available functions and their options are described in the following topics.

### selectRecord for Standard Link Fields

Learn about selectRecord function for Standard Link Fields.

#### Syntax

Use the syntax described in the article “How to Use Record Selector” at <https://support.aurea.com>.

**Note:**

- Use XML names in (double) quotes for specifying fields.
  - To copy the text value of a catalog, use "" and prefix the field number with a "T" e.g. "T5".
  - If defined, the search context is displayed in a drop-down list on the search mask. For further details, see [Search Context](#). Use the same syntax as described for the `SearchContext` control attribute. For further details, see [SearchContext](#) on page 203.
  - When searching in the same info area as the current record (e.g. the **Predecessor** fields in KM and IB), the search context is automatically created, even if not specified in the `Select-Function`: The drop-down list contains "Entire database" plus the parent record(s) of the current record).
  - `filterFields` parameter: Filters on decoded fields (Z-fields) are not supported.
  - Use the **SDK** page to test your selectors: **Selectors > Select Records**
- 

**Example**

**Linking a marketing activity record to an activity (-> Marketing Activity field in MA)**

```
selectRecord({
  from: {
    infoAreaId: "AK"
  },
  target: {
    addLink: true
  }
});
```

## Selecting a Postal Code when creating/editing a company record (Postal Code field in FI)

```
{
    from: {
        infoAreaId: "OR",
        searchAndListName: "OR-Link",
        filterFields:
        {
            Country: {
                source: "target",
                fieldId: "Country"
            },
            FromPC: {
                source: "target",
                fieldId: "ZipCode",
                compareOperator: "<="
            },
            ToPC: {
                source: "target",
                fieldId: "ZipCode",
                compareOperator: ">="
            }
        },
        target:
        {
            copyFields: {
                Country: "Country",
                ZipCode: "FromPC",
                City: "City",
                AreaCode: "TelPrefix"
            }
        }
    });
```

This definition uses a specific Search control definition ("OR-Link"). After selecting a record the values for Postal Code, City, Country and Area Code are copied from OR to the corresponding fields in FI (copyFields).

### Selecting an Activity template with Live Calculations (Using the delayChangeNotification parameter)

The following sample shows the usage of the delayChangeNotification parameter. This is useful when the copyFields is used with Live Calculations. The field values are evaluated everytime the field change event is fired and overwrites the values already set in the previous fields. This parameter ensures that the change event is triggered after all the fields are set.

In the sample shown below, default-values trigger is used in MA that sets the start of the activity to current day and time and sets the end time of the activity after a duration of one hour. It also uses a Activity Template (A031) that uses a duration that is different from the default value trigger. By using the `delayChangeNotification` parameter, you can ensure that when a new activity is created using the template, the end time of the activity is set according to the template and not the default value trigger.

```
selectRecord({
  from: {
    infoAreaId: "A031",
    useAutoSearch: true
  },
  target: {
    delayChangeNotification: true,
    addLink: true,
    copyFields: {
      "6005": "6002",
      "3": "7001", //start at
      "6": "7002", //end at
      "4": "F7004", //duration
      "5": "7005" //unit,
    }
  },
  findAsYouType: {
    searchFields: [6000],
    textFields: [6000, 7001, 7002]
  }
});
```

### **FS version - Selecting a bank account for a product (Bank A/C Details field e.g. in KT)**

```
selectRecord({
  from: {
    infoAreaId: "BV",
    searchContext: {
      items: ["FI"]
    }
  },
  target: {
    copyFields: {
      "SortCode": "SortCode",
      "Institute": "Institute",
      "Branch": "Branch",
      "AcNo": "AcNo",
      "BankDetailsStatNo": "StatNo",
      "BankDetailsSeqNo": "SeqNo"
    }
  }
});
```

Only bank accounts linked to the same company as the product can be selected (searchContext).

## selectRecord for Generic Link Fields

Learn about selectRecord function for Generic Link Fields.

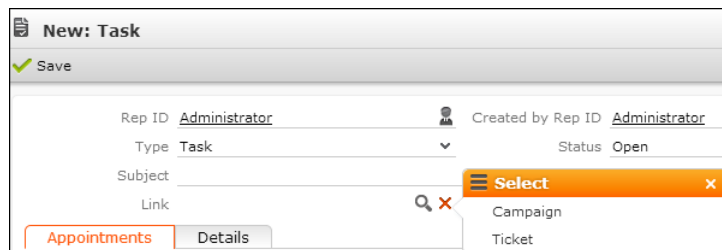
Defining a `selectRecord` function for a generic link field provides the user with a context menu where he can select the info area from which to pick a record.

1. Use the `infoAreaIds` parameter to define the content of the context menu. For further details, see [Syntax](#).

**Example:** the **Link** field in the **Task** info area (A1) (as defined in UPDATE\_DEFAULT):

```
selectRecord({
  from: {
    infoAreaIds: [
      "CM",
      "KM"
    ]
  }
});
```

This defines the following context menu (using the standard table captions for the menu entries' labels):



The user can link a task to records from the **Campaign** or **Ticket** info area.

Generic link fields with no `selectRecord` function defined display a context menu listing all available info areas. Use the `RecordSelector.GenericInfoAreas` Web Configuration parameter to define a subset of info areas globally. For further details, see [RecordSelector.GenericInfoAreas/RecordSelector.InfoAreasWithGenericLinks](#).

**Note:** The context menu only lists info areas that meet the following conditions: \* the info area's key is **StatNo+SerNo**, \* the user has read access, \* a Search&List configuration exists.

## Defining findAsYouType Functions

Learn about findAsYouType Functions.

FindAsYouType enables users to directly type a search string into link fields. The matches are displayed as a drop-down list. By selecting an entry the records are linked according to the settings specified in the Select-Function.

You can define `Select-Functions` with `findAsYouType` for standard link fields. In addition, you can define global `findAsYouType` functions per info area. For further details, see [Global Definition of FindAsYouType](#) on page 503. A global `FindAsYouType` definition is used for all link fields that point to the respective info area and for which no specific `Select-Function` has been defined.

---

**Note:** `FindAsYouType` only supports text, number and variable catalog fields (Field Types C,L,S,K, see [Data Types in Aurea CRM](#) on page 877) as search fields. These field types may be combined in one definition (as both `searchFields` and `textFields`).

---

**Note:** Fixed catalogs and hierarchical catalogs are not supported.

---

**Note:** Field types that are not supported are ignored and a warning is written to the client log.

---

**Note:** For the output fields (parameter `textFields`) there is no restriction: All field types are valid.

---

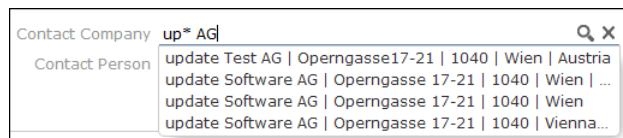
Text and catalog fields support a substring and wildcard search (e.g. `where (Country = "Au*")`). Number fields only return exact matches, i.e. searching for "100" does not return "1000" or "3100". Searching for number ranges requires the definition of a query, e.g. with condition `where (Revenue > '$parValue')`, see [Syntax](#) (parameter `query`).

### ***FindAsYouType for Standard Link Fields***

Learn about `FindAsYouType` function for Standard Link Fields.

To configure `FindAsYouType` for a standard link field:

In the Field Control Editor select a standard link field and define a `Select-Function` `selectRecord()` using the `findAsYouType` property. This provides users with the possibility to type a search string into the link field and select a record from the drop-down list that is then linked to the current record.



## Syntax

Use the following syntax, described in the article, “How to configure Find as you type” at <https://support.aurea.com>.

- parameter `searchFields` - valid field types: text, number and variable catalog fields (Field Types C,L,S,K, see [Data Types in Aurea CRM](#) on page 877).
- text and catalog fields support substring/wildcard search (e.g. `where (Country = "Au*")`)
- parameter `textFields` - valid field types: all - the output fields as shown in the search results drop-down list
- parameter `maxRows` - if more entries are found "..." is displayed at the end of the drop-down list
- parameter `query` - The name of the query to be used for `findAsYouType`. If this parameter is provided, the parameters `searchFields` and `textFields` are ignored. The output fields displayed in the drop-down list are determined by the query's output fields.
- The parameters `link` and `searchContext` are not supported for `FindAsYouType` functions.

---

**Note:** Link fields to which a (global or specific) `FindAsYouType` definition applies, are displayed as editable. If an invalid/erroneous `FindAsYouType` definition is applied to a field, this field is displayed as read-only and an error message is written to the client log.

---

## Example

Searching for a company to be defined for the **Contact Company** field in the **Ticket** info area.

```
selectRecord({
  from: {
    infoAreaId: "FI"
  },
  findAsYouType: {
    searchFields: ["Company", "Synonym", "Country"],
    textFields: ["Company", "Street", "ZipCode", "City", "Country"],
    maxRows: 20
  }
})
```

Typing part of the company's name, synonym or country lists the company plus its address details. The results in the drop-down list are limited to 20.

## Defining selectRecordWithCondition Functions

Learn about selectRecordWithCondition Functions.

Below is the full structure of the record select object.

```
selectRecord({
  from: {
    filter: "",
    filterFields: {},
    fixedFilterName: "",
    sourceCopyFieldGroupName: "",
    infoAreaId: "",
    infoAreaIds: [],
    link: {
      infoAreaId: "",
      recordId: "",
      linkId: 0
    },
  },
  relationName: "",
  searchAndListName: "",
  searchContext: {
    items: []
  },
  selectionMode: "",
  showFilters: false,
  useAutoSearch: true
},
target: {
  addLink: true,
  copyFields: {},
  freeInput: true,
  infoAreaId: "",
  links: []
}
});
```

Many properties are optional or only work in certain combinations. Please refer to the parameters sections below for details:

### from / object

This parameter is mandatory, and sets the SearchView and its behaviour up. It contains one or more of the subsequent properties.

```
selectRecord({ from: {} });
```

### infoAreald / string

This property (only) is mandatory and defines the info area to select records from. If the property 'searchAndListName' (see below) is not set, it is used for getting the SearchAndList-Configuration.

```
selectRecord({ from: { ... infoAreaId: "FI", ... } });
```

### infoAreaIds / string[]

This property can be used to define multiple InfoAreas to select records from. Alternatively, the property 'infoAreald' can be used the same way. Depending on the number of info areas, the user is able to select the info area from a context menu or from a popup list (if >5 info areas). After the user selected the info area, the corresponding search is shown.



```
selectRecord({ from: { ... infoAreaIds: [ // the user is asked to select
between a company and a person to select "FI", "KP" ], ... } });
```

### infoAreas / object[]

Similar to infoAreaIds this property can be used to define multiple info areas to select records from. However, this property allows users to specify the name of a special Search&List configuration to be used by the record selector.

```
selectRecord({ from: { ... infoAreas: [ // the user is asked to select be-
tween a company, a person and a ticket { infoAreaId: "FI", searchAndListName:
"MySpecialCompanySearch" }, { infoAreaId: "KP", searchAndListName: "MySpe-
cialPersonSearch" }, { infoAreaId: "KM" }, "PR" ], ..... } });
```

### link / object

Defines the parent record. If set, only records linked to this record can be selected.

```
selectRecord({ from: { ... link: { // Only records linked with linkId '0'
to the record defined in 'recordId' of infoArea 'FI' can be selected in-
foAreaId: "FI", recordId: "0x0123456789876543", linkId: 0 }, ... } });
```

### searchAndListName / string

The name of the SearchAndList-Configuration (as defined in update CRMesigner) to use for the selection.

```
selectRecord({ from: { ... searchAndListName: "KPSub", ... } });
```

---

**Note:** If not set, the above property 'infoAreaId' is used for retrieving a configuration.

---

### filterFields / dictionary

This parameter defines:

- which fields of the search form should be prefilled
- where do the values used for prefilling come from
- whether the user is allowed to change those fields

This is accomplished by defining the target field by its name or ID and assigning it a value. This value could be another fieldname or -number, which means that the content of this field in the source (e.g. the Expand View) is copied to the desired field in the target, the Search View.

Alternatively, instead of another fieldname or number, there is the possibility to apply a more complex object, telling the RecordSelector if e.g. a fixed text is used to prefill, or the content of a linked field.

```
selectRecord({ from: { ... filterFields: {
```

```
1: 2, // The value of the field with number '2' is copied from the ExpandView to
field '1' in the SearchView
```

```
2: "FieldName", // The value of the field with name 'FieldName' is copied from the
ExpandView to field '2' in the SearchView
```

---

```

FieldName1: "3", // The value of the field with number '3' ("Note": it is also possible
to write field-number as strings) is copied // from the ExpandView to field 'Field-
Name1' in the SearchView

FieldName2: "FieldName", // The value of the field with name 'FieldName' is copied
from the ExpandView to field // 'FieldName2' in the SearchView

FieldName3: { // The text 'FixedText' is copied to field 'FieldName3' and set readonly
value: "FixedText", // The text to copy
isFixed: true // If true, the user cannot edit this field in the SearchForm
},

4: { // The field 'FieldName' of the ExpandView is copied to field '4' of the
SearchView and set readonly

source: "target", // The field is copied from source 'target', what is (mostly) the
ExpandView

fieldId: "FieldName", // The name or number of the field to copy.
// (note: you can also use the dot notation to specify linked field ids
// eg: FI.2 or FI.Company. Note that this linked field must be on the expand form
which the field is read)

isFixed: true,

compareOperator: "=" // The operator used to identify which field in the search
control receives the value.

// A field with this fieldId and operator must already exist in the search control used
by the selector,
// except for the equals operator. if not specified, the "=" is used. Equals operator
matches against
// either a fieldId with no operator or a fieldId with "=" operator in the search control.
} // "Note": If property 'value' is not set, 'source' and 'fieldId' are mandatory!

"FreeC2": { source: "target", fieldId: "FI#2.Company" } // The Company field
from the second FI record associated with the current record
// has its Company field contents set in the FreeC2 search field.
// The number after the # is the link id.

}, ... });

```

8.2 The filter values in filterFields now overrides the values remembered from a previous search. To disable this behavior you can set property important to false.

```

selectRecord({ ... filterFields: { "FreeD1": { fieldId: "Revenue", important:
false } } ...

```

## searchContext / object

If a search context is provided, users of update.CRM web can select whether they want to search the whole database or only in the current context. Searching in the current context only returns results that are linked to a link record of the current record. If configured accordingly, users get a drop-down list in the search form where they can select the context (in which the last item of this list is always the entry to be selected first).

```
selectRecord({ from: {
searchContext: { items: [ "FI", // show only records linked to the company (using
the default link)
{ infoAreaId: "KP", linkId: 2 }, // show only records linked to the person linked
with linkId 2.
{ parentInfoAreaId: "KP", infoAreaId: "FI" }, // show only records linked to
the company going over the indirect person link
null // show records from the entire database
} } ... } });
```

---

**Note:** the special setting of parentInfoAreaId can be used if there is no direct link in the context for the desired info area, but it can be retrieved as a link of the linked record.

---

## selectionMode / string

Defines how records can be selected. Possible values are:

- Single: Select a single record, close the Search popup afterwards, and continue with copying values, if specified.
- Multi: Select one or more records, close the Search popup afterwards, and continue with copying values, if specified.

```
selectRecord({ from: { ... selectionMode: "single", ... } });
```

## useAutoSearch / Boolean

If set to true, the search starts immediately after the Search View is completely created.

Default value is false.

```
selectRecord({ from: { ... useAutoSearch: true, ... } });
```

## showFilters / boolean

If set to false, there is no possibility to define and/or use filters in the Search View.

Default value is true.

```
selectRecord({ from: { ... showFilters: false, ... } });
```

**filter / string**

Pre-sets a filter in the Search View, but as a 'suggestion' which can be changed by the user.

```
selectRecord({ from: { ... filter: "FI.MyCompanyFilter", // The name as defined
in designer or web ... } });
```

**fixedFilterName / string**

Presets a filter to the Search View which cannot be changed by the user.

```
selectRecord({ from: { ... fixedFilterName: "FI.MyCompanyFilter", // The name as
defined in designer or web ... } });
```

**sourceCopyFieldGroupName / string**

Defines the Field Group from which the List control is used for determining which fields to read from the target record to create additional filter parameter values.

```
selectRecord({ from: { ... sourceCopyFieldGroupName: "FI.MyCompanyFieldGroup", ...
} });
```

**relationName / string**

Defines the relation to another record. Used with parameter 'link'. TODO

```
selectRecord({
  from: {
    ...
    relationName: "$Parent"
    ...
  }
});
```

**target / object**

This parameter is optional and defines how the selected record(s) is/are treated. It contains one or more of the subsequent (optional) properties.

```
selectRecord({
  from: {
    ...
  },
  target: {
    ...
  }
});
```

**infoAreald / string**

Defines the info area of the record currently active in the Expand View.

---

**Note:** Do not set infoAreald for selectors defined for link fields! Setting infoAreald on target tells the selector engine to ignore the link of the field for which the the

selector is defined. As a consequence the record reference string is not copied into the target field.

```
selectRecord({
  from: {
    ...
  },
  target: {
    ...
    infoAreaId: "KP",
    ...
  }
});
```

### copyFields / dictionary

Defines which fields of the selected record(s) are copied to the ExpandView.

The syntax and definition is quite similar to the property 'filterFields' of parameter 'from', except 'copyFields' does not support 'complex' definitions.

```
selectRecord({
  from: {
    ...
  },
  target: {
    ...
    copyFields: {
      1: 1, // The value of the field '1' of the selected record is copied to field
'1' in the Expand View
      2: "FieldName", // The value of the field 'FieldName' of the selected record
is copied to field '2' in the Expand View
      3: "$Reference", // The value of the evaluated reference text of the selected
record will be copied to field '3' in the Expand View
      FieldName1: "3", // The value of the field '3' (''Note'': Again,
field-numbers can also be strings!) of the selected record
// is copied to field 'FieldName1' in the Expand View
      FieldName2: "FieldName", // The value of the field 'FieldName' of the selected
record is copied to field 'FieldName2' in the Expand View
      FieldName3: "$Reference", // The value of the evaluated reference text of the
selected record will be copied to field 'FieldName3' in the Expand View
      4: { value: "your value" }, // The 'value' property will be copied to the
field '4' in the Expand View FieldName4: { value: "your value" }, // ...
    }, ... }
});
```

## copyInfoAreaFields / dictionary

Provides the ability to define info area related copyFields. Usually you need this if you have defined multiple infoAreaIds (or infoAreas) in from.

```
selectRecord({
  from: {
    ...
  },
  target: {
    ...
    copyInfoAreaFields: {
      PE: {
        // same object notation as for 'copyFields'
      },
      CP: {
        // same object notation as for 'copyFields'
      }
    }
  },
  ...
});
```

## target / object

### Defining selectFormat Functions

Learn to define selectFormat function.

The `selectFormat` function provides the user with the possibility to select one or more formats (e.g. phone profile, trigger, query, default values etc). Define a `SelectFunction` `selectFormat` for the desired field.

Example: In the **Telephone Profiles** field in the **Rep** info area (ID) users can select one or more phone profiles.

```
selectFormat({
  formatType: 48,
  includePublicFormats: true,
  selectionMode: "multi"
});
```

## Syntax

Name	Type	Description
<code>fieldId</code>	int or string	The unique field ID or XML name of the field to which the format name is copied. (default: <code>fieldId</code> for which this <code>formatSelector</code> is defined)
<code>formatType</code>	int	The number representing the type of formats to select from.  You can find the format numbers in the <code>mm_fixcat-no_custom.txt</code> file (created by starting <code>mmba.exe</code> with the <code>--writeres</code> parameter, >> <b>mmba.exe Parameters</b> in the <i>CRM.Core Administrator Guide</i> ): catalog no. 36 "Type (Format)".

Name	Type	Description
includePrivateFormats	bool	If true formats from the current user are shown. (default: true)
includePublicFormats	bool	If true formats from other users are shown. (default: false)
filter	string	The info area abbreviation for which a format should be shown (e.g. "MA" to list only formats defined for the <b>Activity</b> info area)
filterFieldId	int or string	The unique field ID or XML name of the field that contains the info area abbreviation by which to filter. Only applied when filter is not set. Only formats defined for that info area is available. (Example in UPDATE_DEFAULT: Used for selecting the correct default values format in <b>Question</b> (F2) field <b>Default Values</b> .)
formatName	string	The initially selected format(s). In case of multiselect the format names have to be semicolon-separated. (default: currently entered value of the field with id fieldId)
filterFormatName	string	An optional string used for filtering the format by name (wildcards are supported), e.g. "My*" to list only formats whose name starts with "My".
selectionMode	string	If set to "multi" multiple formats can be selected.

## Examples

To select a private query format (i.e. owned by the current user):

```
selectFormat({
  formatType: 32
});
```

To select private and public formats of type "DefaultValuesForInfoAreas" defined for the **Activity** info area (MA):

```
selectFormat({
  formatType: 36,
  includePublicFormats: true,
  filter: "MA"
});
```

To select multiple private and public formats of type "DefaultValuesForInfoAreas" defined for the info area entered in field no. 5:

```
selectFormat({
  formatType: 36,
  includePublicFormats: true,
  filterFieldId: 5,
  selectionMode: "multi"
});
```

## Defining selectInfoAreaId and selectFieldId Functions

Learn to define selectInfoAreaId and selectFieldId Functions.

The `selectInfoAreaId` and `selectFieldId` functions provide users with the possibility to select an info area and then a field from this info area (Examples in UP-DATE\_DEFAULT: C4, F2, Z5).

To implement this functionality define two `Select-Functions` for the desired fields:

- `selectInfoAreaId` allows the user to select an info area.
- `selectFieldId` allows the user to select a field from the info area specified by `selectInfoAreaId`.

### Syntax

Name	Type	Description
<code>selectInfoAreaId</code>	function	select-function
<code>from</code>	object	This parameter is optional and sets up the Info Area selection dialog and its behavior. It contains the following (optional) property.
<code>infoAreaId</code>	string	The info area ID of a parent info area. If specified only children of this info area can be selected.
<code>target</code>	object	This parameter is optional and defines how the selected info area is treated. It contains one or more of the following (optional) properties.
<code>fieldId</code>	int or string	The unique field ID or XML name of the field that receives the info area ID. If not present the ID of the field the Select-Function is defined for is used.
<code>fieldIdToClear</code>	int or string	The unique field ID or XML name of a field that is cleared when another info area is selected.

Name	Type	Description
<code>selectFieldId</code>	function	select-function



Name	Type	Description
<code>from</code>	object	This parameter is mandatory, and sets up the Field selection dialog and its behavior. It contains following properties.  <b>Note:</b> You must specify either <code>infoaAreaFieldId</code> or <code>infoAreaId</code> for the select-function to work.
<code>infoAreaId</code>	string	The ID of the info area to select the field from.
<code>infoAreaFieldId</code>	int or string	The unique field ID or XML name of the field containing the ID of the info area to select the field from.
<code>target</code>	object	This parameter is optional and defines how the selected field is treated. It contains the following (optional) property.
<code>fieldId</code>	int or string	The unique field ID or XML name of the field that receives the info area ID. If not present the ID of the field for which the Select-Function is defined is used.

## Example

To select an info area and write its ID to the field (defined for the **Info Area** field in the **Question** info area (F2)):

```
selectInfoAreaId({
  target: {
    fieldIdToClear:"Field"
  }
})
```

To select a field from the info area specified in the **Info Area** field (XML name "File"):

```
selectFieldId({
  from: {
    infoAreaFieldId:"File"
  }
})
```













## Table Captions

A table caption is a short identification for a record, like first name and last name for a person or the company name for a company.

Aurea CRM.Web uses table captions in Tree views, in processes, for calendar items, favorites, as header texts for Search and Expand pages using the `{RECORD}` placeholder, in the breadcrumbs, as parent link information on Expand pages and in `RecordListBox` widgets in forms.

To configure a table caption:

- Click **Info Areas** on the CRM.Designer main page on the **Views** area and select an info area.
- Alternatively, you can click **Table Captions** either on the CRM.Designer main page **Views** area or on the **All Info Areas** page.

Table Captions			
Name		Configuration	Description
KP	 	UPDATE_DEFAULT	
KP.Rel	 	UPDATE_DEFAULT	
KPEmailWarning	 	UPDATE_DEFAULT	
KPKMParentLink1	 	UPDATE_DEFAULT	
KPKMParentLink2	 	UPDATE_DEFAULT	
KPshort	 	UPDATE_DEFAULT	
		+ Copy from	▼


If no table caption is defined for an info area, the reference text defined in Aurea CRM win is used as a fallback. For further details, see [Link Fields and Reference Texts](#) on page 19.

## Configuring a Table Caption

You can configure the table caption to include an image and fields of your choice.

A table caption definition consists of an optional image, the fields to be included in the table caption and the arrangement of these fields in a textual context.


To configure a table caption:


1. Select a table caption or create a new one. For further details, see [Designing Configuration Units](#) on page 64.
2. Click  (**Edit**).


**Table Caption 'KP'**

✓ Save ✗ Delete

---







Image  

Prefix-Text  

Format-Text  

Description

Configuration

	Info Area	Link ID	Field
1.	 Person in Company (KP)		Form of address (1000) ▼
2.	 Person in Company (KP)		First Name (1003) ▼
3.	 Person in Company (KP)		Last Name (1002) ▼
	+ Person in Company (KP) ▼		

---

**Special definitions**

When these fields are blank  Use this format instead

+

---

**Referenced by**

Expand [KP](#), [KPMergeOverview](#), [KPTree](#)

Expand Parent Relation [no reference](#)

TreeView [AK](#), [AK.Report](#), [CM](#), [CM.Report](#), [FI](#), [FI.Report](#), [KP](#), [KP.Report](#)

3. Specify the **Fields** whose values you want to display in the table caption. The fields are numbered automatically. Enter these numbers in **Format-Text** as placeholders {1}, {2}, {3} etc.

To reorder fields, click  (**Drag&Drop**) in the respective row and drag it to the desired position.

If you use multi-line fields in table captions, the line breaks are automatically applied if the table caption is displayed in a header. Restriction: The **Company** field in FI only supports three line breaks. Additional lines are appended to the fourth line when displaying the table caption in a header.


You can add fields from the current info area and from linked parent info areas. If you want to use another than the default link, enable the **Link ID** check box and select/enter the desired link ID. For further details, see [Specifying Link IDs](#) on page 20

---

**Note:** Adding fields from parent info areas may lead to performance problems.

---

4. Specify the following:

- **Image** (optional): The name of the image displayed left of the table caption.
  - Click + (**Add**) to select a fixed image.
  - Click  (**Edit**) to define conditions for an image to be displayed, e.g. `CAT_AQ_{1}` Or `CAT_ABC_{javascript:($['1'] || '').toUpperCase();}`, 1 referring to the first field defined under **Fields**. The image is displayed based on the field's value. Use the same syntax as described for the `Large Image Template` field attribute. For further details, see [Large Image Template](#) on page 231.

---

**Note:** Conditional images are supported for `RecordListBox` and `LiveTile` widgets only.

---

The image(s) must be registered in CRM.Designer. For further details, see [Images](#) on page 717.

- **Prefix-Text** (optional): Used to display the record type, e.g. "Contact Person" for KP. The prefix text is used on Expand pages but not in the Tree view where the record type is defined by the tree's parent node.
- **Format-Text:** The text of the table caption using the placeholders {1}, {2}, {3} etc. for the fields specified under **Fields**. For further details, see [Table Caption Syntax](#) on page 268. Example: For the fields **Last Name** (3), **First Name** (2) and **Company** (4) define the **Format-Text** "{2} {3} of {4}" to display "John Doe of Aurea software" in the table caption for KP.

Define/re-use the multilingual CRM.Designer texts for **Prefix-Text** and **Format-Text**. For further details, see [Texts](#) on page 50.

- **Description:** Add an internal description (only visible in CRM.Designer).
5. You can add **Special definitions** (i.e. alternative table captions) in case one or more fields you use do not contain a value. Example: If a person record does not contain a **First Name** the **Format-Text** "{1} {3} of {4}" is used instead: "Contact Person Mr. Doe of Aurea Software".

Alternatively you can use the conditional placeholder {?} in the **Format-Text**. For further details, see [Table Caption Syntax](#) on page 268.

## Table Caption Syntax

Here you can learn to use Table caption syntax.

Use the following syntax:

- Use placeholders for field values: {<number>}.

Example: For the fields **Last Name** (1), **First Name** (2) and **Company** (3) define the **Format-Text** "{2} {1} of {3}" to display "John Doe of Aurea Software" in the table caption for KP.

 **Contact Person John Doe of Aurea software**

- Separation characters (space, colon, comma, semicolon, dash, parentheses etc.) are omitted if the previous and/or next field does not have a value.
- Use the Enter key to insert a carriage return.

If the widget using the table caption supports it, line breaks in the table caption text are preserved when the text is displayed in Aurea CRM.Web. Example - upcoming items in the calendar's sidebar, tooltips for items in the Resources view.

In tab titles and calendar items, only the first line of a multi-line table caption is displayed.

- Use the conditional placeholder {?} to cover different strings in case one or more field values are empty.

Example: Defining "{?}Address: {1} {2}, {3}\n{?}Phone: {4}" e.g. for FI results in e.g. "Phone: +43-1-87855" if the address fields are empty.

For more details and examples, see the article "Table Caption Syntax" at <https://support.aurea.com>.

## Special Table Captions (Naming Conventions)

This topic discusses the naming conventions for items using special table captions.

Some items use special table captions. Use the following naming conventions to define table captions for these items:

- `<InfoAreaId>.Associations`: Defines the contents of the tooltip displayed for the connecting lines (`PB.Associations`) and the labels of the company/person bubbles (`FI/PE/KP/CP.Associations`) in the **Select Affiliates** dialog. See [Affiliation View](#) on page 752. Fallback: `<InfoAreaId>`.
- `<InfoAreaId>.Calendar`: Defines the table captions for calendar items (Activities (MA), Absences (A3), Tasks (A1), To-Dos (TD)). Fallback: `<InfoAreaId>Calendar` - if this does not exist either, the default table caption `<InfoAreaId>` is used.

For appointments in the month view, the start time precedes the table caption:

 16:00 JF web G...

`<InfoAreaId>.Calendar` (and `<InfoAreaId>Calendar`) are also used for the items in the Resources view and in the Dispatching Dashboard (MA, A3).

- `<InfoAreaId>.CoI`: Defines the labels, tooltips and contents of the `RecordListBox` for the relationship graph (`CircleOfInfluence` widget). See [CircleOfInfluence](#) on page 403. Fallback: `<InfoAreaId>`.
- `<InfoAreaId>.CoI.Hierarchy`: The default table caption used for the hierarchical circle of influence sidebar is `PB.CoI.Hierarchy`. See [HierarchicalCircleOfInfluence](#) on page 426 .Fallbacks: `<InfoAreaId>.CoI`, `<InfoAreaId>`.
- `<InfoAreaId>.Graph`: Defines the table captions for items in the campaign graph (CM, AK, AN). See [Campaign Graph](#) on page 758. Fallback: `<InfoAreaId>`.
- `<InfoAreaId>.MarketingCalendar`: Defines the table captions for items in the marketing calendar (CM, AK, VR). See [MarketingCalendar](#) on page 432. Fallback: `<InfoAreaId>`.
- `<InfoAreaID>.Participants`: For each activity, the persons and companies participating are displayed in a separate list. See [ExpandOptions](#). By default, the text defined in the info area's default table caption (i.e. FI or KP) is used. To display different table captions in the participants list, define a table caption named `<InfoAreaID>.Participants` (for FI and/or KP). Fallbacks: `<InfoAreaID>Participants`, `<InfoAreaID>`.
- `<InfoAreaId>QuickLinks`: If a table caption `QuickLinks` is defined for an infoarea, this table caption is used instead of the default table caption, see [Favorites](#) on page 585.
- `<InfoAreaId>.Rel`: Defines table captions used for the individual nodes of the relationship hierarchy tree, e.g. `KPre1` for persons. See [RelationTreeView](#) on page 446. Fallback: `<infoAreaId>`.
- `<InfoAreaId>.Relation/<InfoAreaId>.ReciprocalRelation`: Defines which field(s) from the **Relationship level** info area are used in the preview text on the `Cre-`

ateRelation form. See [CreateRelation](#) on page 408. **BTB**: BL.Relation/BL.ReciprocalRelation; **OTC/FS**: B5.Relation/B5.ReciprocalRelation.

- <InfoAreaId>.ResourceView: Defines the table captions used for activities (MA) and absences (A3) displayed in the Dispatching Dashboard and Resources view. **Fallbacks**: <InfoAreaId>.Calendar, <InfoAreaId>.
- <InfoAreaId>.SerialEntry.<Name>: Defines the text in the Related Data area for quick add pages, e.g. FI.SerialEntry.Offer. See [SerialEntry](#) on page 454.
- <InfoAreaId>.SerialEntry.RelatedList: Defines the list items of the Related Data area for quick add pages, e.g. UP.SerialEntry.RelatedList.
- <InfoAreaId>.Sidebar: Defines the table captions for upcoming items on the Calendar sidebar (Activities (MA), Absences (A3), Tasks (A1), To-Dos (TD)). **Fallbacks**: <InfoAreaId>.Calendar, <InfoAreaId>.
- <InfoAreaId>.VisitPlanning: Defines the table captions for the Visit Planning page (defined for AF/KP/FI/PE/SV/SX). See [VisitPlanning](#) on page 479.
- Y2.BuyingCenter: Defines the table captions for suggested participants on the Buying Center Wall. See [BuyingCenter](#) on page 399.

## Alternate Field Labels









Use the Alternate Field Labels page to change field names for your configuration and its children.

Field names are defined in the data model. Use the Aurea CRM win **Maintenance** module to change field names for the entire database.

To define Alternate Field Labels:

On the CRM.Designer main page (in the **Views** area), click Info Areas.

1. Click **Info Areas** on the CRM.Designer main page in the **Views** area.
2. Select the desired info area.
3. Click  (**Edit**) in the **Alternate Field Labels** area. All fields of the selected info area are listed.

Field-Id	Field-Label	Current Definition
0	Form of address	
1	Title	
2	Last Name	 Surname 
3	First Name	
4	Internal Address	
5	Internal Name	
6	Tel 1	

4. Enter the desired field names. For further details, see [Designing Configuration Units](#) on page 64 and [Editing Texts](#) on page 180.
5. Click **Save**.

Field label changes are applied to **all** controls and all field lists (e.g. when selecting fields in the Query Editor).

Labels you define in individual control definitions overrule the alternate field label. For further details, see [Field Attributes](#) on page 221.

## Custom Field Labels and Inputs

Customize the field labels and field inputs using custom CSS classes.

You can customize the field labels and field inputs using custom CSS classes via the Extended options. The extended options available for field label and field input values are `labelCssClass` and `fieldCssClass`. The usage of these options are shown below:

```
{
labelCssClass: 'custom-label-class',
fieldCssClass: 'custom-field-class'
}
```

The following sample CSS shows the class implementation for the option classes defined above:

```
.custom-label-class
{
font-weight: bold;
}
.custom-label-class > div > span
{
color: red;
font-size: 15pt;
}

.custom-field-class,
.custom-field-class input,
.custom-field-class textarea
{
font-weight: bold;
color: blue !important;
font-size: 15pt;
}
```

## Configuring Line Breaks and Truncating Texts

Display texts with line breaks, truncated, or wrapped, in a list header and list row field of a search result or query result.

You can configure a list view to display text with line breaks, truncated, or wrapped, in a list header and list row field of a search result or query result.

To truncate text set the `List.TruncateTextsInLists` configuration parameter to true. For more informaton, see [List.TruncateTextsInList](#).


To wrap the texts in a list view header or row, perform the following steps:

1. Copy a custom CSS file to the `\web\styles\custom.theme` folder with the content shown below. In this case lets call it `custom.css`.

```
.multiLineCell
{
    white-space: normal !important;
}
```

2. Import the above CSS file in `update.css` located in the same location (`\web\styles\cus-`

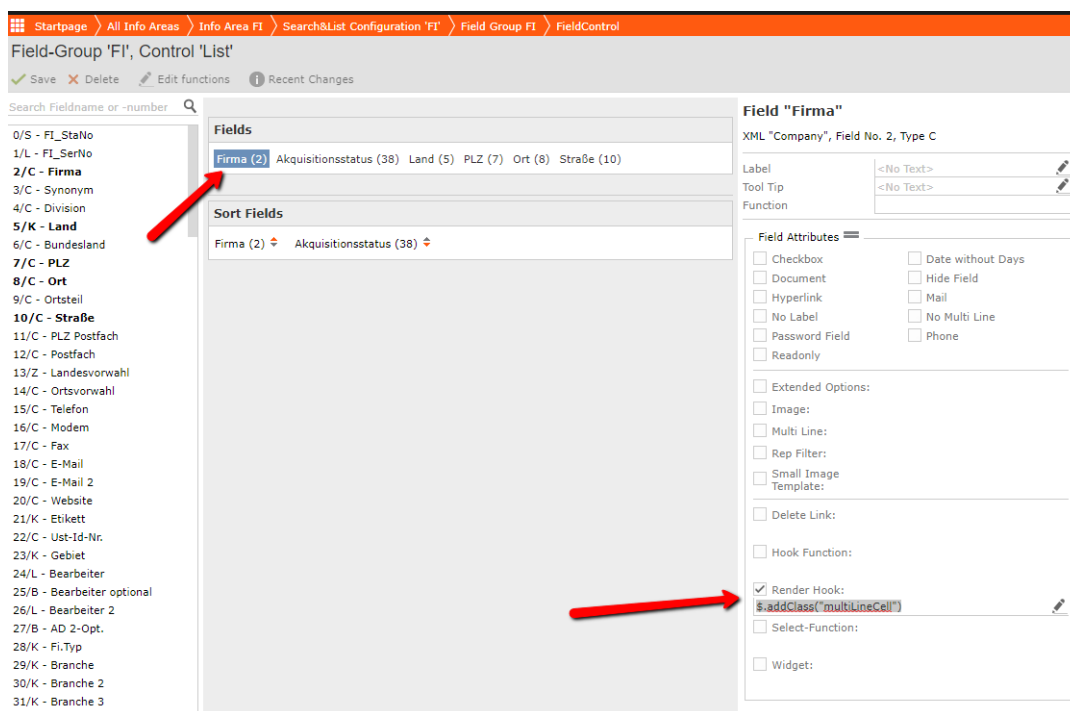
```
1 /*
2 | StyleSheet for Aurea CRM web
3 | Style name: "update"
4 | Styles used for generic items.
5 | Main style sheet: update.css.
6 | This color schema is free distributable as it is.
7 | No warranty for functionality.
8 */
9
10 @import url("browserReset.css");
11 @import url("base/base.css");
12 @import url("configuration/configuration.css");
13 @import url("crm/crm.css");
14 @import url("customer.css");
15 @import url("fonts.css");
16 @import url("../custom.theme/custom.css");
```



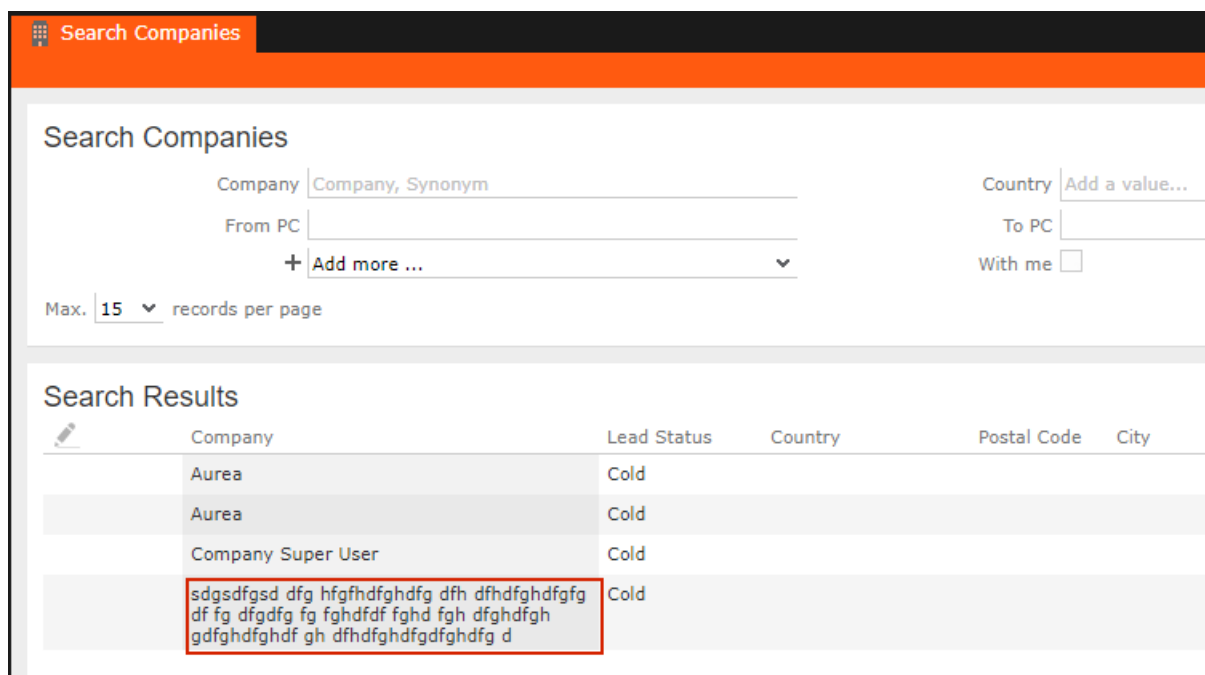
tom.theme).

3. In the CRM.Designer go to the Field Group for the target expand.
4. Select the header or row field whose text content should contain a line break.
5. In the panel on the right, check the Render Hook option, and add `$.addClass("multiLineCell")` in the expression box.





- Save the Field Group then invalidate the cache to apply the new configuration. The resulting field should have line breaks as shown in the sample screenshot below:



## Custom Links

Custom links are not supported anymore.

Update7 CustomLinks, defined in the file customLinks.xml, are obsolete and are not supported anymore.

Please use the following alternatives:

- [Generic Data Model Links](#)
- [Dynamic Filters](#)

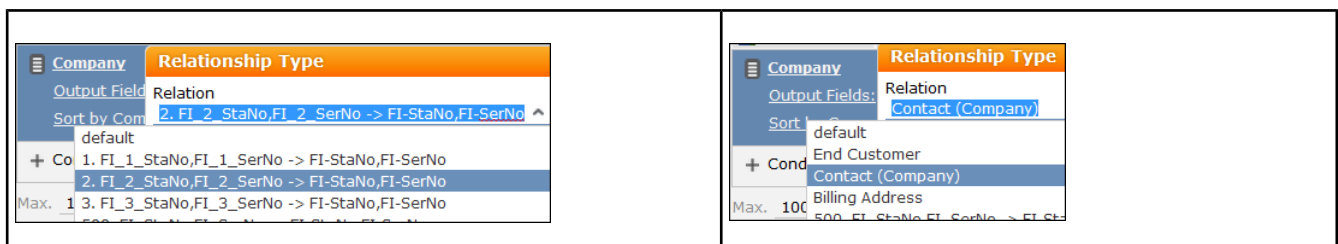
## Link Names

CRM.Designer allows you to define easily comprehensible descriptions for any link relation to be displayed in Aurea CRM.Web.

If more than one link to an info area exists, users need to select the link to be used. Example - when creating a query for that info area.

**Note:** Some info areas can have more one link to it. Example - The **Ticket** info area (KM) contains four links to the **Company** (FI).

Since the default syntax for link relations provided by the data model may be quite cryptic for average users, the **Link Names** page of CRM.Designer allows you to define easily comprehensible descriptions for any link relation to be displayed in Aurea CRM.Web. These names are used in filter and query definitions and in the **Modify List Fields** dialog.



To define link names:

1. Click **Info Areas** on the CRM.Designer main page **Views** area.
2. Search for and double click the desired info area.
3. Click (**Edit**) in the **Link Names** area. For further details, see [Designing Configuration Units](#) on page 64.

All available links are listed with their ID and relation type as defined in the data model. For further details, see [Link IDs](#) on page 19 and Data Model in the CRM.Core Administrator Guide.

Startpage	All Info-Areas	Info Area FI
		0 1:n
		1 1:n <span style="color: red;">✗</span> End Customer <span style="color: grey;">✎ ✗</span>
		2 1:n
		3 1:n
KM (Ticket)		127 Generic
		500 1:n

- Click **+** (**Add**) next to the links you want to define names for. For further details, see [Texts](#) on page 50.

## Info Area Labels

Use the Info Area Labels page to overrule the info area naming as defined in the Aurea CRM data model.

In Aurea CRM.Web the names of info areas are frequently displayed (e.g. in table captions or headers). You can use the **Info Area Labels** page to overrule the info area naming as defined in the Aurea CRM data model.

- Click **Info Area Labels** either on the CRM.Designer main page **Views** area or on the **All Info Areas** page.
- Scroll to the info area whose name you want to change.
- Click ✎ (**Edit**).

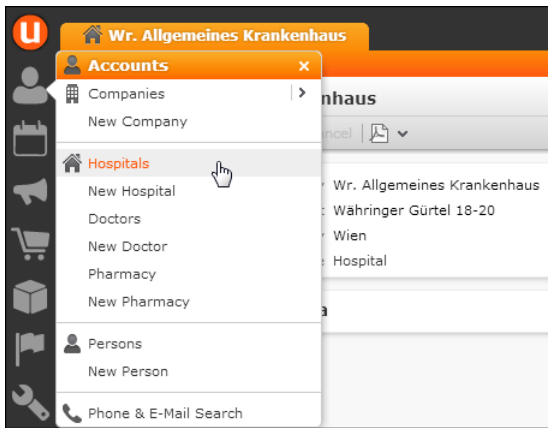
Infoarea	Tablename	Configuration	Singular	Plural
Languages	Languages	UserConfig_SUSANNA_Subst	<input type="checkbox"/> (from mmObjects)	<input checked="" type="checkbox"/>
Language/Station	Language/Station	not defined	<input checked="" type="checkbox"/> from mmObjects	from mmObjects

- Activate the check boxes to enter a new singular and/or plural name respectively. The changes are saved in the `net_infoareas_names` text group.
- Click on the `net_infoareas_names` link at the top of the page, to enter translations for your texts. For further details, see [Editing Texts](#) on page 180.

## Virtual Info Areas

Virtual info Areas are a virtual data model with virtual info areas used to filter physical info areas for records matching defined criteria.

In projects it is often the case that one info area is used for different purposes e.g. in the pharmaceutical industry, a record of the **Company** info area can represent a pharmacy, a hospital or a doctor's practice. These different types of "company" are usually distinguished based on a catalog value (e.g. **Co.Type**), and require different Expand or Tree views, sub-info areas, menu actions etc.



To meet those requirements you can create a "virtual data model" with virtual info areas used to filter "physical" info areas for records matching defined criteria. For further details, see [Creating Virtual Info Areas](#) on page 277. This provides an easy way to work with different representations of one info area in Aurea CRM.Web. For further details, see [Virtual Info Areas in Aurea CRM.Web](#) on page 282.

Differences between physical and virtual info areas:

Physical Info Area	Virtual Info Area
Exists in the Aurea CRM database (e.g. FI, KP, C005)	Does not exist in the Aurea CRM database but uses an existing physical info area
Is accessible for all clients	Is accessible only in Aurea CRM.Web and CRM.Designer
Is defined in the data model (in the Aurea CRM win <b>Maintenance</b> module)	Is defined in the <code>settings.xml</code> file

## Creating Virtual Info Areas

Create Virtual info areas to work with different representations of an info area in Aurea CRM.Web.

To create virtual info area you have to define them in the `<DataBase>` section of the `<update.crm.base>` section of the `settings.xml` file. The following example applies to the OTC version:

```
<update.crm.base>
...
<DataBase>
...
  <VirtualSchema>
    <InfoArea infoAreaId='VHOS' physicalInfoAreaId='FI' fieldId='28' fieldType='K'
value='Hospital' name='Hospital/Clinic' />
    <InfoArea infoAreaId='VDOC' physicalInfoAreaId='FI' fieldId='28' fieldType='K'
value='Doctor' name='Doctor' />
    <InfoArea infoAreaId='VPHA' physicalInfoAreaId='FI' fieldId='28' fieldType='K'
value='Pharmacy' name='Pharmacy' />
    ...
  </VirtualSchema>
...
</DataBase>
...
</update.crm.base>
```

The definition of a virtual info area consists of:

- a 4-digit info area ID, starting with "V" followed by three alphanumeric characters, e.g. `infoAreaId='VF11'`.
- the physical info area ID, defining in which database table the records of the virtual info area are stored, e.g. `physicalInfoAreaId='FI'`.
- the filter criterion defined for a field from the physical info area, determining which records of the physical info area belong to this virtual info area, e.g. `fieldId='5' fieldType='K' value='Austria'`.
- By default, virtual info areas use the XML name of their physical info area. Use the (optional) `name` attribute to define an individual XML name for the virtual info area, e.g. `name='CompanyAUT'`.
- To define a vertical-specific virtual info area, set the (optional) `coreMode` attribute to "vertical" or "false", e.g. `coreMode='vertical'`. For further details, see [Virtual Info Areas & Core Customizing](#) on page 876. If `coreMode` is undefined, the virtual info area is created in the Base vertical.

---

**Note:** If a virtual info area has been defined as "Base", you cannot revert it to "vertical" at any later time.

---

Use CRM.bulkloader to upload the virtual info area definition into the CRM.Designer database. For further details, see [Updating the Data Model](#) on page 775.

## Syntax

Syntax for creating virtual info areas.

There are two possibilities for defining the filter criteria for virtual info areas. The `fieldId` attribute has always to be defined.

---

**Note:** Supported field types for criteria definition: All except hierarchical catalog, array and rep fields, >> [Supported Field Types](#) on page 278.

---

## Long Syntax

You can define the filter criteria in an extra `<IdentFields>` section:

```
<VirtualSchema>
  <InfoArea infoAreaId='VHOS' physicalInfoAreaId='FI' name='Hospital/Clinic'>
    <IdentFields>
      <VarCatIdentField fieldId='142' fieldType='K' value='Hospital' />
      <VarCatIdentField fieldId='142' fieldType='K' code='1' />
    </IdentFields>
  </InfoArea>
</VirtualSchema>
```

## Short Syntax

You can define the filter criteria inside the virtual info area definition. The definition of the criterion consists of the id, type and value of the field of the physical info area, that is used to filter records for the virtual info area.

```
<VirtualSchema>
  <InfoArea infoAreaId='VHOS' physicalInfoAreaId='FI' fieldId='142' fieldType='K'
  value='Hospital' name='Hospital/Clinic' />
</VirtualSchema>
```

## Supported Field Types

Syntax for supported field type for virtual info areas.

To define filter criteria for virtual info areas, use the following syntax depending on the field type:

### Fixed catalogs

`FixCatIdentField` (long syntax) / `fieldType='X'` (short syntax) determine that the following field is a fixed catalog. To define the criteria, specify `code='<Catalog Code>'`.

- Long syntax:

```
<FixCatIdentField fieldId='...' code='Catalog Code' />
```

- Short syntax:

```
<InfoArea='...' fieldId='...' fieldType='X' code='Catalog Code' />
```

Alternatively you can specify the catalog value in the base language by defining `value='<value>'`.

## Variable catalogs

`VarCatIdentField` (long) / `fieldType='K'` (short) determine that the following field is a variable catalog. To define the criteria, you need to specify either the `value`, `code` or `extkey` of the catalog value.

- Long syntax:

```
<VarCatIdentField fieldId='...' value='Text in catalog base language' tenantNo='...' />
<VarCatIdentField fieldId='...' code='Catalog Code' />
<VarCatIdentField fieldId='...' extkey='External Key' />
```

- Short syntax:

```
<InfoArea='...' fieldId='...' fieldType='K' value='Text in catalog base language' tenantNo='...' />
<InfoArea='...' fieldId='...' fieldType='K' code='Catalog Code' />
<InfoArea='...' fieldId='...' fieldType='K' extkey='External Key' />
```

---

**Note:** The `tenantNo` attribute is optional (default = 0).

---



---

**Note:** Empty catalog values are not supported as criteria.

---



---

**Note:** Regardless of whether you use `value`, `code` or `extkey`, the text of each catalog value must be unique in each language (i.e. you must not have two entries with the same text in one catalog).

---



---

**Note:** Since the catalog code is station dependent, a virtual data model using codes for variable catalogs cannot be transported to other instances.

---

## Boolean

`BooleanIdentField` (long) / `fieldType='L'` (short) determine that the following field is a Boolean field.

- Long syntax

```
<BooleanIdentField fieldId='...' value='true/false' />
```

- Short syntax

```
<InfoArea='...' fieldId='...' fieldType='L' value='true/false' />
```

## Other

`IdentField` (long syntax) has to be defined for all other supported field types. When using the short syntax no field type definition is necessary.

- Long syntax

```
<IdentField fieldId='...' value='...' >
```

- Short syntax

```
<InfoArea='...' fieldId='...' value='...' >
```

## XML Export of Virtual Info Areas

By default, exported XML contains the actual info area ID and XML name of virtual info areas.

To export the physical info area ID and XML name instead, define the Web Configuration parameter `Export.Xml.ExportPhysicalInfoAreaIds` (Type: character) and set its value to "true". For further details, see [Export.Xml.ExportPhysicalInfoAreaIds](#).

Example:

- Default XML export (`Export.Xml.ExportPhysicalInfoAreaIds` not defined):

```
<fieldtableshort="VDOC" tablename="Doctor" fieldno="2" fieldname="Company"
...>
```

- `Export.Xml.ExportPhysicalInfoAreaIds="true"`:

```
<fieldtableshort="FI" tablename="Company" fieldno="2" fieldname="Company"
...>
```

You can specify this option for individual report buttons by using the `Export.Xml.ExportPhysicalInfoAreaIds` export option. For further details, see [Export Options](#) on page 858.

## Virtual Info Areas in CRM.Designer

Information on the use of Virtual Info Areas in Aurea CRM.Designer.

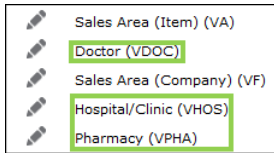
Use CRM.bulkloader to upload the virtual info area definition into the CRM.Designer database. For further details, see [Updating the Data Model](#) on page 775.

In CRM.Designer, virtual info areas appear exactly like physical info areas (i.e. with the same fields and links) and can be configured in the same way.

You can define the following designer units for virtual info areas:

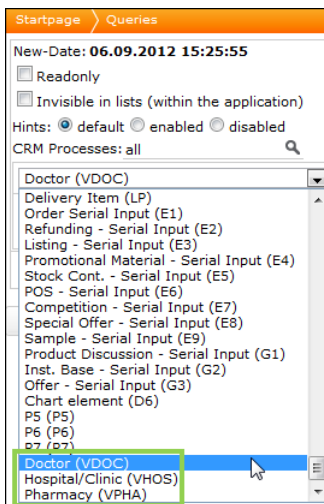
- Info Area Label (used e.g. in headers) in singular and plural
- Alternate Field Labels
- Expand configurations incl. info area default settings
- Table Captions
- TreeView configurations
- Search&List configurations
- Field Groups
- Headers





You can also use a virtual info area in all info area-dependent configuration units:

- Filters
- Queries
- Analyses
- Sub-lists
- TreeView branches
- Expand configuration: Parent Links and Breadcrumb Trail
- Timelines



All links except the "identity" link, i.e. the link to itself point to physical info areas, therefore at some locations (e.g. ) links to virtual info areas are not included in the default link info areas shown in CRM.Designer. Example - **Add Linked Fields**. In this case select the **show all** option to display them.

To display images for virtual info areas (e.g. in a header), register them as `INFOAREA_VXXX`. For further details, see [Images](#) on page 717. These images can be referenced by `{INFOAREAIMAGE}`. For further details, see [Header Attributes](#) on page 326.

## Virtual Info Areas in Aurea CRM.Web

This topic has information about using Virtual Info Areas in Aurea CRM.Web.

Virtual info areas are used in Aurea CRM.Web as follows:

Automatic Use of Virtual Info Area	Configurable Use of Virtual Info Area	No Use of Virtual Info Area (physical info area instead)
<ul style="list-style-type: none"> <li>• Expand Page</li> <li>• TreeView Page</li> <li>• Info Area Default Action</li> <li>• Info Area Default Menu</li> <li>• etc.</li> </ul>	<ul style="list-style-type: none"> <li>• Context menu in queries</li> <li>• Context menu in lists</li> </ul>	<ul style="list-style-type: none"> <li>• Table caption</li> <li>• Menus used in pages other than list or query</li> </ul>

The following sections describe the behavior of Aurea CRM.Web when virtual info areas are used.

### Search&List, SearchSubList, ListView Widget

The Search&List configuration defined in the Search action is applied as follows:

- If the Search&List configuration for a virtual info area is defined, only the records matching the respective filter criteria are displayed. Thus virtual info areas can be used instead of fixed filters.
- If the Search&List configuration for a physical info area is defined, all records of that info area are displayed regardless of whether they match a criterion of a virtual info area or not.

Example: A virtual info area VHOS for companies (FI) that are hospitals is defined. Searching in VHOS returns only hospitals, while searching in FI returns all company records, regardless of whether these are doctors, hospitals etc.

### Context Menu

The context menu of the list is either the menu explicitly defined in the Search&List configuration (e.g. `M_Company`) or the default menu matching the info area of the record. For further details, see [Search&List Configuration](#) on page 365.

If "Default" is defined for the menu in the Search&List configuration, the context menu for lists and sub-lists is determined according to the defined `virtualInfoAreaReadMode`. For further details, see [virtualInfoAreaReadMode](#) on page 285.

The context menu displayed for records in `SearchSubLists` is determined as follows:

1. The `SearchSubList` header contains `SubListParams` for all defined sub-info areas. The `SubListParam virtualInfoArea.ReadMode` defines the behavior of the context menu per sub-info area.
2. The Search action template contains `SubListParams` defining the behavior of sub-lists. The `SubListParam virtualInfoArea.ReadMode` defined here is applied to all sub-lists.
3. The Search action template input argument `virtualInfoAreaReadMode` defines the behavior for the result set and all sublists.
4. The Web Configuration parameter `virtualInfoArea.List.ReadMode` defines the context menu behavior for all lists.

## Filters

Virtual info areas are also used as filter namespaces. Therefore filters defined for the physical info area (e.g. FI) are not visible in a `Search&List` defined for a virtual info area (e.g. VDOC) and vice versa. For further details, see [filterNameSpace](#).

## MiniDetails

MiniDetails are always read from the configuration of the specified, physical or virtual, info area.

## Expand New

When creating a record in a virtual info area, the criteria defined in the virtual data model are automatically applied as fixed defaults and saved to the new record.

If the fields used as criteria are displayed on the Details control, they are automatically filled when the user selects **New**. However, the user may change those field values and therefore create records for other than the virtual info area. To prevent this, check the `Readonly` field attribute for those fields. For further details, see [Readonly](#) on page 227.

Example: A virtual info area VAUT for companies with `Country='Austria'` is defined. If a user creates a new record in VAUT, the **Country** field is automatically set to "Austria" and cannot be changed.

## Expand, TreeView

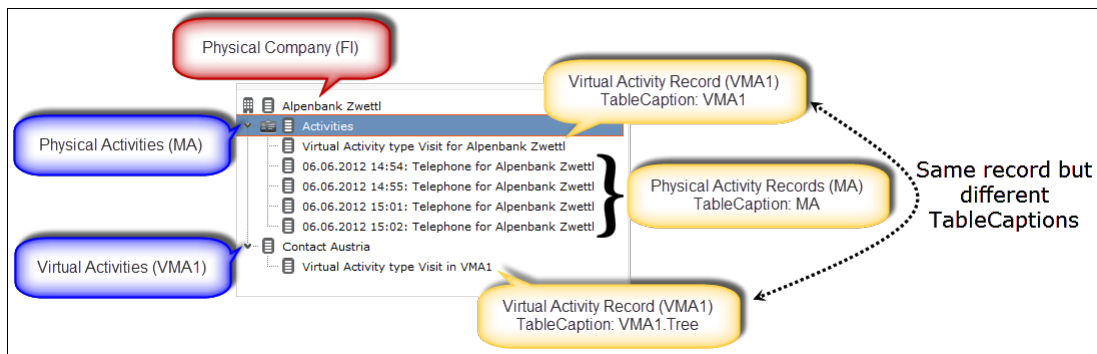
The Expand and TreeView action templates automatically determine whether there is a matching virtual info area and use it. All controls on an Expand/TreeView action template use the virtual info area as well. Therefore table captions and context menus are read from the virtual info area definition.

If no Expand/TreeView configuration for a virtual info area exists, the configuration of the physical info area is used as fallback.

If a record is changed in a way that the criteria don't match the virtual info area definition anymore, the Expand view falls back to the physical info area.

## TreeView table captions

If a TreeView contains nodes for both the virtual and the physical info area, records of the virtual info area are displayed in both nodes.



Each record (regardless of the node it is listed in) compares its info area with the info area defined in the TreeView configuration:

- If they match, the defined table caption is used (if no table caption is defined the base table caption is used).
- If the info areas don't match (i.e. for virtual info area records listed in the physical info area's node), the base table caption (e.g. VMA1) is used.

Thus if a physical info area lists records from both the virtual and the physical info area, they can be distinguished using different table captions.

## ExpandChildren

The context menu displayed for records in `ExpandChildren` is determined according to the defined `virtualInfoAreaReadMode`. For further details, see [virtualInfoAreaReadMode](#) on page 285.

1. The `ExpandChildren` header contains `SubListParams` for all defined child info areas. The `SubListParam` `virtualInfoArea.ReadMode` defines the context menu behavior per child info area.
2. The Web Configuration parameter `virtualInfoArea.List.ReadMode` defines the context menu behavior for all sub-lists.

## Favorites

Since favorites are stored in the Aurea CRM database, they necessarily contain only a reference to the physical info area. Therefore you can not explicitly manage favorites of a virtual info area: If you add such a record to your favorites, it is displayed both below the physical and the virtual info area.

Example: If you add a VDOC record to your favorites, it is visible for the FI favorites and for the VDOC favorites.

For favorites, the table caption of the configured info area is always displayed.

## Link Records

Aurea CRM.Web accepts link records from the physical as well as from one of the related virtual info areas.

## Queries

When defining queries, users can select virtual info areas just like physical info areas. You can even define separate field lists (i.e. `VXXX.Query` field groups) per virtual info area. For further details, see [Special Field Groups Naming Conventions](#) on page 190.

The context menus used for queries and for saved query results are determined by the current setting of the Web Configuration parameter `virtualInfoArea.Query.ReadMode` and/or the defined action template arguments. For further details, see [virtualInfoAreaReadMode](#) on page 285.

When saving a query in the Aurea CRM data base (Web Configuration parameter `Query.SaveTo` set to "Aurea CRM/both"), the query is automatically re-formatted to "physical info area + condition on criterion field"). Thus, queries defined with virtual info areas can also be used by CRM.server. For further details, see [CRM.Server](#) on page 15.

## Default Action, Default Menu

When the default menu or default action of a record is requested, the matching virtual info area's definition is used.

## virtualInfoAreaReadMode

Information on the use of Virtual Info Areas in ReadMode.

Use the `virtualInfoAreaReadMode` parameter to determine whether virtual info areas are used for context menus in the result rows of queries and lists.

This setting only applies, if the context menu of the Search&List configuration is set to "Default" (respectively undefined in `SearchSubList/ExpandChildren` headers and `ListView` widgets). Otherwise the explicitly defined menu is used. For further details, see [Search&List](#), [SearchSubList](#), [ListView Widget](#).

`virtualInfoAreaReadMode` has the following options:

- **Static:** Each results row has exactly the info area (i.e. the context menu and column header) that was requested in the search/query. In this case, each row has the same context menu.

Example: When you search for FI, the FI context menu (`M_Company`) is always displayed, even if the record returned matches the definition of a virtual info area.

- **Dynamic** (default): Each results row reflects the matching (virtual or physical) info area. Therefore each row may have a different context menu (since the **InfoArea** default menu is displayed).

Example: If you run a query for FI, the context menu of each FI record is the one matching the respective virtual info area.

The actual syntax for the `virtualInfoAreaReadMode` parameter depends on the below context:

## Web Configuration parameters

To define the global behavior of lists and queries use the following Web Configuration parameters:

### VirtualInfoArea.List.ReadMode

This parameter defines the default behavior for lists search results sets, `SearchSubLists`, `ExpandChildren` and `ListView` widgets in forms.

### VirtualInfoArea.Query.ReadMode

This parameter defines the default behavior for queries (query snapshot, saved queries, queries in forms and the `QueryRun` action template).

---

**Note:** These parameters are not part of `UPDATE_DEFAULT` but need to be created as Character or Combobox if required. For further details, see [Creating Web Configuration Parameters](#).

---

## Search Action Template

In addition to the `VirtualInfoArea.List.ReadMode` Web Configuration parameter there are three possibilities to define `virtualInfoAreaReadMode` in a `Search` action:

- Add a new parameter `virtualInfoAreaReadMode` to define the behavior for the result set and all sublists.
- Define `virtualInfoArea.ReadMode` in `listParams` to define the behavior only for the result list.
- Define `virtualInfoArea.ReadMode` in `subListParams` to define the behavior only for the sub-lists.

## SearchSublist / ExpandChildren Header

In `SearchSublist` and `ExpandChildren` headers the `SubListParam virtualInfoArea.ReadMode` defines the behavior of the context menu per child info area.

## Query Designer Action Template

There are two possibilities to define `virtualInfoAreaReadMode` in the `Query Designer` action template.

- Use the `virtualInfoAreaReadMode` input argument, to define the behavior for the result set and all sub-lists.
- Or define in `virtualInfoArea.ReadMode` in the `listParams` to define the behavior only for the result list.

## QueryRun Action Template

There are two possibilities to define `virtualInfoAreaReadMode` in the `QueryRun` action template.

- Use the `virtualInfoAreaReadMode` input argument to define the behavior for the result set and all sub-lists.
- Or define in `virtualInfoArea.ReadMode` in the `listParams` to define the behavior only for the result list.

## Query in Form

The `QueryResultPresenter` widget contains the parameter `virtualInfoAreaReadMode` to define the behavior of the query result list.

# Navigation Customizations

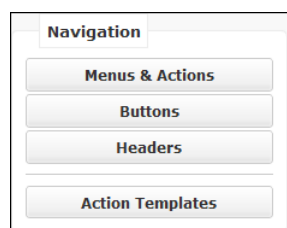
Use the **Navigation** area of CRM.Designer to customize access to pages and functions in Aurea CRM.Web.

You can configure the following:

- Menus and Menu actions, see [Menus](#) on page 290.
- [Buttons](#) on page 304
- [Headers](#) on page 321

Action templates are used to configure the actions called by menu items and buttons. For further details, see [Action Templates](#) on page 350.

Furthermore, actions are called within processes used to guide users to complex tasks in Aurea CRM.Web. For further details, see [Processes](#) on page 547.



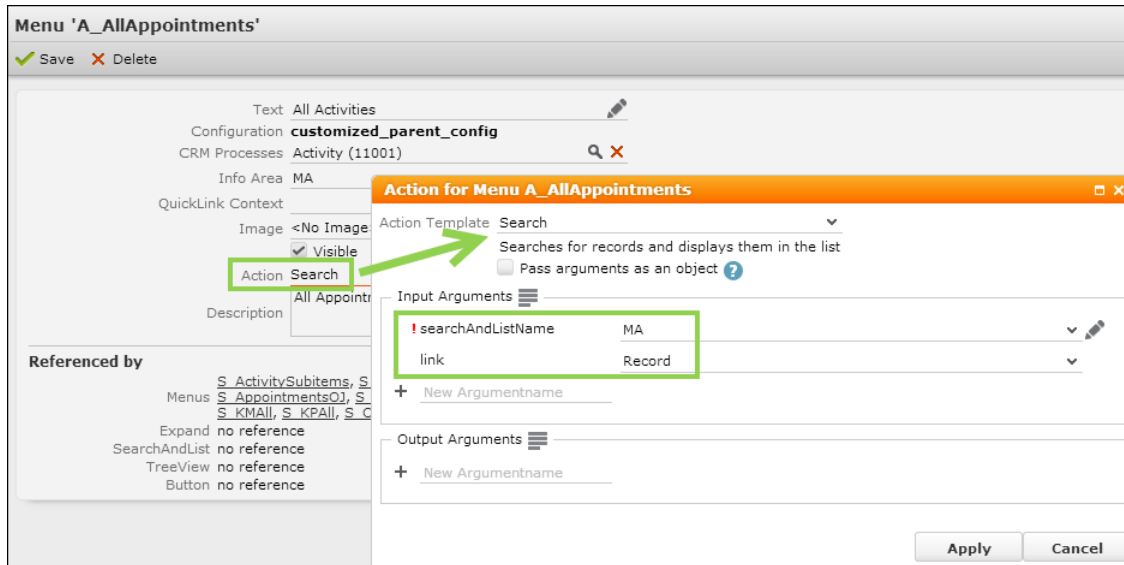
## Defining an Action

Configure the action templates specific to your need.

All actions carried out in Aurea CRM.Web are based on action templates, which need to be configured according to the specific action you want to execute. Action parameters (input arguments) can be mandatory (Example - the `searchAndListName` argument specifying the Search&List configuration of a `Search` action) or optional (Example - an optional link record for a search).

Actions templates are used for menu actions, buttons and within processes. Based on the action template definition, some actions can only be called for certain contexts. Example - only from the application menu or from within a process.

**Example:** The menu action `A_AllAppointments` (configured e.g. in the `S_KPAll` menu) lists all activities linked to e.g. a person record. It calls a `Search` action and passes the parent record (`link`) and a Search&List configuration (`searchAndListName`) to the page.



To define an action:

1. Click **Edit** in the **Action** field of the menu action/button/process step definition.
2. Select an **Action Template**.

The input and output arguments containing a default value are listed.

3. Use the **Show all arguments/Hide empty arguments** buttons to toggle between views and add new arguments.
4. Define the action's **Input Arguments**, e.g. the name of the Search&List configuration to be used for a `Search` action.

**Note:** For a list of valid input arguments, see the parameter lists in the Action Templates chapter, e.g. for the `Search` action. For further details, see [Search Action Parameters](#) on page 369.

5. If you are defining a process step, define the action's **Output Arguments**.

Predefined output arguments: Use `uid` and `uids` to get one or more selected records (e.g. in the `Search` and `QueryRun` actions), or use `uid` to get a newly created record from an `ExpandNew` action.



6. Click + (**Add**) to add custom arguments (both input and output arguments).
7. Click **Apply** and **Save**.

## Default Action

Use default action for a consistent behavior across the application.

You can define a default action used in Expand, List and Tree views providing a consistent behavior across the application when a user does one of the following:

- double-click on list or sub-list rows
- click on a link in a header
- click on a link in a record (in List, Expand, Minidetails view) and in the **Parent Data** area of Expand views

---

**Note:** Click and hold on a link in a record or in the **Parent Data** area of Expand views opens the QuickView. For further details, see [QuickView](#).

---



---

**Note:** Clicking on a link in a Rep ID field always opens the QuickView.

---

- double-click on calendar items
- click on a link to a record in an email (**Notify Rep** functionality)

The listed actions actually call the action `D_Default` which executes the default action according to the context (info area, Search or Expand view etc.).

An info area's default action is defined in the **InfoArea Default-Action** field in the info area's default Expand configuration (i.e. the configuration with the same name as the info area ID, Example - MA for activities), see [Expand Configuration](#) on page 352. The menu action specified there is used for all actions listed above except double-click on list or sub-list rows (see below).

If you do not specify an **InfoArea Default-Action** (i.e. "default" selected) for an info area, the menu action defined in the `Record.DefaultAction` Web Configuration parameter is used, see [Record.DefaultAction](#). If `Record.DefaultAction` is not defined, `A_Expand` is used, see [A\\_Expand](#).

For double-clicks in lists or sub-list, `D_Default` calls the default action as defined in the respective Search&List configuration. If undefined, the info area default action as defined in the default Expand configuration is used with the same fallbacks as described above.

To use the default action in a context menu, configure the menu action `A_ShowDefaultAction`, see [A\\_ShowDefaultAction](#).

**Note:** Do NOT specify `A_ShowDefaultAction` as an info area's default action or in the `Record.DefaultAction` Web Configuration parameter. This results in an infinite loop.

To automatically use the default action for single search hits, check the `Search.AutoExpandOnSingleHit` Web Configuration parameter or set `autoExpandOnSingleHit=true` as input argument for individual `Search` actions (Example - `ID_COMPANY_SEARCH`). For further details, see [Search.AutoExpandOnSingleHit](#).

You can define a common default action used for all sub-lists using the `List.SubList.DefaultAction` Web Configuration parameter. For further details, see [List.SubList.DefaultAction](#).

In Tree views double-clicking a record in the search results lists selects the node of the clicked record in the tree and loads the record without leaving the Tree view. Double-clicking a record in the tree actually calls the action `A_TreeDefault` executing the command "Tree/DefaultAction" that selects the tree node corresponding to the current list record. To globally change this default behavior use the `Tree.ListDefaultAction` Web Configuration parameter. For further details, see [Tree.ListDefaultAction](#). `A_TreeDefault` is part of `M_DefaultTree` which you can use as context menu for records in Tree views (Search&List configuration or sub-list).

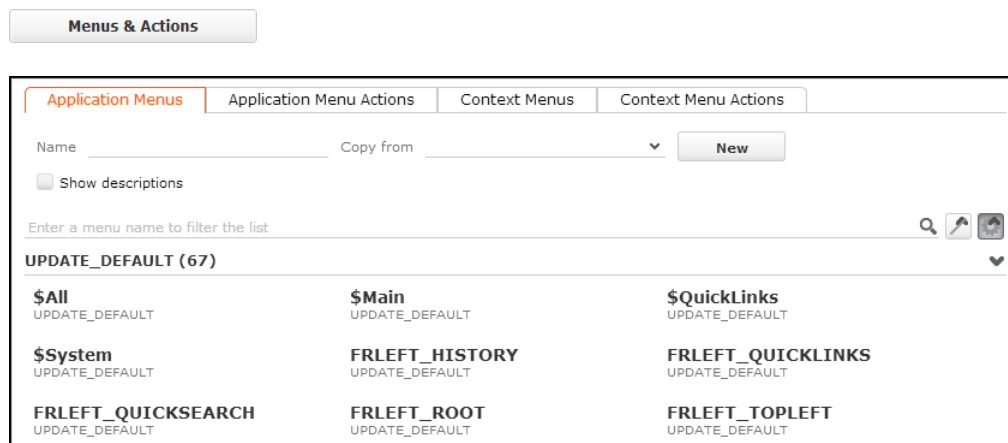
To use the default action for a `recordListBox` widget in a form, set the `command` property to `Actions/Default`. For further details, see [Forms](#) on page 621.

The `uid` of the selected record is automatically passed as command argument. Alternatively, you can define any `onDbClick` event.

## Menus

Use menu page to customize menus and menu actions.

The **Menus** page allows you to customize the menus and menu actions used in Aurea CRM.Web.



Aurea CRM.Web uses two different menu types:

- application menus (the menus and menu items) in Aurea CRM.Web's navigation pane. For further details, see [Application Menu](#) on page 291
- context menus displayed for specific records, info areas etc. For further details, see [Context Menus](#) on page 292.

---

**Note:** Re-using custom menus and menu actions defined in update.seven: You can create new menus and menu actions with the same name as an existing menu (action) if you copy the unit from an existing unit. E.g. if (in update.seven) you copied the entire application menu (`FRLEFT_NET`) to your custom configuration, you can overwrite Aurea CRM.Web's `$Main` by creating a new menu called `$Main` with **Copy From** set to `FRLEFT_NET`.

---

## Application Menu


Configuring Application Menu.

To configure Aurea CRM.Web's navigation pane:

1. On the CRM.Designer main page, click **Menus & Actions**.
2. Switch to the **Application Menus** tab.
3. Configure the desired menus and menu actions. For further details, see [Configuring a Menu](#) on page 292 and [Menu Actions](#) on page 295.

You can not define menu actions that need context information (i.e. a parent record) for the application menu. You need to configure a process instead (e.g. **New Activity**). For further details, see [Processes](#) on page 547.

You can configure the following items for the navigation pane:


- **\$System:** The Aurea CRM menu .
- **\$Main:** The contents of Aurea CRM.Web's application menu plus **CRM.phone** (if installed), **History**, **Favorites** (incl. Quick Links) and **Notifications**.

Customize the application menu by configuring sub-menus and menu actions. For further details, see [Configuring a Menu](#) on page 292 and [Menu Actions](#) on page 295.

---

**Note:** To hide a menu or menu item from the navigation pane, disable the **Visible** check box for this menu or menu action.

---

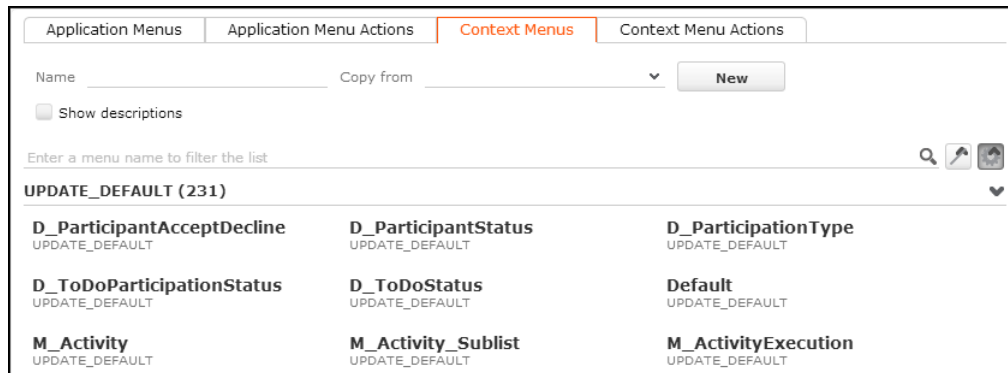
- **\$QuickLinks:** Configures the content of the Quick Links. Add the menu actions you want to display in the "My Quick Links" area of the **Favorites** panel (same functionality as  (**Modify Quick Links**) in Aurea CRM.Web).

## Context Menu

Configuring Context Menu.

To configure a context menu:

1. On the CRM.Designer main page in the **Navigation** area, click **Menus & Actions**.
2. Switch to the **Context Menu** tab.



3. Select the desired context menu or create a new one. For further details, see [Designing Configuration Units](#) on page 64 and [Configuring a Menu](#) on page 292.

You can extend existing context menus without the need to copy the parent definition to your configuration. For further details, see [Extending a Menu](#) on page 294.

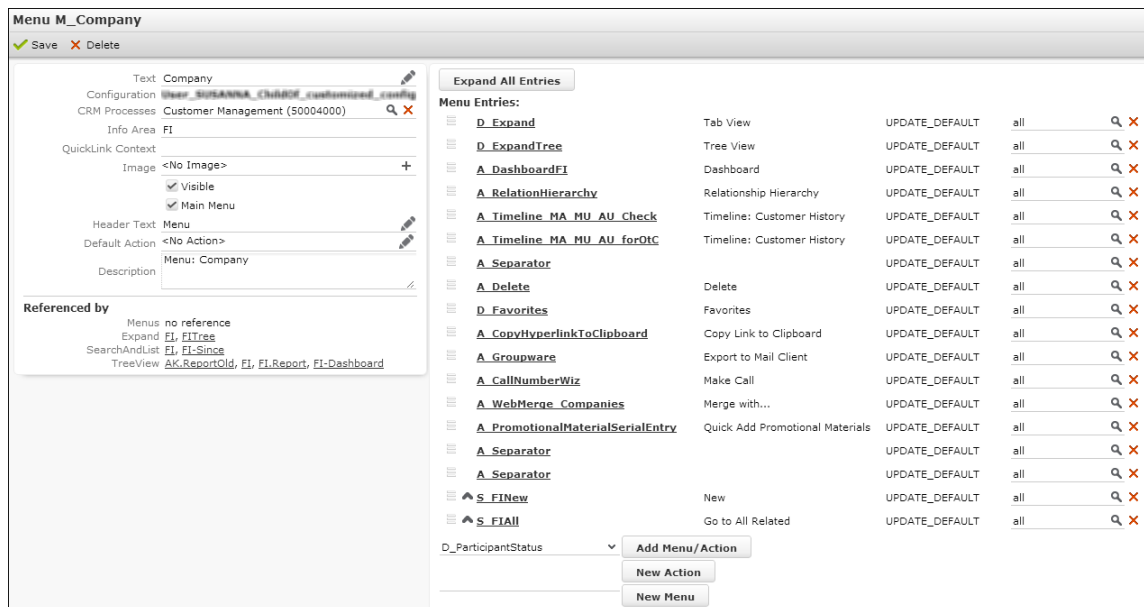
4. Add the context menu to the Expand, Search&List configurations etc. where you want it to be available.

## Configuring a Menu

Learn to configure a Menu.

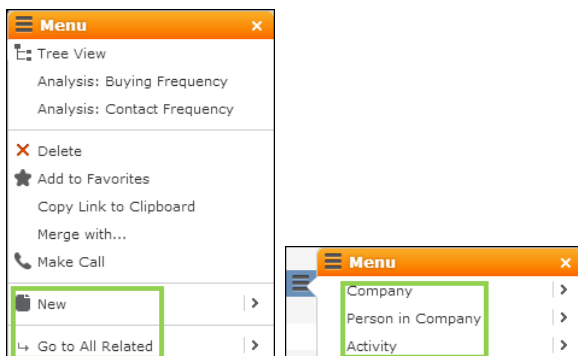
To configure a menu:

1. Select a menu or create a new one. For further details, see [Designing Configuration Units](#) on page 64.



2. Specify the following:

- **Text:** The menu's label. The text is (also) visible for sub-menus and when a menu is called from a list row (search results, query results).



**Note:** Without **Text** the menu is not displayed.

- **CRM Processes:** Assign one or more CRM processes to the menu. For further details, see [Roles & CRM Processes](#) on page 727.
- **Info Area:** Use this option to hide the menu depending on the current user's access rights (defined in the Aurea CRM win **Rights** module). Enter one or more info area IDs (separated by semicolon): If the current user has no access rights for the specified info area (**Deny Access**), the menu is hidden.
- **QuickLink-Context:** only used for menu actions
- **Image:** An optional image displayed left of the menu's label. The image must be registered in CRM.Designer. For further details, see [Images](#) on page 717.
- **Visible:** Shows or hides the menu in a parent menu.

---

**Note:** Even with Visible unchecked, the menu is still displayed where it is directly referenced by Aurea CRM.Web. Therefore, it is not recommended to set application root menus or main context menus to invisible.

---

- **Main Menu:** Only used in CRM.Designer. Checked for parent menus, unchecked for sub-menus.
- **Header Text:** The caption of a context menu, only visible for root menus, but not for sub-menus.
- **Default Action:** only used for menu actions
- **Description:** Add an internal description (only visible in CRM.Designer).

The right part of the page lists all sub-menus and menu actions currently configured for the menu.

3. Select a menu or menu action in the list below and click **Add Menu/Action** to add an existing menu or menu action.

To reorder the menu items, click  (**Drag&Drop**) in the respective row and drag the item to the desired position.

Use `A_Separator` to insert a separator line between menu items.

4. Enter a name and click **New Action** or **New Menu** to create a new sub-menu or menu action. For further details, see [Menu Actions](#) on page 295.

---

**Note:** While the new menu or action is immediately created, you must click the **Save** button before editing the new menu or action. Otherwise the reference to the new menu can be lost.

---

5. Click **Save**.

## Extending a Menu

Learn to extend a Menu.

Instead of copying an existing menu to your configuration, you can just extend the menu and thus make sure that changes to the parent configuration is reflected after the next update.

To extend a menu:

1. Select the desired menu from the **Application Menus** or **Context Menus** tab.
2. On the menu definition page click **Extend**.
3. Add the desired menu actions.

## Menu Actions

Learn to define Menu actions.

A menu action consists of an action template plus input arguments defining the action to be carried out when the menu item is selected, e.g. open a search page, create a new record, launch a process etc.

### Configuring a Menu Action

To configure a menu action:

1. Select a menu action from the **Application Menu Actions** or **Context Menu Actions** tab or create a new one. For further details, see [Designing Configuration Units](#) on page 64 and [Configuring a Menu](#) on page 292.

Referenced by	
Menus	S_FINew
Expand	no reference
SearchAndList	no reference
TreeView	no reference

2. Specify the following:
  - **Text:** The label of the menu action.

---

**Note:** Without **Text** the menu action is not displayed.

---

- **CRM Processes:** Assign one or more CRM processes to the menu action. For further details, see [Roles & CRM Processes](#) on page 727.
- **Info Area:** Use this option to hide the menu action depending e.g. on the current user's access rights. For a complete documentation of all possible settings, see the article "How to hide menu items" at <https://support.aurea.com>.

---

**Note:** Conditional access rights are **not** taken into account.

---



---

**Note:** To hide buttons depending on access rights, specify a menu action in the buttons configuration (instead of a page call). For further details, see [Configuring a Button](#) on page 305.

---

Enter one or more of the following values (separated by semicolon):

- `<InfoAreaId>`: Hides the menu if the current user has no access rights for the specified info area (**Deny Access**).
- `<InfoAreaId>:<new | update | delete | view>`: Defines which access rights are required for the menu action to be displayed. Separate multiple values by comma.

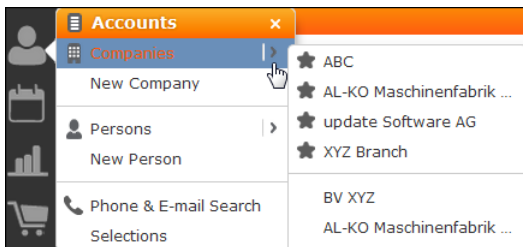
Example: `FI:update,new` defines that only users with the right to change and create companies sees this menu action.

- `INFO`: For general menu actions like `A_Delete` you can specify `INFO` or a `'.'` instead of `<InfoAreaId>` to reference the info area whose access rights definition is read.

For Expand configurations, the info area is determined based on the info area of the currently displayed record.

For Search&List configurations, the info area is determined based on the value specified in the Info Area field of the root menu, i.e. the menu that contains this menu action. Example: Adding `INFO:delete` to `A_Delete` hides the **Delete** menu item if its root menu (e.g. `M_Company`, `M_Person`, etc.) has **Info Area** defined.

- `-QuickView`: Hides the menu action from the QuickView. For further details, see [QuickView](#).
- **QuickLink-Context**: Enter the `<InfoAreaId>` of an info area, to display favorites and recently used records (i.e. the browsing history) from that info area as a sub-menu of the menu item.



If a table caption `<InfoAreaId>QuickLinks` is defined for the info area, this table caption is used for displaying the menu entries (fallback: `<InfoAreaId>`).

Records from virtual info areas are listed under both their physical parent and the virtual info area. For further details, see [Favorites](#) on page 585.

- **Image**: An optional image displayed left of the menu action's label. The image must be registered in CRM.Designer. For further details, see [Images](#) on page 717.
- **Visible**: Uncheck this option to hide the menu action in Aurea CRM.Web.
- **Action**: Define the action to be executed by selecting the menu item. For further details, see [Defining an Action](#) on page 287.
- **Description**: Add an internal description (only visible in CRM.Designer).



## Dynamic Menu Actions

Use dynamic menu actions in context menus to provide consistent user experience. Example: Using `D_Expand` and `D_ExpandTree` (instead of `A_Expand` and `A_ExpandTree`) displays the respective menu items depending on the context:

- Both **Tab View** and **TreeView** are visible for records in List view.
- **Tab View** (`D_Expand`) is hidden for a record in Expand view.
- **Tree View** (`D_ExpandTree`) is hidden for a record in Tree view.

In addition to these menu actions defined in `UPDATE_DEFAULT` you can use the following naming convention when defining dynamic menu actions: `*Open*In*<View>` where `<View>` may be either `ExpandView`, `TreeView` or `SearchView`. If a record is already displayed in the corresponding view, the menu action is hidden from the context menu. Examples: `A_OpenCompanyInExpandView`, `A_OpenRecordInCompanyTreeView` etc.

## Standard Menus and Menu Actions

Learn about the re-usable menu in Aurea.CRM.

`UPDATE_DEFAULT` contains a number of re-usable menu actions:

- standard menus used for various info areas. For further details, see [Standard Menus](#) on page 297
- common menu actions like `A_NotifyRep`, that can be used in any menu. For further details, see [Common Menu Actions](#) on page 298
- special menu actions like `D_CopyOfferToOrder`, that only make sense in a certain context. For further details, see [Special Menus and Menu Actions](#) on page 300.

---

**Note:** Some menus and menu actions are described in other chapters of this book.

---

## Standard Menus

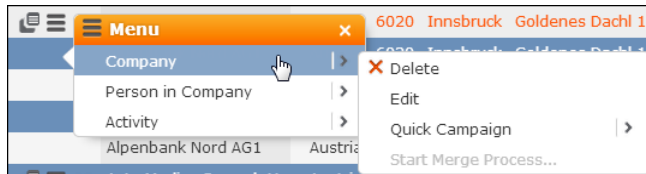
Learn to create customized multi select context menu.

### M\_MultiSelect

This context menu is displayed when multiple records have been selected in a list or sub-list. By default, `M_MultiSelect` contains the menu actions `A_MultiDelete` (for deleting the selected records) and `A_MultiEdit` (for editing multiple records). For further details, see [A\\_MultiEdit](#).

You can create a customized multi-select context menu named `XXX.MultiSelect` to be used for a specific Search&List configuration (`XXX` being the name of the standard context menu specified in the Search&List configuration). For further details, see [Search&List Configuration](#) on page 365. This context menu is then automatically used, e.g. `M_Company.MultiSelect` instead of `M_Company`.

If multiple rows are selected in a query result that contains more than one info area, a context menu containing all respective multi-select context menus is automatically created:



### Common Menu Actions

Find the list of Common Menu Actions in this topic.

Following are the common menu actions:

#### A\_All<infoAreaID>

These menu actions navigate to a linked search listing records of <infoAreaID> that are linked to the current record.

#### A\_Delete

Deletes the current record.

#### A\_Expand

Displays an Expand view of the current record.

#### A\_ExpandInNewTab

Displays an Expand view of the current record in a new tab.

#### A\_ExpandTree

Displays a tree view of the current record.

#### A\_MailMerge

This is the standard menu action for creating letters from records in Aurea CRM.Web. For further details, see [Mail Merge Configuration](#) on page 844. It calls the `MailMerge` action and creates a letter by merging record data into a document template. For further details, see [MailMerge](#) on page 430.

#### A\_New<infoAreaID>

These menu actions call an `ExpandNew` action for the respective info area.

#### A\_NotifyRep

Executes the default action for the current record, e.g. displays an Expand or Tree view. For further details, see [Default Action](#) on page 289.

## A\_ShowDefaultAction

Executes the default action for the current record, e.g. displays an Expand or Tree view. For further details, see [Default Action](#) on page 289.

## D\_Expand

Displays an Expand view of the current record. The `D_Expand` menu action is automatically hidden from the menu if the current view is an Expand view already.

In `UPDATE_DEFAULT` the dynamic menu actions `D_Expand` and `D_ExpandTree` are used in the `M_Company` and `M_Person` menus (instead of `A_Expand` and `A_ExpandTree`).

## D\_ExpandTree

Displays a Tree view of the current record. The `D_ExpandTree` menu action is automatically hidden from the menu if the current view is a Tree view already.

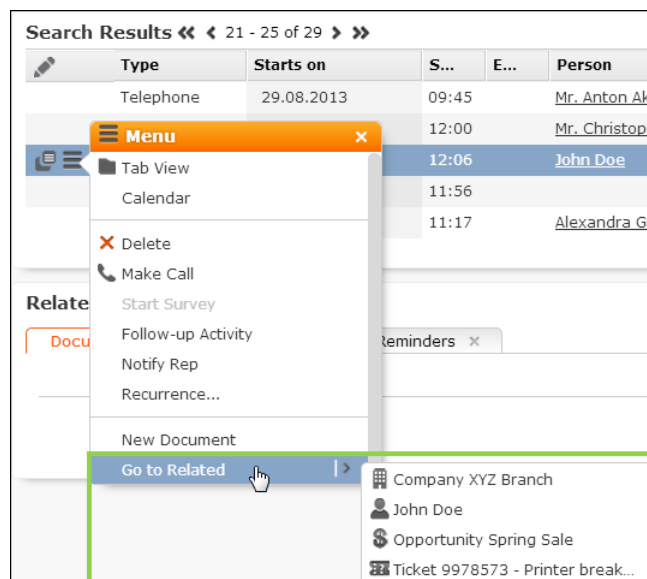
In `UPDATE_DEFAULT` the dynamic menu actions `D_Expand` and `D_ExpandTree` are used in the `M_Company` and `M_Person` menus (instead of `A_Expand` and `A_ExpandTree`).

## D\_OpenLinked

This dynamic menu action displays the parent record(s) of the current record analogous to the **Parent Data** area below an Expand view. For further details, see [ExpandOptions](#). This enables user to easily access parent records e.g. from within a list.

Which parent records are displayed in the sub-menu is determined by the Expand configuration of the selected record (**View Parent Relations**). For further details, see [Expand Configuration](#) on page 352.

Example: If `D_OpenLinked` is configured into the context menu of MA (`M_Appointments`), all parent records linked to an activity are displayed under **Go to Related**.



In `UPDATE_DEFAULT` `D_OpenLinked` is configured e.g. in `M_Relationship`, `M_Relationship_Expand` and `S_KPAll`.

## ID\_NET\_EMPTY\_TAB

Opens a new empty tab.

## Special Menus and Menu Actions

Find the list of Special Menus and Menu Actions in this topic.


Following are the Special Menus and Menu Actions:

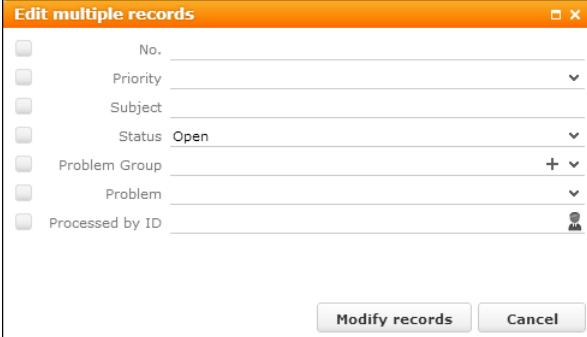
### A\_AnalysisDefaultAction

Defines the action that is executed when the user double-clicks on a result row on the **Analysis** search page.

Default = open and run the analysis.

### A\_MultiEdit

Opens the **Edit multiple records** dialog ("Bulk Editor"). Clicking  (**Edit**) in the list header when multiple rows are selected calls `A_MultiEdit` as well.



The **Edit multiple records** dialog displays all editable fields contained in the list. When called from a query result, the dialog displays all editable fields defined in the respective info area's default list control. Example - for companies the list control from the FI field group.

The following fields are not displayed in the **Edit multiple records** dialog:

- decoded fields (Z-fields)
- custom fields defined in `customFields.xml`
- fields flagged as VirtualCore (i.e. virtual fields with fixed IDs from 4200 to 4230, e.g. **Created by ID**). For further details, see [Fields with Fixed IDs](#) in the *CRM.Core Administrator Guide* (2nd table).
- fields defined as read-only in the data model or via field attribute.

Changing a field value automatically checks the check box. Unchecking the check box resets the field value. For hierarchical catalogs, the parent catalog is displayed as well. If both parent and child catalog are changed, resetting the change of either parent or child value resets the other value automatically.

In `UPDATE_DEFAULT` `A_MultiEdit` is configured in the `M_MultiSelect` and the `M_Company.MultiSelect` and `M_Person.MultiSelect` context menus. For further details, see [M\\_MultiSelect](#).

Aurea CRM uses transactions when editing multiple records in a list. For further details, see [Support for Database Transactions](#) on page 855. If an error occurs while editing multiple records (Example - a restriction due to a conditional field right), all changes are reversed (roll-back) and a message is displayed.

## D\_Campaign\_Approval

`D_Campaign_Approval` is configured in `M_Campaign` in `UPDATE_DEFAULT`. It allows the rep entered as **Budget Approved by ID** to accept/reject an event (budget approval).

- Selecting the sub-entry **Positive** results in the following:
  - **CM Budget approval on** is set to the current date.
  - **TD Participation Status** is set to "Accepted".
  - **TD Status** is set to "Completed".
- Selecting the sub-entry **Negative** results in the following:
  - **CM Budget approval on** is set to empty.
  - **TD Participation Status** is set to "Denied".
  - **TD Status** remains "Open".

This dynamic menu action is only displayed for campaign records with pending approval and for the specified rep.

For more details, see [Approving Events](#) in the *Web Business Logic User Guide*.

## D\_Campaign\_Verification

`D_Campaign_Verification` is configured in `M_Campaign` in `UPDATE_DEFAULT`. It allows the rep entered as **Verified by ID** to accept/reject an event (legal approval).

- Selecting the sub-entry **Positive** results in the following:
  - **CM Verified on** is set to the current date.
  - **TD Participation Status** is set to "Accepted".
  - **TD Status** is set to "Completed".
- Selecting the sub-entry **Negative** results in the following:
  - **CM Verified on** is set to empty.
  - **TD Participation Status** is set to "Denied".
  - **TD Status** remains "Open".

This dynamic menu action is only displayed for campaign records with pending approval and for the specified rep.

For more details, see [Approving Events](#) in the *Web Business Logic User Guide*.

## D\_CopyOfferToOrder

Copies an **Offer** (PR) plus **Offer Items** and **Participants** to an **Order** (PR). This dynamic menu action is part of the `M_Offer` context menu in BTB and OTC. It is displayed if the offer is linked to an FI record and hidden for offers linked to a PE record because offers must be linked to a company.

The menu action calls the JavaScript action template with the following input arguments:

- `$function: u8.services.businessObjects.copy`, see [u8.services.businessObjects.copy](#).
- `uid: Record`
- `destinationInfoAreaId: AU`

Similar menu actions: `A_CopyOfferToWR` (copies **Offers** to **Service Contracts** (WR)) and `A_CopyOfferFromServiceContract` (creates **Offers** from **Service Contracts**).

## D\_NewAppointment\_My(\_FI)

`D_NewAppointment_My` and `D_NewAppointment_My_FI` are configured in `M_CalNew` (**New Calendar Entry**) in `UPDATE_DEFAULT`. They allow creating an activity for the currently logged-in user or his company.

These menu actions are automatically hidden for the SU if no KP record is assigned.

## D\_Questionnaire

`D_Questionnaire` is configured in `M_Appointments` and `M_Tasks` in `UPDATE_DEFAULT`. It is only active if a questionnaire is actually linked to the activity (MA) or work order (AF) record.

For information on defining questionnaires and conducting surveys, see [Surveys](#) in the *Web Business Logic User Guide*.

For information on how to start a survey from another info area than MA or AF (e.g. Y1), see the article “How to configure Surveys for any info area” at <https://support.aurea.com>.

## ID\_NET\_CALC\_TERRITORYALLOCATION

In `UPDATE_DEFAULT` (BTB and FS) the menu action `ID_NET_CALC_TERRITORYALLOCATION` is configured in the `ID_NET_PROCESSES` menu (**Sales > Calculate territory allocation**).

`ID_NET_CALC_TERRITORYALLOCATION` creates a CRM.server to-do (Type = 12) to create/update the territory allocations. For further details, see [CRM.Server](#) on page 15.

For details on [territory management](#), see *Business Logic Manual*.

The Web Configuration parameter `Sales.TerritoryManagement.InfoAreas` controls which info areas are included in the calculation. For further details, see [Sales.TerritoryManagement.InfoAreas](#).

## ID\_NET\_CALCULATE\_ACCOMODATION\_VENUE\_EVALUATION

In `UPDATE_DEFAULT` the menu action `ID_NET_CALCULATE_ACCOMODATION_VENUE_EVALUATION` is configured in the `ID_NET_MARKETING` menu (**Marketing > Calculate accommodation & venue evaluation**). The menu action is only displayed if the Web Configuration parameter `UpdateServer.RepID` is defined. For further details, see [UpdateServer.RepID](#).

`ID_NET_CALCULATE_ACCOMODATION_VENUE_EVALUATION` creates a CRM.server to-do (Type = 10) calculating the average of all event ratings (accommodation and venue). For further details, see [CRM.Server](#) on page 15.

Use the Web Configuration parameter `UpdateServer.EventEvaluationStartDate` to define a campaign start date: Only campaigns that start after the specified date are taken into account. For further details, see [UpdateServer.EventEvaluationStartDate](#). If undefined, the average is calculated for all campaigns in the database.

For details, see [Evaluating Events](#) in the *Web Business Logic User Guide*.

## ID\_NET\_CALCULATE\_TOPIC\_LECTURE\_EVALUATION

In `UPDATE_DEFAULT` the menu action `ID_NET_CALCULATE_TOPIC_LECTURE_EVALUATION` is configured in the `ID_NET_MARKETING` menu (**Marketing > Calculate lecture subject & speaker evaluation**). The menu action is only displayed if the Web Configuration parameter `UpdateServer.RepID` is defined. For further details, see [UpdateServer.RepID](#).

`ID_NET_CALCULATE_TOPIC_LECTURE_EVALUATION` creates a CRM.server to-do (Type = 11) calculating the average of all event ratings (lecture and speaker). For further details, see [CRM.Server](#) on page 15.

Use the Web Configuration parameter `UpdateServer.EventEvaluationStartDate` to define a campaign start date: Only campaigns that start after the specified date are taken into account. For further details, see [UpdateServer.EventEvaluationStartDate](#). If undefined, the average is calculated for all campaigns in the database.

For details, see [Evaluating Events](#) in the *Web Business Logic User Guide*.

## ID\_NET\_STOCKTAKING

In `UPDATE_DEFAULT` (BTB and FS) the menu action `ID_NET_STOCKTAKING` is configured in the `ID_NET_PROCESSES` menu (**Sales > Stock Taking**).

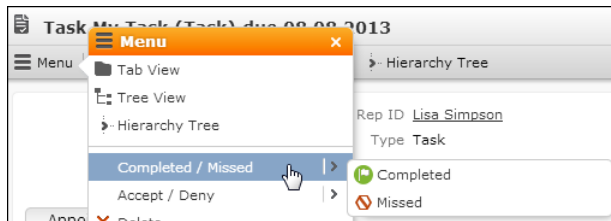
`ID_NET_STOCKTAKING` displays a dialog where the user selects a stock taking format and enters a time frame. Clicking **OK** creates a CRM.server to-do to create an inventory. For further details, see [CRM.Server](#) on page 15. For details on stocks and stock taking, see [Web Business Logic User Guide](#).

## ID\_NET\_TD\_BACKGROUND\_QUERIES

In UPDATE\_DEFAULT the menu action ID\_NET\_TD\_BACKGROUND\_QUERIES is configured in the ID\_NET\_QUERIES\_AND\_STATISTICS menu (**Queries & Analyses > Background Queries**). To allow users to access the to-do records of background queries easily via the application menu,

1. create a filter with condition "Rep/Group ID = <UpdateServer.RepID>"
2. assign this filter to the ID\_NET\_TD\_BACKGROUND\_QUERIES menu action.

## M\_Tasks



The context menu for Tasks (A1) M\_Tasks contains dynamic sub-menus and menu items allowing the user to change the status of **To-Do** records (TD) linked to a task.

The menu is displayed if the following conditions are met:

- **Create To-Do** is checked.
- The current rep is defined as required participant.
- The task's **Status** is "Open".

As soon as all to-dos linked to a task are set to "Completed", the task's status is set to "Completed" as well and the menu items are no longer displayed.

For details on the business logic, see **Changing the Status of a Task** in the *Web Business Logic User Guide*.

## Buttons

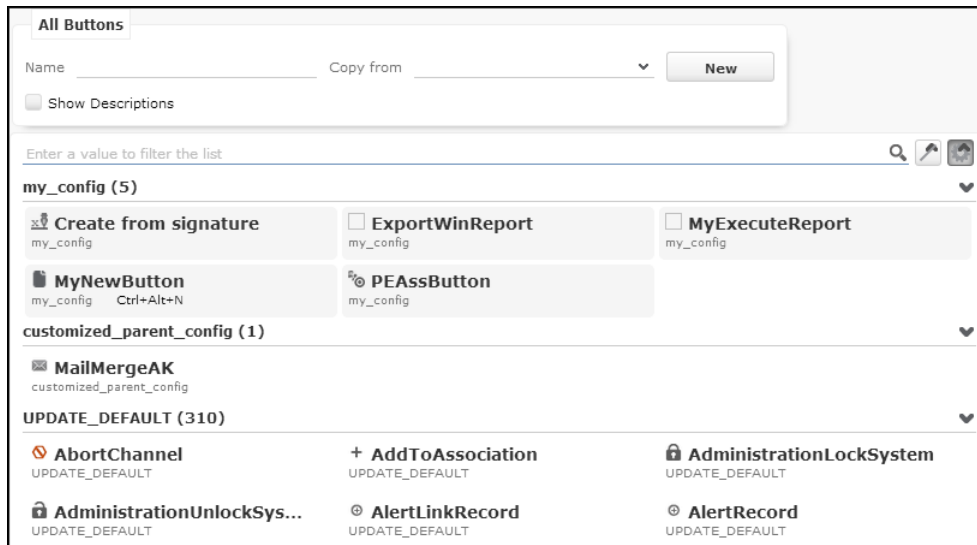
Learn to create buttons and manage them in CRM.Designer.

Like menu actions, buttons execute actions defined in CRM.Designer. You can either assign an existing menu action to the button or create a new action based on an action template. For further details, see [Configuring a Button](#) on page 305.

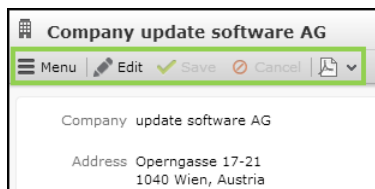
On the CRM.Designer main page, click **Buttons** to create and modify buttons used in Aurea CRM.Web.



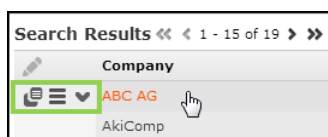




The **Buttons** page displays the button's icon and name, the configuration it belongs to and its hot-key, if defined. Buttons are usually configured into headers (see [Headers](#) on page 321 and [Header Buttons](#) on page 327) but can also be used in forms, see [Forms](#) on page 621.



**Note:** To determine which buttons are displayed in list rows you need to specify `listParams` in the respective action. For further details, see [ListParams](#) on page 374 (`hideControlColumns` | `showContextMenu` | `showMiniDetails` | `showQuickViewAction`).



## Configuring a Button

Learn to configure a button.

To configure a button:

1. Select a button or create a new one. For further details, see [Designing Configuration Units](#) on page 64.

The screenshot shows a configuration window for a button titled "Button 'New Offer to Ticket'". The fields are as follows:

- Label:** New
- Image:** New-Button (with a small icon)
- Action:** New Record
- Hot-Key:** Ctrl+Alt+N
- Query State Action:** <No Action>
- CRM Processes:** Offer (5050001)
- Help Text:** <No Text>
- Description:** Enter description here. (with a text area and a slash icon)

Below the description field are five checkboxes:

- Hide button in QuickView
- Hide button text
- Owner Draw Button
- Disable button
- Hide button

A "Save" button is located below the checkboxes. At the bottom, the "Referenced by" section shows "Header PRKMSublistNew.SubList".

The following settings are available:

- **Label:** The button's name either displayed as a tool-tip or as text on the button, depending on the header configuration. For further details, see [Header Attributes](#) on page 326 (**Buttons with Text** option).
- **Image:** The button's icon. The image must be registered in CRM.Designer. For further details, see [Images](#) on page 717.
- **Action:** Define the action to be executed by clicking the button.
  - Select **Menu Action** to call an already defined menu action. For further details, see [Menu Actions](#) on page 295.
  - Select **Action Template** to define a new action based on an action template. For further details, see [Defining an Action](#) on page 287.

---

**Note:** To show or hide buttons depending on the user's access rights, you need to specify a menu action that is configured accordingly. For further details, see [Menu Actions](#) on page 295 (**Info Area** option).

---

- **Hot-Key:** the hotkey or keyboard shortcut for activating the button.

You need to manually enter the hotkey definition in the **Hot-Key** textbox, since just pressing the respective key combination has no effect. Use the following syntax:

- `Ctrl/Alt/Shift` as modifiers
- `F2` to `F12` for function keys
- combine the individual elements with "+"

Examples: `Shift+Alt+G`, `Ctrl+S`, `F7`, `Shift+F6`, `Ctrl+#49`

The following hotkeys are not supported:

- F1 and any combination with F1 (Ctrl+F1, Alt+F1, Shift+F1)
- Alt+F4

Function keys with browser-specific functions (e. g. F3, F5 etc.) should not be used as well.

The defined keyboard shortcut is displayed in the button's tooltip.



**Note:** For buttons in groups no tooltip is displayed. For further details, see [Grouping Buttons](#) on page 328.

- **Query State Action:** Determines under which conditions the button is active. Specify a menu action containing an appropriate JavaScript function to flexibly control whether the button is enabled or disabled. If the JavaScript function returns `false` the button is enabled, if it returns `true` the button is disabled (grayed out).

Alternatively, an object in form `{disabled:<true|false>, done:<true|false>}` can be returned, where the `disabled`-property is interpreted as before and the `done`-property signals whether the periodic re-querying of the state should be stopped.

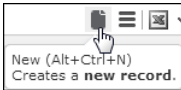
UPDATE\_DEFAULT contains the following standard JavaScript functions to be used with **Query State Action**:

- `C_OneRowSelected`: the button is enabled if exactly one row is selected.
- `C_OneRowSelectedOrNoList`: the button is enabled if exactly one row is selected or if used in an `Expand` or `Tree` header.
- `C_NoRowSelected`: the button is enabled if no row is selected.
- `C_RowsSelected`: the button is enabled if one or more rows are selected
- `Check_VirtualInfoArea`: the button is disabled if used in the header of a virtual info area (example in UPDATE\_DEFAULT: `DefaultValuesDefine` button).

In addition, you can use the `QueryStateBinding` action template to enable/disable buttons using bindings. Examples in UPDATE\_DEFAULT:

- `C_SocialInbox_CanAssignPerson` (used to enable/disable the `SE_AssignPerson` button)
- `C_SocialInbox_CanRespond` (used for the `SE_Respond` and `SE_Invite` buttons)

- **CRM Processes:** Assign one or more CRM processes to the button. For further details, see [Roles & CRM Processes](#) on page 727.
- **Help Text:** Add a description of the button's purpose. The **Help Text** is displayed in the button's tooltip (in a new line).



The default text group for button tooltips is `ButtonHelp`.

You can use the Markdown syntax to format your help text. For further details on Markdown syntax, see <http://en.wikipedia.org/wiki/Markdown>.

- **Description:** Add an internal description (only visible in CRM.Designer).
- **Hide button in QuickView:** If checked, the button is hidden from the QuickView. For further details, see [QuickView](#).
- **Hide button text:** The button is displayed without its label, i.e. only the icon is visible. A tooltip with the button text is displayed on mouse-over.
- **Owner Draw Button:** Do not modify! This option is checked for buttons that are displayed as drop-down lists in specific headers (e.g. the rep/filter set drop-down lists in the Calendar sidebar). This option is **for internal use only**: Do not change this setting.
- **Disable button:** If checked, the button is displayed inactive (grayed out).
- **Hide button:** If checked, the button is not visible for the users even if it is configured e.g. in a header. The next button to the right moves up and takes the place of the hidden button (thus avoiding a gap).

## 2. Click **Save**.

For information on how to configure **toggle buttons**, see the article “How to toggle buttons in a header” at <https://support.aurea.com>.

Toggle button examples in UPDATE\_DEFAULT: `Toggles(M_CalendarViews)`, `Toggles(M_AnalysisResultViews)` and `Toggles(M_MarketingCalendarInfoAreas)`.


## Standard Buttons

Find the list of standard buttons in this topic.

Below is the list of standard buttons:

### CancelEdit

Use this button to exit from edit mode of a form.

The `CancelEdit` button  allows users to switch a form from Edit to View mode. A dirty check is displayed allowing the user to return to Edit mode and save his changes. The button calls the `Command` action with the `Command` argument set to `Cancel`.

## Copy/Paste Record

Use these buttons to copy and paste records.

UPDATE\_DEFAULT contains a **Copy Record** and **Paste Record** button allowing users to copy a record and create a new one by pasting the data from the clipboard. To do this, add the **Copy Record** button to the `Expand` header and the **Paste Record** button to the `New` header of the desired info area. For further details, see [Standard Headers](#) on page 337.

The buttons calls the `Command` action with the `Command` argument set to `Clipboard/Copy` and `Clipboard/Paste` respectively.

## Copy/Paste Record and Access Rights

Depending on the defined access rights and field attributes pasting a record from the clipboard results in the following behavior:

The Paste Record button is deactivated if

- the Expand view is in View mode
- the clipboard does not contain a record from the same (base) info area
- the Expand view is in Edit mode but the user has insufficient update rights.

These conditions are checked again when the paste command is handled.

Field values are not pasted if:

- the field is not part of the Details control
- the field is a Z-field
- the field in the Details control has the `ReadOnly` attribute set to true.
- the field is set to read-only in the data model (**Fields > Properties > "readonly"**)
- the user does not have sufficient conditional access rights for the field (in New mode: **Deny New**, in Update mode: **Deny Update** when updating a value or **Deny Deletion** when deleting a value).

## DefaultValuesDefine

Use this button to define the values for new records that are created.


The `DefaultValuesDefine` button <sup>[101]</sup> allows users to define default values for creating new records. It calls the `DefaultValues` action. For further details, see [DefaultValue](#) on page 415.

In UPDATE\_DEFAULT the `DefaultValuesDefine` button is configured into the MA `SearchResults` header.

The `DefaultValuesDefine` button is disabled by default when used in the header of a virtual info area (**Query State Action = Check\_VirtualInfoArea**).

## DuplicateRow

Use this to quickly duplicate a selected row.

The `DuplicateRow` button  can be configured into `SearchResults` headers and into the special header `SerialEntry.List` for quick add configurations: It allows users to duplicate the selected list row (i.e. create an identical copy of the selected record, including all link fields etc.). The default hotkey for this button is `Ctrl+Shift+V`.


Alternatively, you can configure the menu action `A_DuplicateRecord` into the desired context menu(s).

When duplicating a record in a list, the list refreshes and the duplicated record is automatically displayed in Edit mode with focus on the first editable field, provided that this new record appears within the list's currently displayed records (i.e. if you duplicate the last list row, the newly created record is not part of the current page so you do not see it in Edit mode). When duplicating a record in another view, the newly created record is displayed in its default view (e.g. Expand view).

This feature was especially developed for quick add pages, but can be used in any list if required. For further details, see [SerialEntry](#) on page 454.


## Edit

Use this button to make the form editable.

The `Edit` button  allows user to switch a form from View to Edit mode. The button calls the `Command` action with the `Command` argument set to `Edit`.

## EmailReply

Use this button to reply to email from within Aurea CRM.Web.

The `EmailReply` button  allows users to reply to email activities from within Aurea CRM.Web. This can be done by clicking the `EmailReply` button or pressing `Ctrl+R` a new email activity is created and linked to the original email activity. Users can write the email and then send it by using the `OpenAppointmentEmail` button

. For further details, see [OpenAppointmentEmail](#) on page 318 and [Send As email](#) on page 835.

In `UPDATE_DEFAULT` the `EmailReply` button is configured into the `MAEmail Expand` header which is used in the corresponding alternative Expand configuration for email activities. For further details, see [Alternative Expand Configurations](#). Thus the `EmailReply` button is available only for **Activity** records of type "email".

## Export/Report Buttons

Learn to create report in different formats using this button.

Aurea CRM.Web users can export data from Aurea CRM.Web to different file formats. Example - Microsoft Excel or PDF. For more information on XML exports and reports, see [XML Exports & Reporting](#) on page 856.



The following export/report formats are available (marked **Y**) in the specified context (list, expand, tree, query, analysis grid):

Format	List	Expand	Tree	Query	Analysis Grid
CSV	Y	Y	N	Y	Y
XML	Y	Y	N	Y	Y
HTML	Y	Y	Y	Y	Y
PDF	Y	Y	Y	Y	Y
Microsoft Excel	Y	N	N	Y	Y
Microsoft Word	Y	Y	N	Y	Y

**Note:** For Microsoft Excel/Word reports to work, Microsoft Office 2010 or higher is required.

**Note:** If an Expand view contains combined fields, they are exported as defined in the Edit mode configuration. For further details, see [Add Combined Field](#) on page 214.

**Note:** If a list contains an array field (e.g. **SIC 1** in FI), all fields from the array are exported. To export only those fields from the array that you actually use, set **Deny Access** for the undesired fields of the array. For further details, see [Access Rights](#) on page 829.

The export/report formats available in a certain context depends on the export/report buttons configured in the respective header.



All export/report buttons call the `Command` action template with `Command = Export`. Further settings (export options, XSL transformation, plug-in) are specified in the `CommandArguments` parameter. For further details, see [Export/Report Action Definition](#) on page 314. Use the `maxRows` parameter to determine the maximum number of rows actually fetched from the server. For further details, see [Export/Report Action Definition](#) on page 314.

Each export/report button, except `Direct Export All` and `Direct XslReport All Excel`, supports the possibility to export only the selected records from a list:

- If no record is selected, all currently visible records are exported.
- If one record is selected, all currently visible records are exported. If the `Export.SingleListRecord` Web Configuration parameter is checked, only this record is exported. For further details, see [Export.SingleListRecord](#).
- If more than one record is selected, the selected records are exported.

The `Direct Export All` and `Direct XslReport All Excel` buttons export all existing records.

The following buttons are available in `UPDATE_DEFAULT`:

Name		Parameters	Process
Direct Export		Command = Export	Export - XML or CSV
Direct Export All		Command = Export CommandArguments = {maxRows: -1}	Export (all records) - XML or CSV








**Note:** The `Direct Export` and `Direct Export All` buttons can be used to export records displayed in a list or expand view. The Web Configuration parameter `Export.Type` defines the export format (CSV (default) or XML). For further details, see [Export.Type](#).




**Note:** Both buttons are hidden by default.

**Note:** `Direct Export` exports data (number of selected or visible records in a list) by requesting it from the server. As this export is carried out server-sided, you can add a call-back function that "prepares" the exported data. This can be necessary if, for example, you want to export data from multiple queries in a form.

**Note:** `Direct Export All` exports data (all records of a list) by requesting it from the server. Use call-back functions like for the `Direct Export` button.



Name		Parameters	Process
Direct XslReport All Excel		Command = Export  CommandArguments = {maxRows: -1, report:"ExportExcel[EmitXmlNativeFormat=true]}"}	MS Excel Report (all records)
Direct XslReport Excel		Command = Export  CommandArguments = {report: "ExportExcel[EmitXmlNativeFormat=true]}"}	MS Excel Report
<p><b>Note:</b> <code>Export.Xml.EmitNativeFormat=true</code> ensures that data types are correctly mapped from Aurea CRM to Microsoft Excel, &gt;&gt; . Numeric fields (long, short, float) are mapped to "Number" in Excel, date fields to "DateTime" and logical fields to "Boolean". All other field types are passed as strings (also time fields, since Excel's XML format does not support time without date).</p>			
Direct XslReport HTML/Expand		Command = Export  CommandArguments = {report:"ExportExpandHtml[Target=_blank;ContentDisposition=inline]}"}	HTML Re- port/Ex- pand
Direct XslReport HTML/List		Command = Export  CommandArguments = {report:"ExportListHtml[Target=_blank;ContentDisposition=inline]}"}	HTML Re- port/List
Direct XslReport HTML/Tree		Command = Export  CommandArguments = {report:"ExportTreeHtml[Target=_blank;ContentDisposition=inline]}"}	HTML Re- port/Tree
Direct XslReport PDF/Expand		Command = Export  CommandArguments = {report: "ExportExpandPdf[Export.Xml.FatClientCompatible=true],Format,application/pdf}"}	PDF Re- port/Ex- pand
Direct XslReport PDF/List		Command = Export  CommandArguments = {report:"ExportListPdf,Format,application/pdf}"}	PDF Re- port/List

Name		Parameters	Process
Direct XslReport PDF/Tree		Command = Export  CommandArguments = {report:"Export-TreePdf,Format.application/pdf"}	PDF Re- port/Tree
<p><b>Note:</b> For information on Aurea CRM's PDF reporting plug-in, see the article "How to use the reporting PDF plug-in" at <a href="https://support.aurea.com">https://support.aurea.com</a>.</p>			
Direct XslReport WordML/Expand		Command = Export  CommandArguments = {report:"ExportExpand-Wordml"}	MS Word Report/Ex- pand
Direct XslReport WordML/List		Command = Export  CommandArguments = {report:"ExportList-Wordml"}	MS Word Re- port/List

In addition, you can create buttons (or menu actions) that execute a specific query and create a report from the results without displaying the result rows first. For further details, see [QueryReport](#) on page 436.

### **Export/Report Action Definition**

Learn to define action of Export/Report button.

Each export/report buttons calls the `Command` action with `Command = Export`. Further settings are specified by the `CommandArguments` parameter, e.g. `{maxRows: -1, report:"ExportExcel[EmitXmlNativeFormat=true]}`.

The `CommandArguments` parameter supports the following parameters:

#### **maxRows**

Determines the maximum number of records to be exported.

`maxRows` is mandatory when defining Export All buttons (`Direct Export All` and `Direct XslReport All Excel` in `UPDATE_DEFAULT`).

`maxRows` is applied as follows:

- if undefined, the Export All action (e.g. `Direct Export All`) is executed as e.g. `Direct Export`, i.e. only the selected records are exported.
- if defined (e.g. `{maxRows: 500}`), this value is used.
- if set to `{maxRows: -1}`, the value of the `Export.MaxRecords` Web Configuration parameter is used. For further details, see [Export.MaxRecords](#).

## report

Determines export options, the XSL transformation and plug-in defining the type, layout and content of the export or report.

### Syntax:

```
{report: "<XSLT>[<export option1>;<export option2>;etc.],Format.<mime type>[<plug-in parameter1>;<plug-in parameter2>;etc.]"}
```

For a list of available export options, see [Export Options](#) on page 858.

For a list of available parameters for the PDF plug-in, see the article “How to use the reporting PDF plug-in, additional PDF plug-in parameter” at <https://support.aurea.com>.

Export options can be passed with and without transformations. The options, transformations and plug-ins are separated by comma. Additional options for transformations and plug-ins can be specified in square brackets following the transformation/plug-in. Options inside square brackets must be separated with a semicolon.

The options specified with (only!) the first transformation are applied to the whole report as well as to the first transformation: Therefore writing

```
{report:"Export.Encoding=UTF-8,ExportExcel [Export.Xml.EmitNativeFormat=true]"}
```

is the same as writing

```
{report:"ExportExcel[Export.Encoding=UTF-8; Export.Xml.EmitNativeFormat=true]"}
```

### Examples:

without transformation:

```
{report:"Export.Xml.EmitConfiguration=true,Export.Xml.EmitCatalogs=true"}
```

with transformation:

```
{report:"ExportExpandHtml[Target=_blank; ContentDisposition=inline]"}
```

with plug-in:


```
{report:"ExportListPdf,Format.application/pdf[FontFile=%WINDIR%\\Fonts\\arial.ttf]"}
```

with transformation and plug-in:

```
{report:
"ExportListPdf[Export.Xml.EmitConfiguration=true;Export.Xml.EmitConditions=true],
Format.application/pdf[EncryptionPassword=update;EncryptionPermissions=8;
EncryptionStrength=true]"}
```

## Legend

Learn about the Legend button and its configuration.

The Legend button  is configured in the following special headers: CalendarHeader, DayWeekPlanningCalendar, DispatchingDashboard, MarketingCalendarHeader, ResourcePlanHeader and VisitPlanning.

It displays the meaning of the color coding of the items displayed in the respective view, e.g. activities by type in the calendar or items by info area in the marketing calendar etc.

The colors are defined in the `..\styles \<theme>\crm\colors.css` file. For further details, see [Customizing Styles](#) on page 15.

For information on changing the color coding of items, see the article “How to Customize the Color Coding of Calendar Items” at <https://support.aurea.com>.

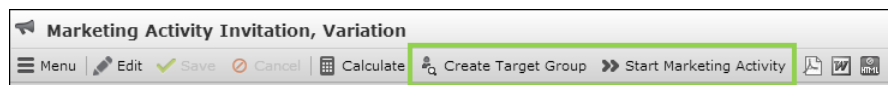
In the marketing calendar the colors defined for the calendar are used as follows:

Item Status	Color from calendar
Scheduled	u8-color-calendar-item-scheduled
In Progress	u8-color-calendar-item-superior
Completed	u8-color-calendar-item-done
Canceled	u8-color-calendar-item-missed
Archive	u8-color-calendar-item-nostate
Template	u8-color-calendar-item-info
Budget Approval	u8-color-calendar-item-fax

## Marketing Activity Buttons

Learn to create target groups and marketing activities.

The `CreateTargetGroup` and `CreateMarketingActivity` buttons configured in the AK and AD `Expand` headers in `UPDATE_DEFAULT` allow to manually start the creation of target groups and marketing activities.



These buttons are active if the following conditions are met:

- The Web Configuration parameter `UpdateServer.RepID` is defined. For further details, see [UpdateServer.RepID](#).
- An **Execution (Marketing Activity)** record (AD) exists for this marketing activity.

For details on the business logic, see **Creating a Target Group** and **Executing a Marketing Activity** in the *Web Business Logic User Guide* ([https://help.aurea.com/crm/crm-web-bl-btb/crm-web-bl-otc/dita/topics/c\\_web\\_bl\\_btb-1\\_1-Aurea-CRM-web\\_crm.html](https://help.aurea.com/crm/crm-web-bl-btb/crm-web-bl-otc/dita/topics/c_web_bl_btb-1_1-Aurea-CRM-web_crm.html)).

**Note:** To provide timely feedback on the generated records, set the polling interval for messages to a few minutes. For further details, see [Jobs.PollingInterval](#).

## NewCommand

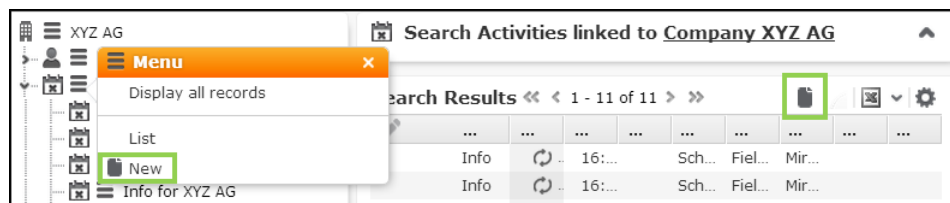
Use this to create a new record from tree view.

The `NewCommand` button allows users to create new records from inside a Tree view without leaving the Tree view. For further details, see [Tree](#) on page 381.

The `NewCommand` button is configured in the Base vertical and is therefore available in all verticals. In `UPDATE_DEFAULT` it is used in various `SearchResults` and `SearchResultsLinked` headers (Example - in header group `MATree`). For further details, see [Headers for TreeViews](#).

**Note:** Creating new **Document Link** records (D3) inside the tree is not supported.

A corresponding menu action `A_TreeNew` is configured in `M_TreeChildren` (the default context menu used for the info area nodes of a tree). For further details, see [Info Area Settings](#).



Clicking on `NewCommand` calls the command "New" that executes the `ExpandNew` action for the current info area.


You can use the `NewCommand` button in all headers of record lists (i.e. not only in Tree views): `SearchResults`, `SearchResultsLinked`, `SearchSubList`, `SubList`.

When used in a Tree view, the "New" command is executed as follows:

- If the info area is a tree node, the `ExpandNew` page is displayed inside the Tree view.
- If the info area is not a tree node, the `ExpandNew` action is executed in a new tab.

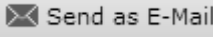
## NewSiblingRecord

Use this button to create sibling record.

The `NewSiblingRecord` button  allows users to create a new record starting from the Expand or List view of a record of the same info area. Example - create a new MA record while viewing an existing MA record. The new record is created with the same parent link(s) (Example - FI and PE) as the current record. Additional field values (Example - additional links) can be passed to the new record via "copy fields". For further details, see [NewSiblingRecord](#) on page 317.


## OpenAppointmentEmail

use this button to e-mail activity records.

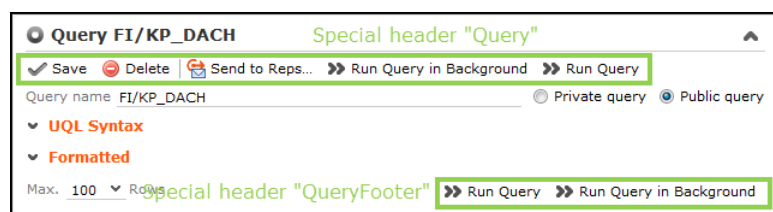
The `OpenAppointmentEmail` button , which is configured into the `MAExpand` header in `UPDATE_DEFAULT`, allows users to send activity records via email. For further details, see [Send As email](#) on page 835.


## Query Background Processing Buttons

Use this button for background processing of queries.

The `Query_RunInBackground` button  configured in the special headers `Query` and `QueryFooter` in `UPDATE_DEFAULT` allows the background processing of queries by CRM.server. For further details, see [Background Queries](#) on page 502.

Once the query results are retrieved and stored as a query snapshot, the user is notified with a message that his query results are available.



Analogous, the `Query_RunInBackground_CSV` button  allows the background creation of a CSV file containing the query results. CRM.server creates a D1 document that is linked (via a D3-link) to the message the user receives upon completion.

---


**Note:** If this button is used, no query snapshot is created but only the CSV file.

---

To allow users to directly create a CSV file with the query results, you need to configure the `Query_RunInBackground_CSV` button into the `Query` and/or `QueryFooter` header(s).


## Refresh

This button is used to refresh.

The `Refresh` button  can be used e.g. to refresh the items list on quick add pages. For further details, see [SerialEntry](#) on page 454. Add the `Refresh` button to the respective `SerialEntry.List` header (e.g. for `PR`) to allow recalculating e.g. discounts based on changed quantities.

## ResetFormLayout


Use this button to reset the layout to parent configuration.

The `ResetFormLayout` button  (**Reset Layout**) allows users to revert Details controls they rearranged using drag&drop to the original layout as defined in the current (parent) configuration. For further details, see [Details Controls Extended by Users](#).

Clicking **Reset Layout** removes the control's extension in CRM.Designer.


## Save

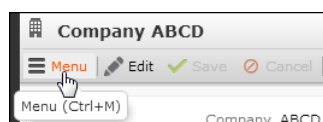
You can save changes to a record using this button.

The `Save` button  allows users to save changes of a record. The button calls the `Command` action with the `Command` argument set to `Save`.

## ShowMenu

Learn the use of ShowMenu button.

The `ShowMenu` button  allows users to open an info area's context menu in Expand views.



## SumTotal

The `SumTotal` button  calculates the sum for numeric fields in a list.

You can configure the `SumTotal` button into any `SearchResults` header. For further details, see [SearchResults/SearchResultsLinked/SearchResultsSelect](#). The `QueryResult` header contains this button by default.

The `SumTotal` button is only active if all search or query result records are displayed on the current page because the calculation is performed on the client and would otherwise lead to falsified results.

## ViewAllFields

You can view and edit all fields of a current record using `ViewAllFields`.

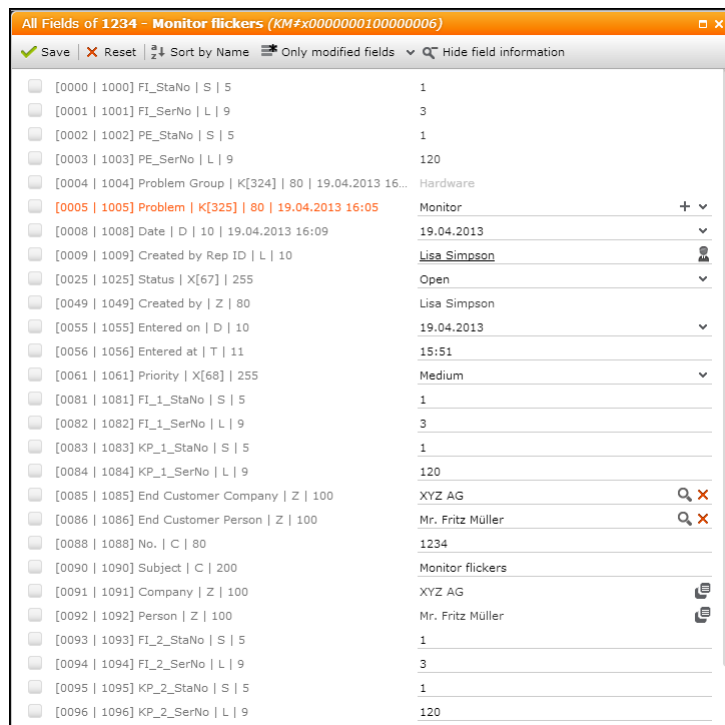
The `ViewAllFields` button  allows users to view and edit all fields of the current record independent of the current field control definition. The `ViewAllFields` button calls the `u8.Crm.Widgets.ExpandAllFieldsPopup.open` function.

You can configure the `ViewAllFields` button for Expand, Tree and List views. Alternatively, you can configure the `A_ViewAllFields` menu action into an info area's context menu.

By default, the `ViewAllFields` button is visible for administrative users only, i.e. users with access to the CRM process 1.5.1.3.108 "All Fields" (CRM process ID B\_500000840).

The `ViewAllFields` button is only active if exactly one record is selected.

**Query State Action** = `C_OneRowSelectedOrNoList`.



In the **All Fields** dialog users can

- enter new values or change existing ones according to his access rights as defined in the Aurea CRM win **Rights** module (see below).
- sort the fields by name or number
- toggle between all fields, fields containing a value and modified fields
- display the following additional information:
  - field number and "base" field number
  - field label
  - field type. For further details, see [Data Types in Aurea CRM](#) on page 877
  - catalog number for catalog fields
  - field length (e.g. number of allowed characters for text fields)
  - date of last modification

Changing a field value automatically checks the check box. Unchecking the check box resets the field value. Click **Reset (Reset)** to undo all changes. For hierarchical catalogs, if both parent and child catalog are changed, resetting the change of either parent or child value resets the other value automatically.

Mandatory fields (**Deny Deletion** in the rights definition) are highlighted.

**Note:** The `ReadOnly` and `Must-Field` field attributes (as defined in CRM.Designer) have no effect on the **All Fields** dialog.



The following fields are displayed read-only:

- links to parent records. (Links to additional records can be edited via the record selector `↻`.)
- other decoded fields (e.g. rep fields)
- fields defined as read-only in the Aurea CRM win **Rights** module (**Deny Update**).
- the system fields **Created on** (4010), **Updated on** (4011), **Added on StNo.** (4012), **DelCd.** (4013) and **CmCd.** (4014) are displayed read-only for all users except the SU. For more information, see **System Fields** in the *CRM.Core Administrator Guide* ([https://help.aurea.com/crm/crm-core-admin-guide/crm-core-admin-guide/dita/topics/c\\_core\\_admin-6\\_7-System-Fields\\_crm.html](https://help.aurea.com/crm/crm-core-admin-guide/crm-core-admin-guide/dita/topics/c_core_admin-6_7-System-Fields_crm.html)).
- system fields with ID  $\geq$  4015 (**Created at**, **Updated at** etc.) are not editable even for the SU.

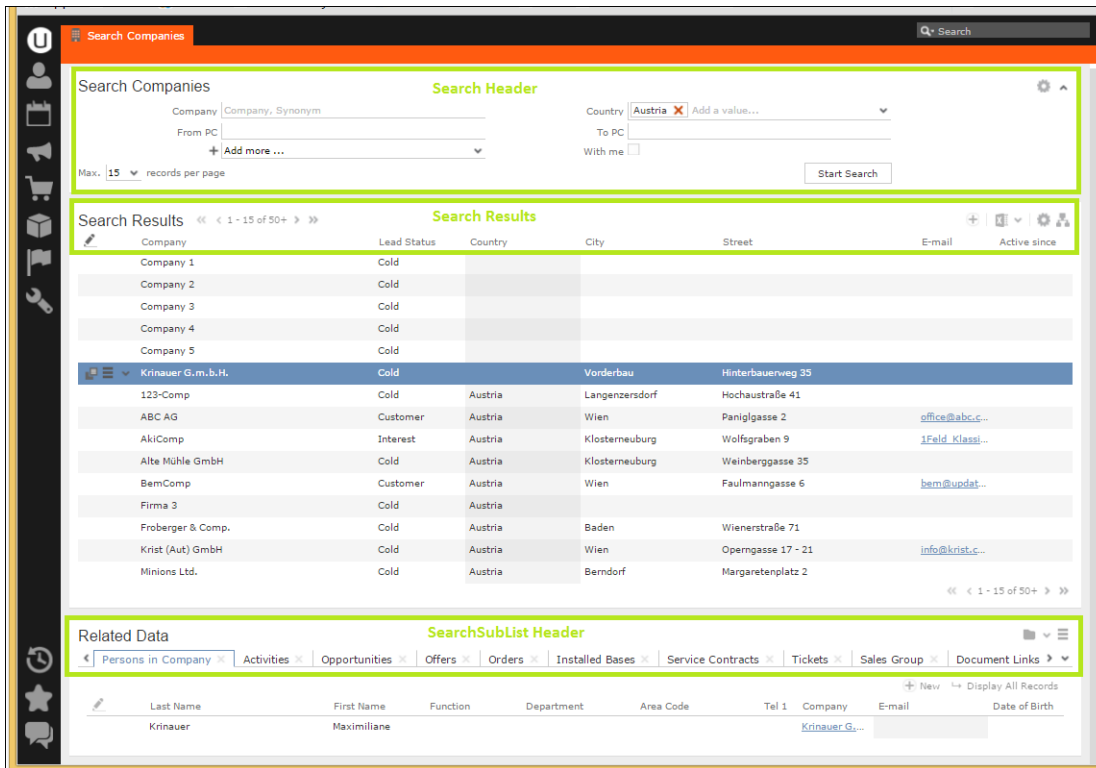
The following fields are hidden:

- fields with **Deny Access** defined in the Aurea CRM win **Rights** module
- custom fields (defined in `CustomFields.xml`)
- fields flagged as `VirtualCore` are hidden if the `Query.ShowVirtualFields` Web Configuration parameter is set to `false`. For further details, see [Query.ShowVirtualFields](#).

## Headers

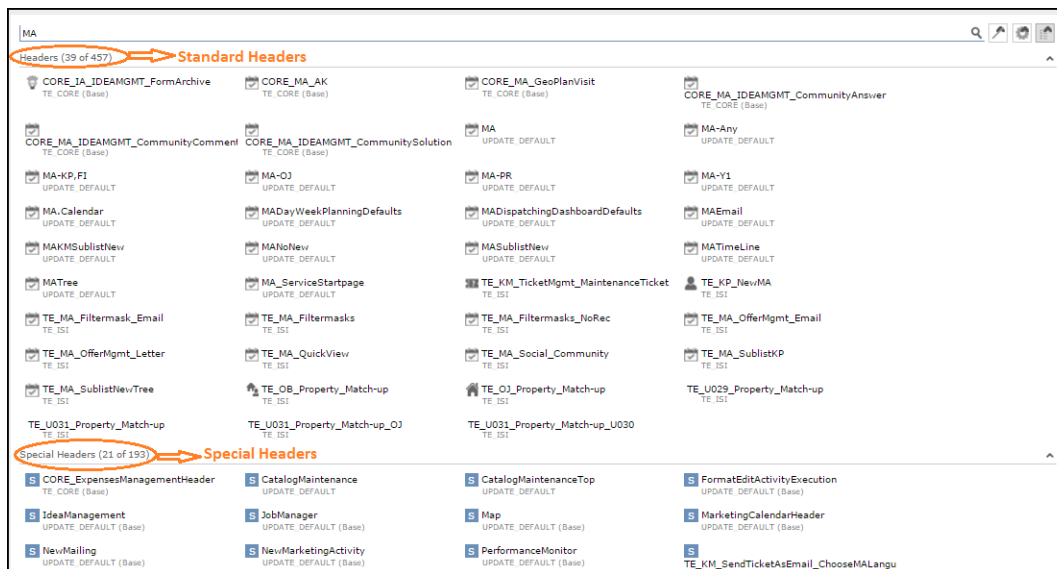
You can know about headers in Aurea CRM and how to configure them in designer.

Each page in Aurea CRM.Web contains one or more headers, e.g. the company search contains a `Search`, `SearchResults` and `SearchSubList` header (**Related Data**).



There are two types of headers:

- **Standard headers** providing the standard header functionality of pages in Aurea CRM.Web, e.g. Search, Expand, ExpandChildren etc., see [Standard Headers](#) on page 337. Standard headers are always info area-specific.
- **Special headers** providing some special functionality, e.g. the various headers for the **Calendar** page or the **Participants** header for MA, see [Special Headers](#) on page 342. Special headers can be info area-specific or info area-independent.



Info area-specific headers are "collected" in header groups. For further details, see [Header Groups](#) on page 323. If a header is not defined in a header group, the corresponding header from the DEFAULT header group is used. For further details, see [DEFAULT Header Group](#).

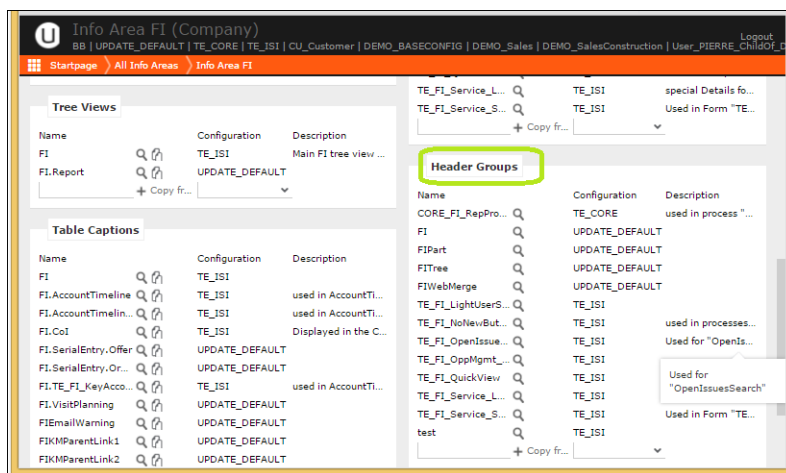
Headers are assigned to pages in their respective configurations (see [Expand Configuration](#) on page 352) or via other headers. For details, see [Related Data area \(Header Info Areas\)](#) on page 329.

## Header Groups

The configuration of info area-specific headers is organized in Header Groups.

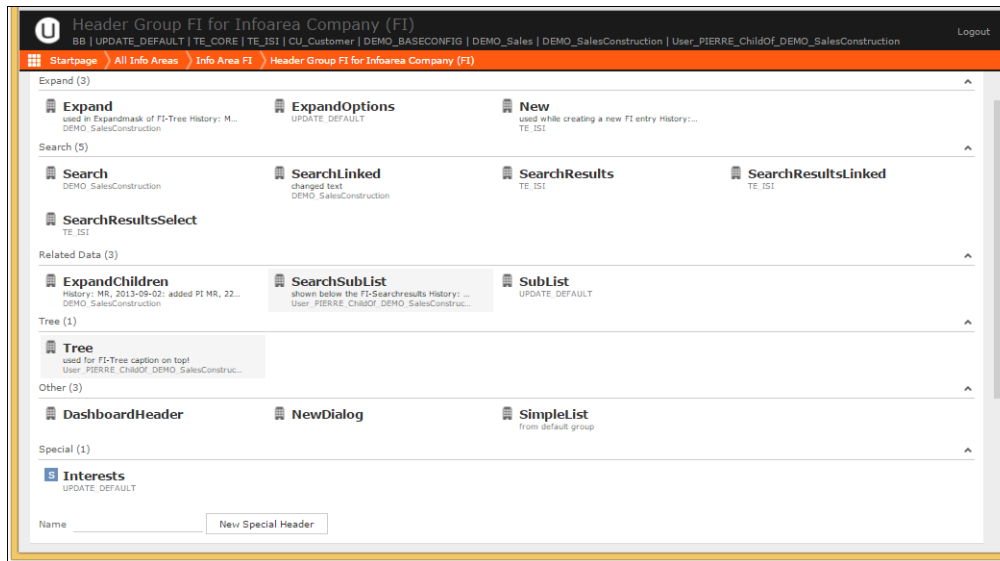
A header group contains one or more **headers**, e.g. the `Search`, `Expand`, `SearchResults` headers for the selected info area. For further details, see [Designing Configuration Units](#) on page 64.

The following screenshot shows the header groups defined in the infoarea F1.



If a header is not defined in a header group, the corresponding header from the DEFAULT header group is used. For further details, see [DEFAULT Header Group](#).

The following screenshot shows the different headers in the F1 header group:



## DEFAULT Header Group

The DEFAULT header group contains minimum/standard header definitions valid for each info area. These headers are used as a fallback for info areas for which no header (group) is defined.

Examples:

The `Expand` header of the DEFAULT header group contains Edit/Save/Cancel and the various buttons for creating reports - this header is used by all info areas.

The `Search` header of the DEFAULT header group contains the following **Formattext**: `{INFOAREAIMAGE} Search {INFOS}` - this is also applicable for all info areas.

## Naming Conventions

The default header group used for an info area is called `<InfoAreaId>`, e.g. `FI` is the default header group for the **Company** info area.


To define headers for specific views or functions (e.g. headers to be used in the Tree view or special headers to be used for displaying calendar items), use the following naming conventions:

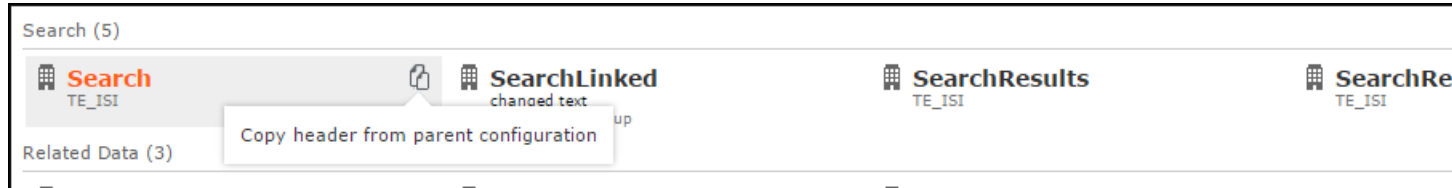
- `<InfoAreaId>.Calendar`: The `Expand` header configures the buttons for the calendar popup view for this info area, e.g. `MA.Calendar`.
- `<InfoAreaId>.SerialEntry`: The page header for quick add pages, see [SerialEntry](#) on page 454.
- `<InfoAreaId>.SerialEntry.List`: The header of the item list on quick add pages.
- `<InfoAreaId>.SerialEntry.SubList/SerialEntry.RelatedHeader`: The headers configuring the Related Area of a quick add page.
- `Y1.BuyingCenter`: Used by the Buying Center Wall, see [BuyingCenter](#) on page 399. The `Expand` header configures the page's header, the `ExpandChildren` header configures the page's related data area.

## Configuring a Header

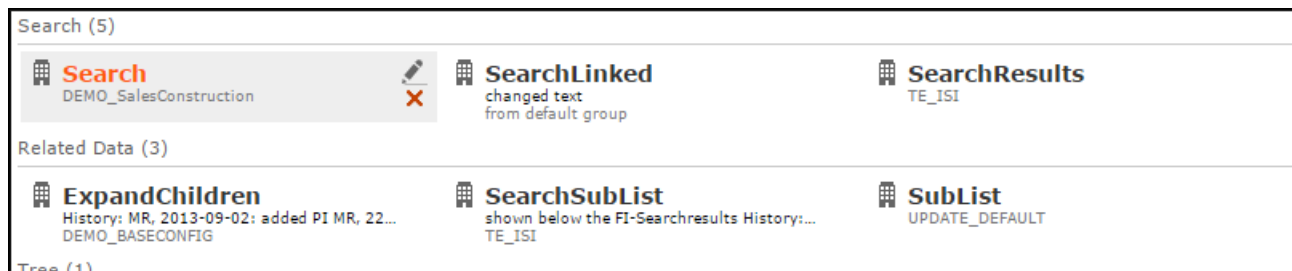
Learn to configure header in this topic.


To configure a header:

1. Click **Headers** on the CRM.Designer main page (**Navigation** area) and select an info area.
2. Point at the header you want to configure, the **Copy Header From Parent Configuration** () button appears.



3. Click (). The edit and delete buttons appear.



4. Click Edit (). The configuration page displays.
5. Depending on the header, you can configure the following:
  - general attributes (Text, Image etc.), see [Header Attributes](#) on page 326
  - buttons calling specific functions, see [Header Buttons](#) on page 327
  - Related Infoarea for Search and Expand views (`SearchSubList`, `ExpandChildren`), see [Related Data area \(Header Info Areas\)](#) on page 329
  - the links area below an Expand view (`ExpandOptions`), see [Header Links](#) on page 333
  - extended page headers (e.g. for the Tree view) displaying more record details, see [Page Headers](#) on page 334

For examples of all header types and styles, see **Headers** on the **SDK** page.

## Header Attributes

Learn how to configure header attribute.

For each header you can define:

- **Text:** The text displayed in the header, e.g. `{INFOAREAIMAGE} Search {INFOS}`. Enclose placeholders in `{}`.

### Search Companies

Use the following placeholders:

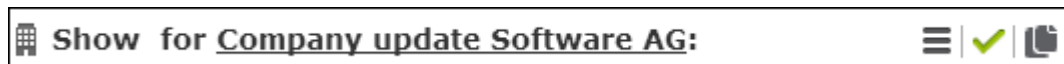
- `{RECORD}`: The current record's table caption, see [Table Captions](#) on page 265. Use `{RECORD:<Table Caption Name>}` to specify another than the default table caption.
- `{RECORDLINK}`: Like `{RECORD}`, but as a link to that record. Clicking the link executes the info area's default action.
- `{RECORD[param]}`: The table caption of the record whose uid is passed as `param`.
- `{RECORDLINK[param]}`: A link to the record whose uid is passed as `param`.
- `{NORECORD:text}`: A text (or text ID) to be displayed in the header. Use `{NORECORD:text}` together with `{RECORD}` in process headers.
- `{IMAGE:imagenam}`: An image with the specified name. The image must be registered in CRM.Designer, see [Images](#) on page 717
- `{INFOAREAIMAGE}`: Displays the info area's default image. The image must be registered in CRM.Designer as `INFOAREA_<InfoAreaId>`.  
Use `{INFOAREAIMAGE[:size]}` to specify the size: "n" for 24x24 px or "s" for 16x16 px.
- `{INFO}`: The name of the current info area.
- `{INFOS}`: The plural name of the current info area as defined on the [Info Area Labels](#) page, see [Info Area Labels](#) on page 275.
- `{<InfoAreaId>}`: The name of an info area.
- `{<InfoAreaId>s}`: The plural name of an info area as defined on the [Info Area Labels](#) page.
- `{SUBINFOS}`: A drop-down list with configured sub-info areas, see [Related Data area \(Header Info Areas\)](#) on page 329.
- `{LINKS}`: The configured links displaying related data, see [ExpandOptions](#).
- **Image:** The name of the image displayed left of the header text. The image must be registered in CRM.Designer, see [Images](#) on page 717.

---

**Note:** Headers support only one image: The image defined in **Image** overrides the image defined in **Text**. If no **Image** is defined and the **Text** contains the table

caption of a record (`{RECORD}`): If the table caption contains an image definition, this image is displayed.

- **Show buttons with text:** If checked, buttons are displayed with their defined label text, otherwise only the button image is displayed, see [Header Buttons](#) on page 327.
- **Show buttons in extra line:** If checked, buttons are displayed in a separate line, i.e. below the header text.
- **Use plain style (no borders):** The header is displayed without the bottom line.



- **Use page header style:** Uses a style without background color.



This style is automatically applied, if a header actually is the header of a page, e.g. a `Search` header.

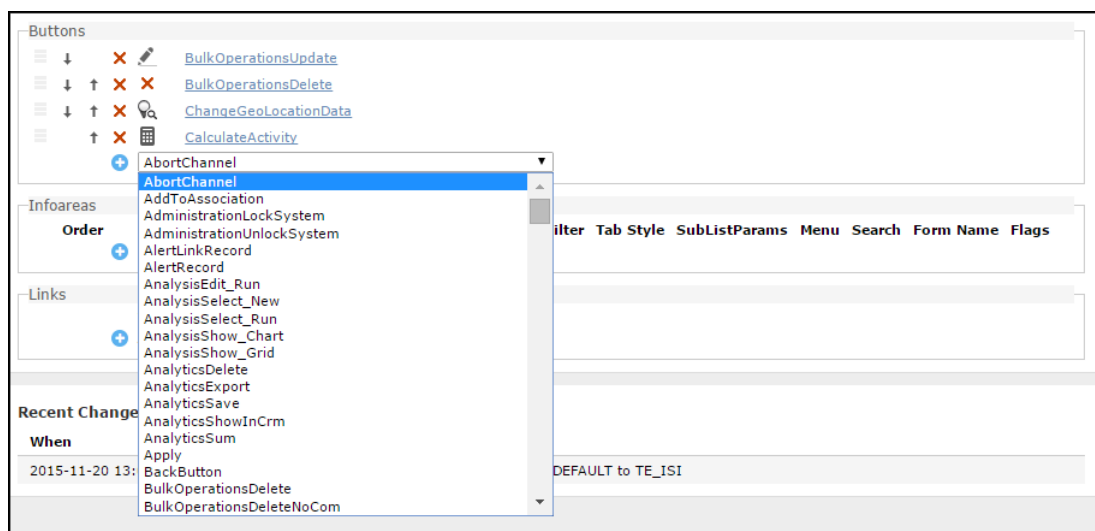
- **Fixed tab width (SublistTabWidth):** currently no function

## Header Buttons

Learn to configure buttons into headers.

To configure buttons into headers (Example: `MA SearchResults` header):

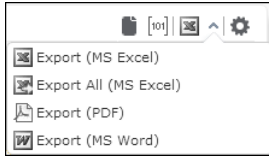
1. Select a button from the drop-down list.



2. Click + (**Add**).
3. Click on a button's name to change its settings, see [Configuring a Button](#) on page 305. To create a new button, you have to switch to the **Buttons** page of CRM.Designer.

Use the  $\downarrow/\uparrow$  buttons to rearrange the order of the buttons.

Use separator and group elements to structure the buttons displayed in the header, see [Grouping Buttons](#) on page 328.



## Grouping Buttons

Learn to group buttons to avoid clutter on the user interface.

You can group several buttons configured in a header to achieve a less cluttered user interface without removing functionality. Example: The various report buttons are grouped in the default SearchResults header in UPDATE\_DEFAULT.

---

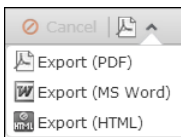
**Note:** Buttons with the attribute **Owner Draw Buttons** cannot be part of a button group.

---

## Group Start/End

You can create a group of buttons using the invisible `Group Start` and `Group End` buttons: All buttons in between are part of the group. The first button is displayed by default.

Use `Group Start (with Text)` to display the button text of the first button or `Group Start (without Text)` to hide it. The texts of the buttons in the drop-down list are always displayed.



Users can select the other buttons from a drop-down list. The user's choice is stored, making the last used button visible by default.

To disable this behavior, create the Web Configuration parameter `Header.KeepGroupTopElement` (type: Character). For further details, see [Header.KeepGroupTopElement](#). Set it to "true" or "1" to disable the storing of the last selection: The first button of the group is then always displayed on top.

## Group Start (Menu)

Alternatively you can use the `Group Start (Menu)` button to create a button group. This displays a context menu button  $\equiv$ . All buttons configured below the `Group Start (Menu)` button are part of that context menu.





## Related Data area (Header Info Areas)

Learn about the Related Data area.

The **Infoareas** area of the `ExpandChildren` and `SearchSubList` header definition defines the **Related Data** area. It determines which sub-lists containing records from linked info areas or other related information displayed via forms are displayed at the bottom of Expand and Search pages.

The Related Data area is displayed using the `DashboardPanel` widget (with default layout = Tab) with each sub-info area configured in the **Infoareas** area as a `DashboardPanelItem`. For further details, see [Dashboards](#) on page 632. The settings for these items are defined directly in the `ExpandChildren` and `SearchSubList` header configurations.

---

**Note:** You can only use info areas with a data model link to the info area to which the header definition applies.

---



---

**Note:** You cannot define sub-info areas for info area-independent headers.


---

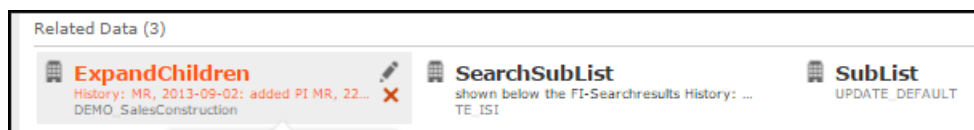
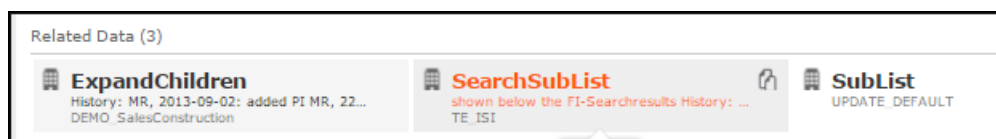
To define the Related Data area:



1. Click **Info Areas** on the CRM.Designer main page (**Views** area) and select an info area.

Or

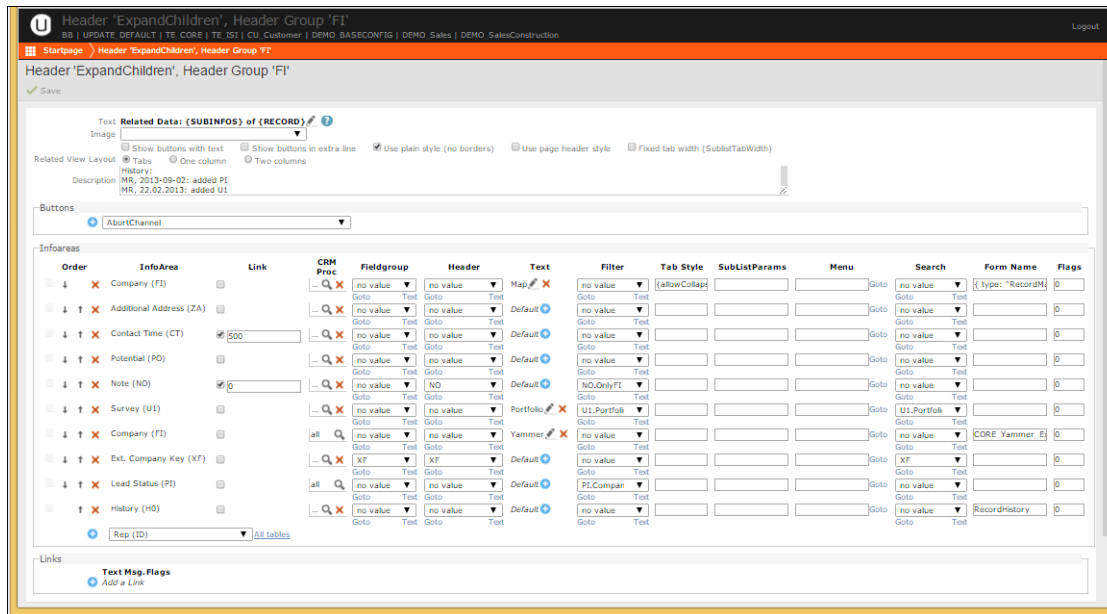
Click **Headers** (either on the CRM.Designer main page (**Navigation** area) and select an info area.

2. Point at the header you want to configure, the **Copy Header From Parent Configuration** () button appears.



3. Click . The edit and delete buttons appear.
4. Click Edit . The configuration page displays.

The following screenshot shows the **Expand Children** related data area configuration page.



5. **Related View Layout:** Determine the initial layout of the Related Data area: Tabs, One columns, Two columns. This setting is used as initial default and is overruled by the user's settings.

**Note:** The display settings for the Related Data area are stored in the browser cache. Therefore to reset the initial layout, make sure users clear their browser cache.

6. Add/edit the desired sub-info areas:
- Select an info area from the drop-down list and click **+** (**Add**).
  - Use the **+/+** buttons to rearrange the order of sub-info areas.
7. For each sub-info area you can define the following:
- **Link:** Check **Link** to select another than the default link, see [Link IDs](#) on page 19. For the **Document Link** info area (D3) you need to specify the generic link (ID 127).
  - **CRM Proc:** Assign one or more CRM processes to the sub-info area. For further details, see [Roles & CRM Processes](#) on page 727.
  - **Fieldgroup:** Defines the layout of the child records. The `List` control of the specified field group is used. For further details, see [Field Groups](#) on page 188. If undefined, the default field group `<infoAreaId>` is used.
  - **Header:** Defines the header used for the sub-info area. The `SubList` header of the specified header group is used. For further details, see [SubList](#). If undefined, the `SubList` header of the header group `<infoAreaId>` of the sub-info area is used.

- **Text:** Defines a label for the sub-info area. If undefined, the info area name (plural) is used. Use this for filtered sub-lists or for a special relation like displaying "sub-projects" of a project.
- **Filter:** You can specify a filter to filter the child records.

Alternatively, you can enter a link name, see [Link Names in JavaScript](#) on page 22. Example: If you specify `$Link[FI]` for the MA sub-info area of the PR `ExpandChildren` header, the **Activity** sub-list contains all activities linked to the company and not just those linked to the current offer record.

---

**Note:** You can either specify a filter **or** a link name. A combination of both is not possible.

---

- **Tab Style:** Defines display options for the sub-info areas (i.e. `DashboardPanelItems`) using the JSON syntax, e.g. `{allowRemove: false, allowCollapse: true}`.

The following options are available:

- `allowCollapse`: If set to `true`, the user can collapse the item (default: `true`).
- `allowRemove`: If set to `true`, the user can remove the item (default: `true`).
- `collapsed`: Determines the initial state (collapsed | expanded) of the item (default: `false`).
- `visible`: Determines whether the item is initially visible (default: `true`).

**Tab Style** also allows you to specify a CSS class to color-code a tab. Multiple CSS classes can be separated by a space.

To ensure that the assigned style is always visible (and not just when the tab is active), add `!important` to the style elements, for example:

```
.tab_MyRedText
{
  font-weight: bold !important;
  color: #b40202 !important;
}
```

---

**Note:** The **TabStyle** field is limited to 50 characters.

---

- **SubListParams:** Defines the appearance and behavior of the sub-list. For further details, see [ListParams](#) on page 374.

---

**Note:** `SubListParams` defined in the header are overridden by the corresponding `subListParams` defined in the `Search` action. For further details, see [Search](#).

---

- **Menu:** Defines the context menu displayed for the records in the sub-list. For further details, see [Context Menus](#) on page 292. If undefined, the context menu specified in the respective Search&List configuration is used.
- **Search:** The name of the Search&List configuration used for the sub-info area. For further details, see [Search&List Configuration](#) on page 365.
- **Form Name:** A form or widget to be displayed. You can enter one of the following:
  - a form's name, e.g. `RecordHistory`. For further details, see [Displaying a Record's History](#) on page 824
  - a JavaScript object specifying a form with at least a property `formName`, e.g. `{formName: "MyForm", <more properties>}`
  - a JavaScript object specifying a widget, e.g. `{type: "RecordMap", options: {size: {height: 400}}}`

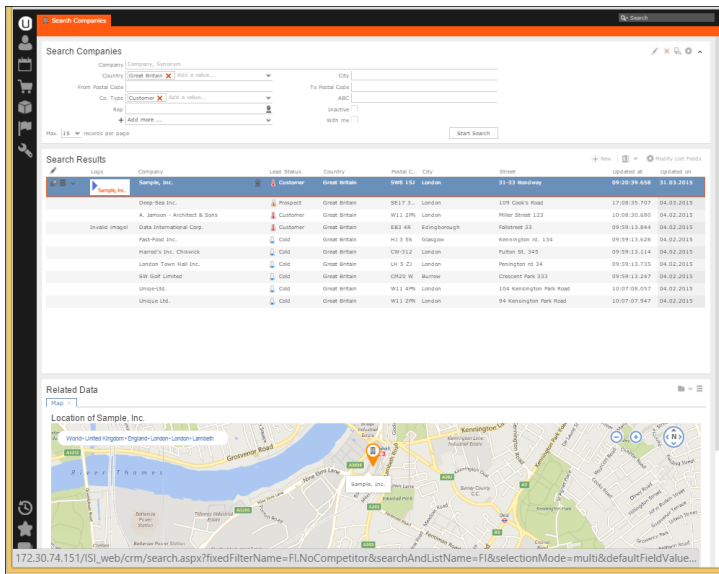
In addition you can specify under which conditions the Related Data area displays records using the `infoAreaID` property, e.g. `{formName: "SerialEntry.Offer.RelatedForm", infoAreaId: "+"}` - this displays the Related Data area of the generic Offer quick add page (BTB) regardless of which record is selected in the list.

For more information, syntax and examples, see the article “How to define related view” at <https://support.aurea.com>.

- **Flags:** Defines additional settings. The following options are available:
  - `ENABLECONFIG (1)`: Allows the user to configure the sub-list.
  - `COMPANYRELATED (2)`: Only for sub-lists of a linked company. If this flag is set, only records that are directly linked to the company are displayed, records linked to a person in that company are not displayed. (Use this e.g. for a MA sub-list to only display company-related activities).
  - `CHILDRELATION (4)`: Lists all children of an info area that has a hierarchy defined. Example in `UPDATE_DEFAULT`: The `ExpandChildren` header for A1 contains a tab listing sub-tasks.

Flags can be combined by adding them. 5 would mean that flags 1 and 4 are set.

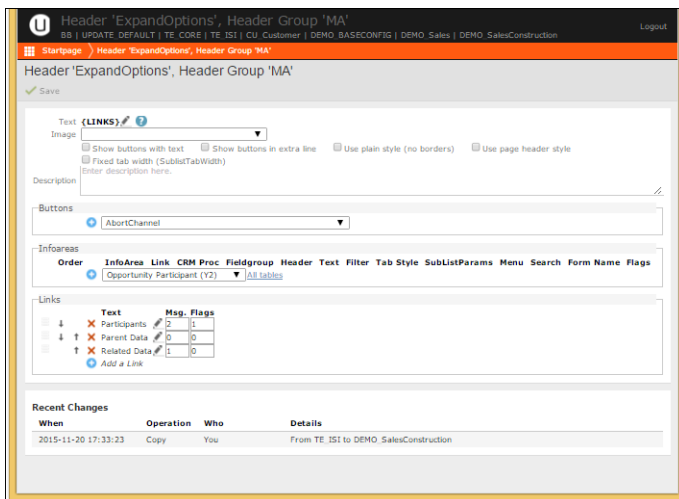
The following screenshot shows the resulting headers in the **Search Companies** page. In the **Related Data** area it is displaying the google map showing the location of the selected company.



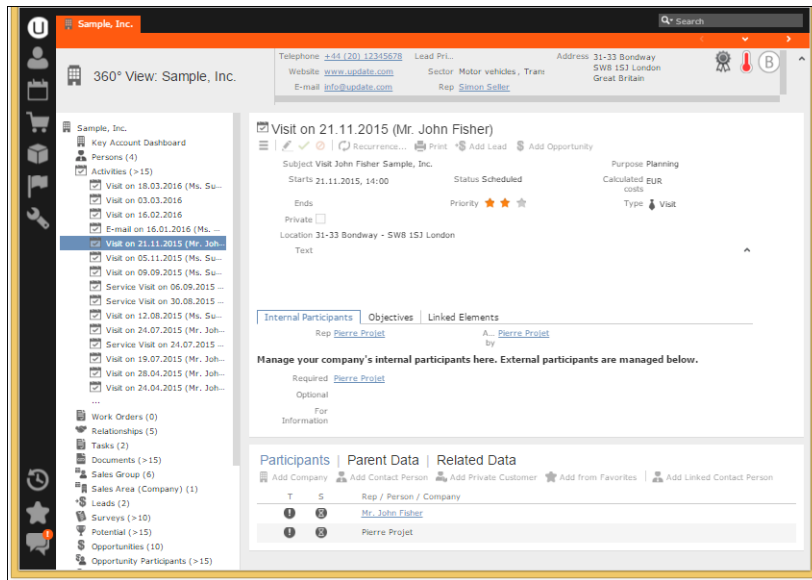
## Header Links

Header links are used in the ExpandOptions header to define what is displayed below the details mask of an Expand page.

The following screenshot shows the MA ExpandOptions header configuration page. The header's Links area shows the three links added for **Participants**, **Parent Data**, and **Related Data**.



The following screenshot shows the result of the configuration where the three linked areas **Participants**, **Parent Data**, and **Related Data** are displayed in the activity window of a user.



Define {LINKS} as the header's **Text**. The {LINKS} placeholder is replaced by the options defined under **Links**.

For each link option, you need to define the following:

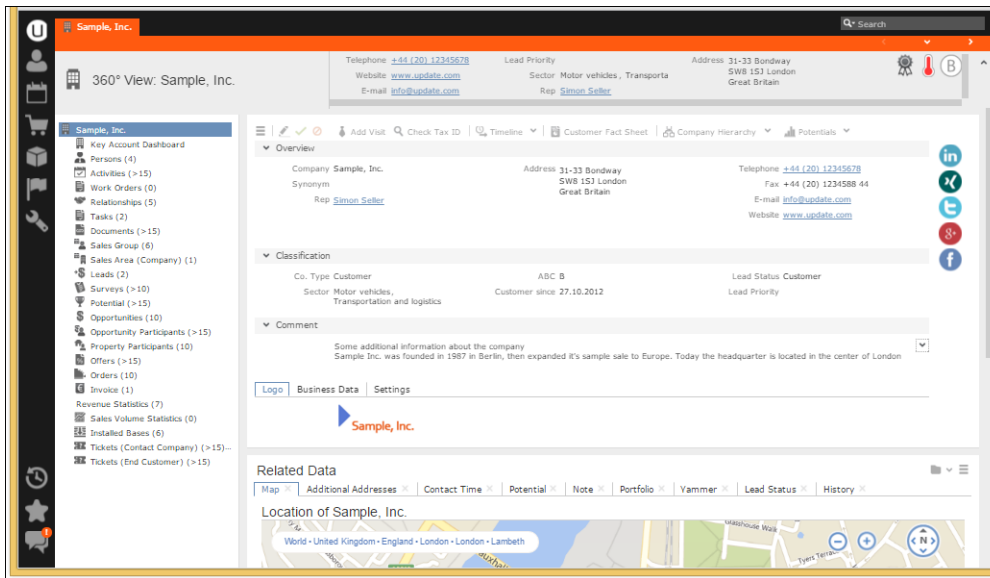
- **Text:** The text to be displayed as a link.
- **Msg.:** If a user clicks a link, the header calls the page's message call-back function to handle the click. The following messages are available on the Expand page:
  - **Message "0":** Displays Processes & Links, i.e. the parent records of the current record and their context menus.
  - **Message "1":** Displays Related Data using the `ExpandChildren` header with the related sub-info areas. For further details, see [Related Data area \(Header Info Areas\)](#) on page 329.
  - **Message "2":** Displays Participants of an activity if selected.
- **Flags:** If all flags are set to zero, the link with Message "0" (Processes & Links) assigned is opened by default when the user loads the page. If you want another link option to be opened by default, set the flag of this option to "1". In the example above, Participants is the option which is displayed when the page is opened.

## Page Headers

Use page headers to display important information about a parent record, in order to provide context information.

A page header consists of a header text, read-only record information (MiniDetails view) and optional buttons displayed in an extra row or as a context menu.

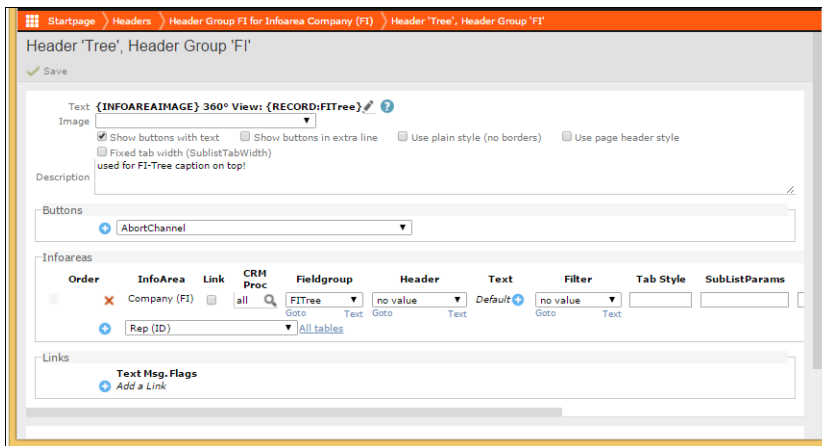
The following screenshot shows the FI `Tree` page header of a Company view.



You can define page headers for the following views:

- Tree view (Tree header)
- Record-related searches (SearchLinked header)
- Record-related analyses and queries (special headers)

The following screenshot shows the Tree header configuration page of the FI(Company) header group.



To configure a page header:

1. **Text:** If undefined, the default table caption is used. You can either reference a specific table caption (Example in UPDATE\_DEFAULT: {INFOAREAIMAGE} 360° View: {RECORD:FITree} for the Tree header for FI) or enter a header **Text**. For further details, see [Header Attributes](#) on page 326 and [Table Captions](#) on page 265.

**Configuration for 'FITree'**

Image

Prefix-Text: Company

Format-Text: {1}

{5}, {2}

{3}, {4}

Configuration: UPDATE\_DEFAULT

Description:

**Fields**

Field Name	Info area
1. Company (2)	Company (FI)
2. Country (5)	Company (FI)
3. Sector (29)	Company (FI)
4. Co. Type (28)	Company (FI)
5. City (8)	Company (FI)

If taken from the table caption the header text supports line breaks and different formatting for the 1st and subsequent lines. For further details, see [Table Captions](#) on page 265.

**Text** overrules the table caption.

If you specify an **Image**, this overrides the image defined in the **Text**.

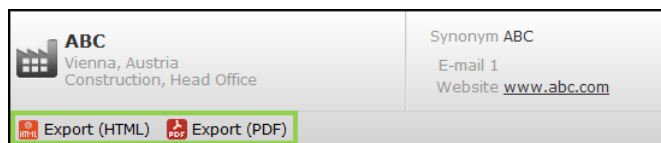
2. **Infoareas:** To configure the read-only details for the page header, specify an info area with a **Fieldgroup**: The MiniDetails control definition of that field group is used (FITree in the above example).

Images, fields with/without labels and icons are supported. For further details, see [Field Attributes](#) on page 221. Optional graphical elements defined by the `Large Image Template` field attribute are displayed side by side instead of below each other. For further details, see [Large Image Template](#) on page 231.

**Note: Restriction:** The MiniDetails control used in the page header may not have more than 4 rows.



3. **Buttons:** If **Show buttons in extra line** is disabled, they are grouped in a context menu in the bottom right corner of the header. If **Show buttons in extra line** is enabled, they are displayed as buttons in a second row.



**Note:** If neither a table caption with line breaks nor an info area is defined, a "normal" header is displayed.

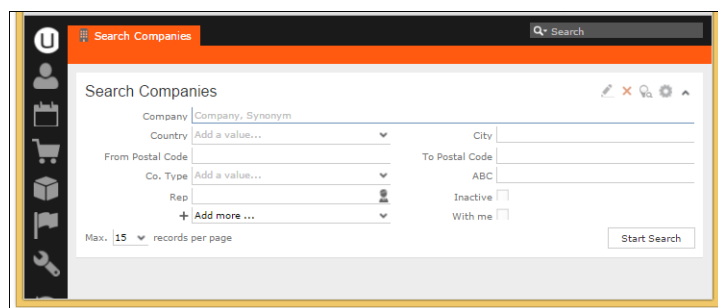
## Standard Headers

Find the list of standard headers in this topic.

Below is the list of standard headers:

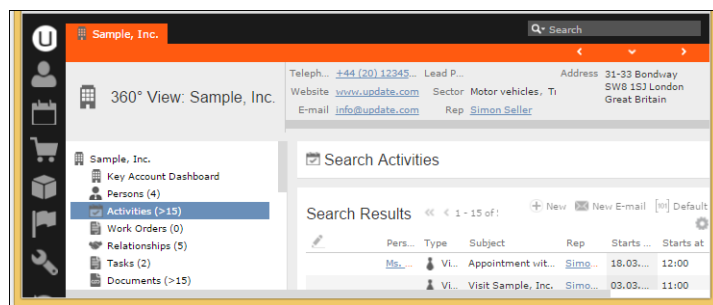
### Search/SearchLinked

Search pages displays the `Search` or `SearchLinked` header at the top of the page.



If a Search page is not linked to a record (e.g. the **Company** search called from the application menu), the `Search` header is used.

The `SearchLinked` header is used for linked searches, i.e. a link to parent record is defined in the `Search` action. For further details, see [link/linkId](#). Example: All MA linked to the current FI in a Tree view:

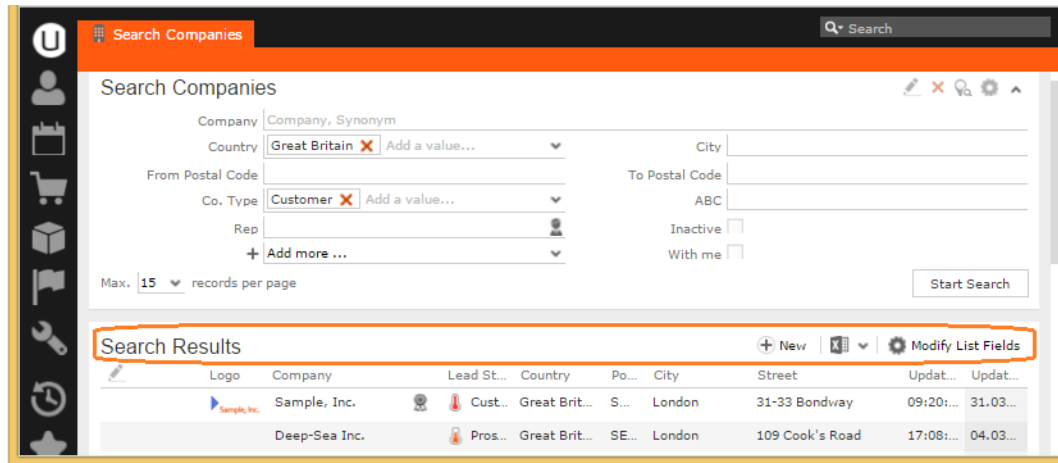


As you can define placeholders in the header text (see [Header Attributes](#) on page 326), the definitions of these headers are info area-independent, i.e. the `Search` and `SearchLinked` headers from the `DEFAULT` header group can be used. For further details, see [DEFAULT Header Group](#).

## SearchResults/SearchResultsLinked/SearchResultsSelect

The `SearchResults` header is displayed between the search mask and the results list. For linked searches, the `SearchResultsLinked` header is used, if available.

Fallback: `SearchResults`



This header usually contains the text "Search Results" and allows users to navigate within the result list. This header may also contain buttons. For further details, see [Header Buttons](#) on page 327. If the user clicks a button in the `SearchResults` header, the button's action is executed on the record(s) selected in the list. The linked record (or parent record) referenced in the action is the page's (optional) linked record. For a more detailed explanation, see [Defining an Action](#) on page 287.

The `SearchResults` header contains a standard **New** button to create a new record of the same type with the same link (if a linked record exists). This button uses calls the `ExpandNew` action with the `infoAreaId` parameter for the record's info area and the `link` parameter as link for the new record. For further details, see [Expand](#) on page 578.

When using this standard **New** button in the `SearchResults` header, you cannot use that header for both a search started from the application menu and a search started from a context menu for most info areas: Child info areas need a linked record, e.g. a company or person, which is not available from the application menu. In this case you need two different `SearchResults` headers: one for the action called from the context menu containing the standard **New** button, and one when the page is called from the application menu containing a **New** button launching a process or not containing a button.

To implement two different headers, either define two `Search&List` configurations or define an alternate `SearchResults` header in the `Search` action.

Example in `UPDATE_DEFAULT`: When creating a new activity (MA) from the application menu, the `SearchResults` header of header group `MA` is used. The **New** button in this header launches the `NewAppointment` process, where the user has to select a parent record for the activity.

When creating a new activity from a context menu (e.g. via `A_AllAppointments` in `M_Company`), the `SearchResults` header of header group `MA-KP,FI` is used. This header contains the standard **New** button described above.

If a `Search` action is called within a process and no special `SearchResults` header is defined replacing the standard `SearchResults` header, the `SearchResultsSelect` header is used.

Example in `UPDATE_DEFAULT`: The `SearchResultsSelect` header in header group `KP` contains two buttons: The `WizardNew` button fires event 2, the `WizardSearchViaCompany` button fires event 3.

## SearchSubList

The `SearchSubList` header is displayed below the search results

The screenshot shows a search interface for 'Search Companies'. It includes search filters for 'Company' (update\*), 'Country', 'From PC', 'To ZIP', and 'With me'. Below the filters is a 'Search Results' table with columns: Company, Lead Status, Country, Posta..., City, and Street. The table contains several rows, with 'update software AG' highlighted. Below the search results is a 'Related Data' section with a sub-table for 'Persons' with columns: Last Name, First Name, Function, Department, Area Code, Tel 1, Company, and E-mail 1. The 'Persons' table lists several individuals, including Angerer, Doe, Kaindl, Kellermann, Özalpin, and Yuceel.

list.

It controls which related data (e.g. linked info areas) is displayed when a record from the search results list is select and how the sub-lists (i.e. records from child info areas) are displayed. For further details, see [Related Data area \(Header Info Areas\)](#) on page 329. The header text and buttons for each sub-info area are defined per sub-info area in the corresponding `SubList` headers. For further details, see [SubList](#).

## SubList

The `SubList` header defines the buttons available for sub-lists.

The screenshot shows a sub-list header with tabs for 'Activities', 'Mailings', 'Addresses', 'Opportunity Participants', 'Relationships', and 'Inte'. Below the tabs is a table with columns for various data points and checkboxes. The table contains several rows, with the first row showing 'Visit' and '23:...'.

The layout of the sub-list rows is defined in the `Infoareas` section of the `SearchResults` or `ExpandChildren` header of the parent info area (**Fieldgroup** property). For further details, see [Related Data area \(Header Info Areas\)](#) on page 329.

## New

The `New` header is displayed when creating new records. Expand action template in New mode.

The default text is "New <Info Area>", e.g. "New Company".

A generic `New` header containing a `Save` button is defined in the `DEFAULT` header group. For further details, see [DEFAULT Header Group](#). The `New` header has the **Show buttons with text** and **Show buttons in extra line** check boxes checked: The buttons are displayed in a separate line with the button text visible.



For some info areas a special `New` header with a `Match-up` button is used instead. This calls the match-up function instead of simply saving the new record. Examples in `UPDATE_DEFAULT: FI, KP`.

To include the linked record in the header text, you need separate `New` headers for linked info areas. This is not defined in the `UPDATE_DEFAULT` configuration, because the link is always displayed below the details control (**Parent Data**). For further details, see [ExpandOptions](#).

## Expand

Expand pages display the `Expand` header at the top of the page.



## ExpandChildren

The `ExpandChildren` header is displayed below the `Expand` view when **Related Data** is selected.



It controls which related data (e.g. linked info areas) is displayed and how the sub-lists (i.e. records from child info areas) are displayed. For further details, see [Related Data area \(Header Info Areas\)](#) on page 329. The header text and buttons for each sub-info area are defined per sub-info area in the corresponding `SubList` headers. For further details, see [SubList](#).

## ExpandOptions

The `ExpandOptions` header defines the Related Data area below the details mask, allowing the user to view related records in the lower half of the page. You can display **Parent Data**, i.e. the parent links of the record and their context menu, or **Related Data** containing available sub-lists. For appointments the **Participants** area is available as well.

04.08.2012 12:00: Visit for Mr. Franz Müller (ABC)

Menu Edit Save Cancel Recurrence... Send as E-mail

Activity Type Visit

Date 04.08.2012

Time 12:00

Priority High

Result

Subject

FreeC1

Subject Details Links Marketing Activity Survey A

Rep ID Susanna Pfann

Participant ID Required Susanna Pfann

Optional

For Information

Text

Participants | Parent Data | Related Data

Add Company Add Person Add Linked Person Add from Favorites

T	S	Rep / Person / Company
!	ⓧ	Mr. Franz Müller
!	ⓧ	Susanna Pfann

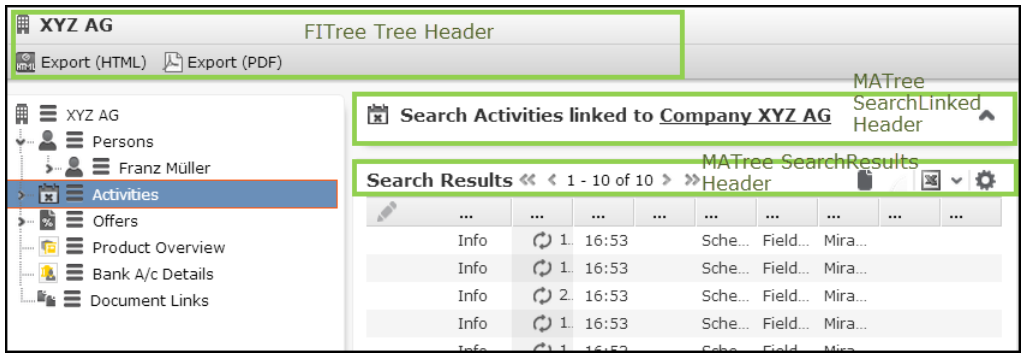
For information on configuring the `ExpandOptions` header, see [Header Links](#) on page 333.

For information on configuring the buttons for the **Participants** area, see [Participants Header \(MA\)](#).

The parent records that are displayed in the **Parent Data** area is determined by the Expand configuration of the current record (**View Parent Relations**), see [Expand Configuration](#) on page 352.

## Headers for TreeViews

At the top of a Tree page, the `Tree` header is displayed similar to the `Expand` header of Expand pages.



When clicking on an info area node in the tree, a Search view for this info area is displayed using the `SearchLinked` and `SearchResultsLinked` headers from the header group specified in the corresponding `Search&List` configuration. For further details, see [SearchResults/SearchResultsLinked/SearchResultsSelect](#) and [Info Area Settings](#). If no `SearchResultsLinked` header is defined, the `SearchResults` header is used as a fallback.

When clicking on an existing record in the tree, an Expand view with data of that record is displayed using the `Expand` header from the header group specified in the corresponding `Expand` configuration. For further details, see [Expand](#) on page 578 and [Record Settings](#).

When creating a new record from within the tree, the `New` header from the header group specified in the corresponding `Expand` configuration is used. For further details, see [New](#) and [NewCommand](#) on page 317.

### SimpleList

The `SimpleList` header is used for the section headers in the Favorites popup, in the Checked Out Documents popup list and as header of Most Recently Used lists on Process Ask pages.

The default text is "{INFOAREAIMAGE} {INFOS}".



### Special Headers

Special headers are the special functionality header.

Special headers provide some special functionality, e.g. the various headers for the **Calendar** page or the `Participants` header for MA.

Special headers can be info area-specific (`Participants` header for MA, various special headers for quick add pages etc.) or info area-independent (headers used by the calendar, headers for processes etc.).

Below are the list of special headers:

## Analysis Headers

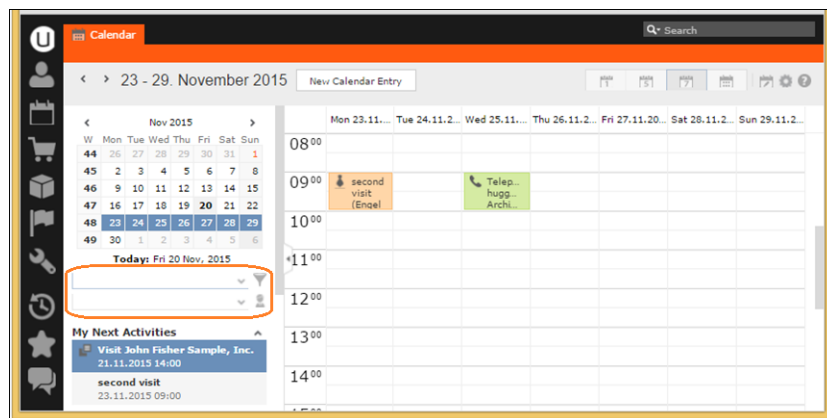
- `AnalysisHeader`: Contains the text and buttons for the Analysis Editor.
- `AnalysisHeaderRecord`: Used instead of `AnalysisHeader`, if the analysis was called from a record's context menu. This Header displays the text "Analysis <name> for <record>" instead of just "Analysis <name>".
- `AnalysisFooter`: Contains the `AnalysisEdit_Run` button in the bottom right corner of the Analysis Editor.
- `AnalysisResult`: The analysis result's header containing the name of the analysis and the **Show Chart** and **Show Table** toggle buttons (`Toggles(M_AnalysisResultViews)`).
- `AnalysisResultRecord`: Used (instead of `AnalysisResult`) if a link record is passed to the analysis result. For further details, see [link/linkId](#).

## Calendar Headers

### CalendarAdjustments

This special header controls the filter options on the Calendar sidebar. It contains the `RepSets` and `FilterSets` buttons for displaying the rep and filter set drop-down lists.

Any other button configured in this header is not be visible.



### CalendarHeader/DayWeekPlanningCalendar

These special headers contain the buttons to control the calendar view (**Day**, **Work Week**, etc. configured by `Toggles(M_CalendarViews)`) for the **Calendar** and the **Day&Week Planning** pages respectively. For further details, see [Calendar](#) on page 402 and [DayWeekPlanning](#) on page 410.

Users can select multiple appointment records on the **Calendar/Day&Week Planning** pages: You can add buttons for exporting these records or launching a process that makes use of the selected records. For further details, see [ProcessAsk Action Parameters](#). If no appointment record is selected, all records of the current view are used for the export/process.

The column names for exports from the **Calendar/Day&Week Planning** pages are configured in the `calendar_export_text` Text Group. For further details, see [Texts](#) on page 176.

## CalendarSidebar

This special header contains the upcoming items definition for the Calendar sidebar, specifying which **Activities**, **To-Dos**, **Tasks** and **Absences** are listed on the sidebar.

Order	InfoArea	Link	CRM Proc	Fieldgroup	Header	Text	Filter	Tab Style	SubListParams
↓	✗ Activity (MA)	<input type="checkbox"/>	all	no value	no value	My Next Activities	MA.Sidebar		{ "sortFields": ["De
↓	✗ Task (A1)	<input type="checkbox"/>	all	no value	no value	My Next Tasks	A1.Sidebar		{ "sortFields": ["Du
↓	✗ To-Do (TD)	<input type="checkbox"/>	all	no value	no value	My To-dos	TD.Sidebar		{ "sortFields": ["Du
↑	✗ Absence (A3)	<input type="checkbox"/>	all	no value	no value	My Absences	A3.Sidebar		{ "sortFields": ["St

The following settings are available:

- **Text:** Defines the caption to be displayed above the list, e.g. "My next activities" for MA.
- **Filter:** Select a **Filter** for each info area to determine which records are displayed (default `XX.Sidebar` for all four info areas).

You can set the filters to be used via the Web Configuration parameter as well (by defining the options as `<InfoAreaId>.<filtername>`). The filters defined in the `CalendarSidebar` header overrule the filters defined in the `Calendar.SidebarItems` Web Configuration parameter.

- **SubListParams:** Define the sort order of the upcoming items using the JSON syntax. For further details, see <http://www.json.org>.

### Examples

ascending:

```
{ "sortFields": ["DueDate", "DueTime"] }
```

descending:

```
{ "sortFields": ["-DueDate", "-DueTime"] }
```

Use the Web Configuration parameter `Calendar.SidebarItems` to determine which upcoming items are displayed in the calendar sidebar. For further details, see [Calendar.SidebarItems](#). Removing them from the header definition has no effect.

Table captions for upcoming calendar items: `<InfoAreaId>.Sidebar`

Fallback: `<InfoAreaId>.Calendar` - if this does not exist either, the default table caption `<InfoAreaId>` is used.

Use the Web Configuration parameter `Calendar.SidebarMaxRecords` to determine the number of upcoming items to be listed. For further details, see [Calendar.SidebarMaxRecords](#).

## CircleOfInfluence

This special header is used by the relationship graph (Circle Of Influence). For further details, see [CircleOfInfluence](#).



## NotificationCenter

The special header `NotificationCenter` contains the configuration of Aurea CRM.Web's Notification Center ("Inbox").

The sub info areas in the header definition define which tabs are shown in the Inbox (default: A1, ER, MD, TD). The order of the sub info area definitions determines the order of the tabs. The definition of the **System** tab (for displaying system notifications) uses a dummy info area that is replaced by the `Form` parameter.

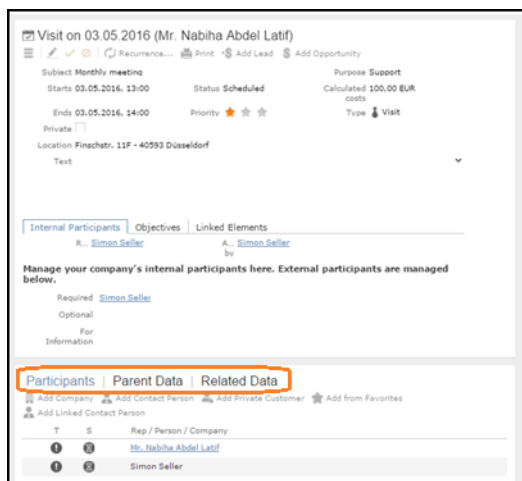
The value specified in the Web Configuration parameter `Jobs.PollingInterval` determines the interval for polling for new Messages/Reminders/Tasks/To-Dos. For further details, see [Jobs.PollingInterval](#).

The timeout for displaying notifications can be configured per notification type using the Web Configuration parameters `Notification.TimeoutError`, `Notification.TimeoutInformation` and `Notification.TimeoutWarning`. For further details, see [Notification.TimeoutError/Information/Warning](#).

For details on customizing notifications in Aurea CRM.Web (subject, text, target, facility, severity etc.), see the article "The Notification service" at <https://support.aurea.com>.

## Participants Header (MA)

The special header `Participants` configured in header groups `MA`, `MATree`, `MAEmail` etc. contains the buttons displayed in the **Participants** area of activity (MA) records.

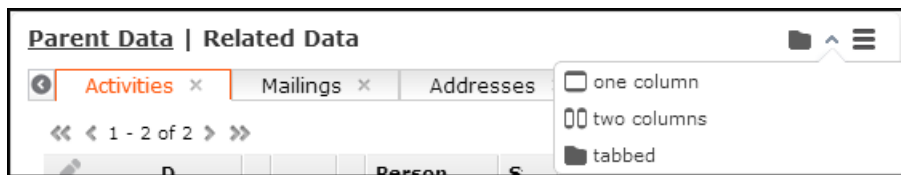


## Questionnaire

This special header is used by the Survey page (`surveyconduct.aspx`). By default, the header text consists of the name of the survey plus description (**Text** field in F1).

## RelatedDashboardPanel

This special header contains the buttons allowing users to customize the Related Data area for Search and Expand views. For further details, see [Related Data area \(Header Info Areas\)](#) on page 329. For Expand views these buttons are automatically appended to the `ExpandOptions` header.



Depending on the **Tab Style** settings (e.g. `{allowRemove: false, allowCollapse: true}`), users can switch between views (one column, two columns and tabbed), expand/collapse, hide and reorder the sub-info areas. The user's settings are saved per client and separately once for one column and tab views and once for the two columns view.

## SearchFilter

This special header contains the buttons of the Filter definition dialog.

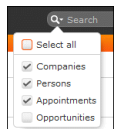
## Quick Search

You can search across multiple info areas using Quick Search.

The Quick Search allows Aurea CRM.Web users to quickly search across multiple info areas in the Aurea CRM.Web database. The Quick Search is displayed in the top right corner of Aurea CRM.Web.



Users can select the info areas they want to search.



## Configuring the Quick Search

Learn to configure quick search.

The Quick Search configuration is specified using the JSON syntax. For further details, see <http://www.json.org>.

The default Quick Search is defined in the `QuickSearch.json.js` file located in the `..web\data\settings` directory.

To customize the Quick Search, create your own configuration file and apply it via the Web Configuration parameter `QuickSearch.Definition`. For further details, see [QuickSearch.Definition](#).

The `QuickSearch.json.js` file contains search providers defining the following:

- the Quick Search drop-down list (labels, defaults)
- the actual search definitions via AQL statements (see [AQL Syntax](#) on page 882) and/or predefined queries

```
{
  searchProvider: [
```

```

    new u8.Crm.SearchProvider({
      isActive: true,
      label: u8.services.schema.idToLabelPlural("FI"),
      useFirstSearchTermForAllParameters: true,
      query: new u8.Crm.QueryCommand({
        statement: "select (Company, City, Country) from (FI as root) where
(Company = '$parValue' or Synonym = '$parValue' or Country = '$parValue')",
        catalogValueEncoding: "Text"})
    }),
    new u8.Crm.SearchProvider({
      isActive: true,
      label: u8.services.schema.idToLabelPlural("MA"),
      maxRows: 10,
      query: new u8.Crm.QueryCommand({
        name: "QuickSearchPredefinedQueryName"})
    })
  ]
}

```

Each search provider specifies an entry of the Quick Search drop-down list (i.e. an info area to be searched). The **Select all** option is hard-coded and cannot be removed.

In the BTB vertical the standard Quick Search definition contains FI, KP, MA and Y1. Y1 is not searched by default. The standard definitions for the OTC and FS verticals contain FI, KP and MA.


---

**Note:** Info areas that are locked in the current user's access rights definition are not searched.

---

The following parameters are used:

Name	Type	Description
<code>isActive</code>	boolean	Determines if the search provider is used by default (i.e. the entry in the Quick Search drop-down list is checked).  <b>Default:</b> <code>false</code>
<code>label</code>	string	Sets the title of the quick search result for the provider.

Name	Type	Description
maxRows	int	<p>Optional: The maximum number of rows to be displayed as a search result.</p> <p><b>Default:</b> 5.</p> <p>If more than <code>&lt;maxRows&gt;</code> records are found, the title displays <code>(&lt;maxRows&gt;+)</code> as a hyperlink.</p>  <p>Clicking on the link displays all search results in a <code>QueryRun</code> page.</p>
useFirst-SearchTermForAll-Parameters	boolean	<p>If set to <code>true</code>, the first search term entered is used for all search fields, e.g. to handle searches for person names consistently.</p> <p><b>Default:</b> <code>false</code></p>
query	u8.Crm.QueryCommand	<p>The query that is executed (define the query as <code>&lt;name&gt;</code> or <code>&lt;statement&gt;</code>).</p>
name	string	<p>The name of a predefined query.</p>
statement	string	<p>The query statement specified using AQL, see <a href="#">AQL Syntax</a> on page 882. (The <code>\$parValue</code> object contains the search string entered by the user).</p> <p><code>\$parValue</code> automatically adds a wildcard <code>*</code> at the end of the search string (<code>string*</code>). To enclose the search string in wildcards (<code>*string*</code>), use <code>*\$parValue</code> in the query statement.</p> <hr/> <p><b>Note:</b> <code>*\$parValue</code> is only supported by text fields and only in AQL queries (i.e. not in predefined queries).</p> <hr/>

The Quick Search definition only supports text, number and variable catalog fields. These field types may be combined in one definition (in both the search and output fields. If an unsupported field type is used, a warning is written to the client log.

To hide the Quick Search, remove the value (path and file name) from the `QuickSearch.Definition` Web Configuration parameter or remove all `searchProvider` definitions from the \*.js file.

---

**Note:** Since the Quick Search queries multiple fields from multiple info areas, make sure to use indexed fields and limit the amount of info areas and fields to avoid performance problems.

---

# 3

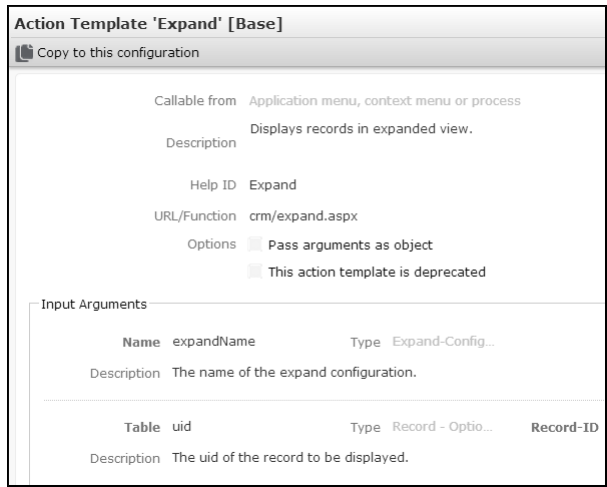
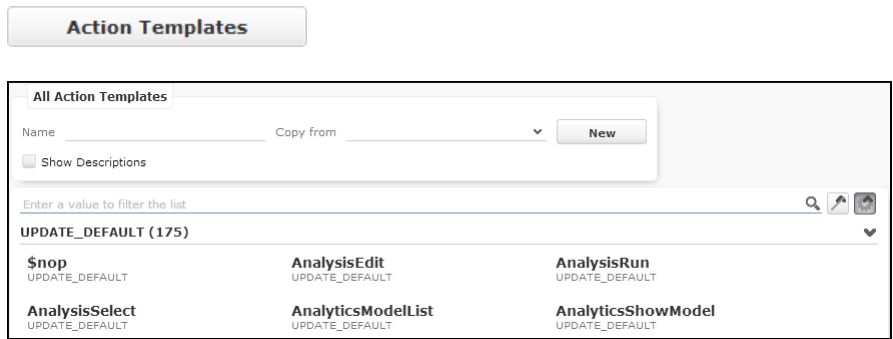
## Using CRM

CRM.Web is easy to use and offers numerous features to facilitate your daily activities.

### Action Templates

Learn about the various action template and their configuration and uses.

On the CRM.Designer main page, click the **Action Templates** button to display all available action templates.



The **Callable from** setting determines in which context an action template can be used, e.g. "Application menu, context menu or process", "Data Provider". According to this setting the action templates are listed in the **Action Template** drop-down list.

Example - when configuring a menu action. For further details, see [Defining an Action](#) on page 287.

**Note:** Legacy action templates from update.seven are flagged as deprecated. When customizing an action that uses a deprecated action template, this action template is displayed at the end of the list of available action templates prefixed with an "!". Change the action to use a valid Aurea CRM action template instead.

**Note:** Legacy action templates exist for migration reasons only and should no longer be used with Aurea CRM.Web, as very limited bug fixing and no new features are provided.

The following chapters provide information about Aurea CRM.Web's standard action templates and their parameters.

## Expand

The Expand action template allows a record to be displayed in details view.

The `Expand` action template (`expand.aspx`) can be used for actions defined for menu entries, buttons and within processes. For further details, see [Defining an Action](#) on page 287. The contents and functionality of an `Expand` page is configured via the Expand configuration referenced in the action. For further details, see [Expand Configuration](#) on page 352.

The screenshot shows the Aurea CRM interface for an activity. The main window title is "01.06.2013 15:23: Telephone for Mr. Fritz Müller (XYZ AG)". The activity details are as follows:

- Activity Type: Telephone
- Date: 01.06.2013
- Time: 15:23
- Priority: High (3 stars)
- Status: Scheduled
- End Date: (empty)
- End Time: (empty)
- Create To-Do:
- Private:

The 'Details' tab is active, showing the following fields:

- Rep ID: Lisa Simpson
- Participant ID: (empty)
- Required: Lisa Simpson
- Optional: (empty)
- For Information: (empty)

Below the details, there are sections for 'Participants', 'Parent Data', and 'Related Data'. The 'Participants' section shows a table with columns for Rep/Group ID, Due on, Due at, and Subject.

Rep/Group ID	Due on	Due at	Subject
Peter Parker	01.07.2013	18:00	call again

The `Expand` action's functionality (show, new, update etc.) is determined by the `mode` parameter. For further details, see [mode](#).

UPDATE\_DEFAULT contains predefined action templates for "show" and "new" modes:

- **Expand:** Displays a record for a given info area and record ID in details view.

**Note:** Set `relationName` to `$Parent` to display a record's parent record.

- **ExpandNew:** Creates a new record from a given info area with up to two optional linked records.

## Expand Configuration

Learn about the use of Expand Configuration

The Expand configuration is used for displaying the following views of a record:

- Expand view (Tab view)
- **QuickView** - a read-only popup view of the record.
- Popup Expand view - an editable popup view of the record, in UPDATE\_DEFAULT e.g. used for calendar items. For further details, see [createNewPopup](#).
- read-only Expand view on Process Ask pages. For further details, see [ProcessAsk](#) on page 579.

**Configuration for 'MA'**

Field Group: MA Show MA

Header Group: MA Show MA

Table Caption: MA Show MA

Menu: M\_Appointments Show M\_Appointments

Info Area Default Action: default

Info Area Default Menu: use default menu

Description:

---

View Parent Relations

1. Company (FI)	X	↓
2. Person in Company (KP)	X	↓ ↑
3. Property (OJ)	X	↓ ↑
4. Opportunity (Y1)	X	↓ ↑
5. Marketing Activity (AK)	X	↓ ↑
6. Ticket (KM)	X	↓ ↑
7. Installed Base (IB)	X	↓ ↑
8. Offer (PR)	X	↓ ↑
9. Activity (MA) (LinkId=2)	X	↑

10. Text (04) Tablecaption-Name:  Show all

LinkId:

---

Breadcrumb-Parent: Person in Company (KP) Tablecaption-Name:  Show all

LinkId:

Add Condition for this Field: Activity Type (0)

Field	Cmp	Value	Alternative
X ↓ ↑ 1. Activity Type (0) +	=	Telephone CatCode = 0	MATel
X ↓ ↑ 2. Activity Type (0) +	=	Letter CatCode = 2	MALetter
X ↓ ↑ 3. Activity Type (0) +	=	Visit CatCode = 1	MAVisit
X ↓ ↑ 4. Activity Type (0) +	=	E-mail CatCode = 5	MAEmail

+ Add an alternative



The following naming conventions are used for Expand configurations:

- When a record is opened in Quick View, the Expand configuration `<InfoAreaId>.QuickView` is used.

Fallback: `<InfoAreaId>`.

- The Expand view of a calendar item opens as a popup (single click on the item's icon): The Expand configuration used for this popup is `<InfoAreaId>.Calendar.UPDATE_DEFAULT` contains the following definitions: `A1Cal`, `A3Cal`, `MACal`, `TDCal`

Fallback `<InfoAreaId>Cal` - if this does not exist either, `<InfoAreaId>` is used.

An Expand configuration consists of the following items:

- **Field group:** Defines the layout of the details view. The Details control of the specified field group is used. For further details, see [Field Groups](#) on page 188.
- **Header group:** Defines the header group of the `Expand` or `New` header depending on the mode (see [mode](#)) displayed above the details control. For further details, see [Standard Headers](#) on page 337. The `ExpandOptions` header of this header group defines the display options for the bottom half of the window. The `Expand-Children` header of this header group defines which related data (e.g. linked info areas) is displayed.
- **Table caption:** Defines the texts displayed in the header. For further details, see [Table Captions](#) on page 265. Define table captions when using alternate expand definitions to display different header texts depending on the record type. For further details, see [Alternative Expand Configurations](#).
- **Menu:** Defines the context menu displayed for the record. For further details, see [Context Menus](#) on page 292.
- **InfoArea Default Action:** Defines the default action that is executed when a user double-clicks on a record in a list or clicks on a link in a header. For further details, see [Default Action](#) on page 289. Define the default action for each info area in the default Expand configuration (i.e. the configuration with the same name as the info area ID e.g. MA for activities).

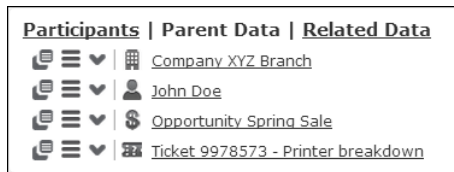
If you do not specify an **InfoArea Default-Action** (i.e. "default" selected), the action defined in the `Record.DefaultAction` Web Configuration parameter is used. For further details, see [Record.DefaultAction](#).

---

**Note:** Do not define `A_ShowDefaultAction` as default action. This results in an infinite loop.

---

- **InfoArea Default-Menu:** Defines the info area's default menu. This menu is used as a fallback for lists and sub-lists. The Expand page always uses the menu selected under **Menu**.
- **Description:** An internal description only visible in CRM.Designer.
- **View Parent Relations:** Contains the parent records of the current record.

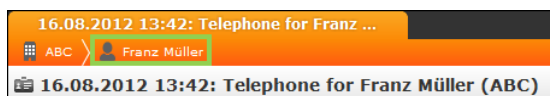


The existence of each configured parent relation is checked on the server. If a record is linked to a parent record, the link to this parent is displayed in the **Parent Data** area and in `D_OpenLinked`, if configured. For further details, see [ExpandOptions](#) and [D\\_OpenLinked](#).

**Note:** To display the records of the **Parent Data** area as links and with context menus and icons, you need to define Expand configurations for those info areas. If no Expand configuration exists, the parent records are displayed as text using the info area's default table caption if no table caption exists, the reference text defined in Aurea CRM win is used.

**Note:** If both **Person (PE)** and **Person in Company (KP)** are defined as parent relations and if a person is linked to a company, KP is displayed. Otherwise PE is displayed.

- **Breadcrumb-Parent:** Allows you to define the parent info area that is shown in the breadcrumb trail since breadcrumbs are hierarchical, you can specify only one parent.




Further parent info areas are determined automatically from the data model and/or the breadcrumb definition of the parent info areas. If you do not specify a **Breadcrumb-Parent**, no breadcrumb trail is visible. The breadcrumb trail displays the standard table caption of the selected info area unless you specify a different **Tablecaption-Name**. Use the **LinkId** to specify another than the default link. For further details, see [Specifying Link IDs](#) on page 20.

**Note:** If "Person in Company (KP)" is selected as breadcrumb parent and no KP record is found, the person record (PE) is automatically used instead.

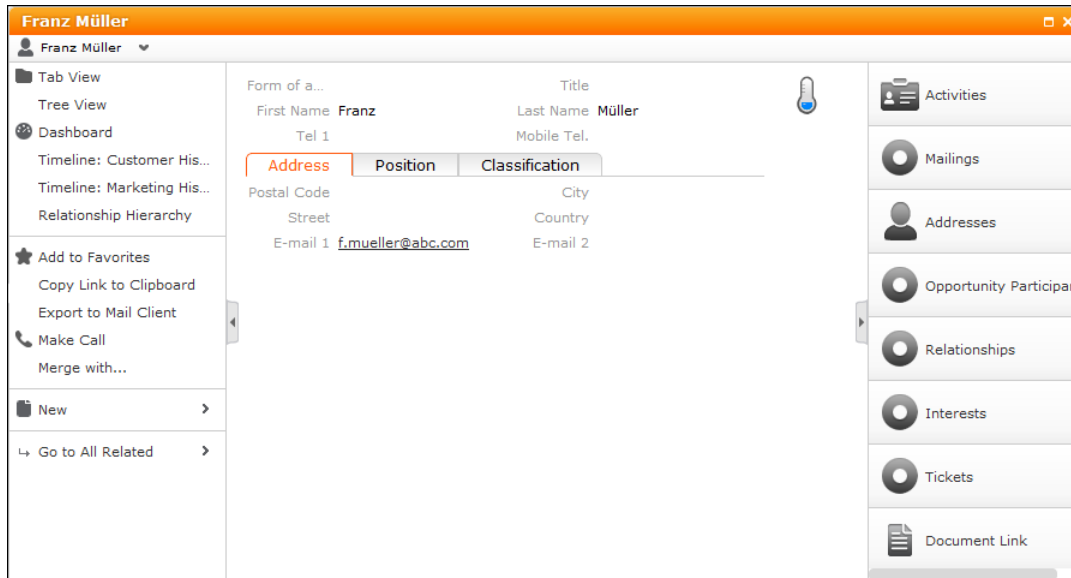
- **Alternatives:** Allows the implementation of "filter masks" i.e. different masks depending on the values of the current record. For further details, see [Alternative Expand Configurations](#).

## QuickView

The QuickView consists of a read-only popup view of a record.

Quick View is opened by click+hold on a link to a record or by clicking  (**QuickView**) in list rows and the Parent Data area.

The QuickView offers an easy way to get details about a record without leaving the current view. It allows users to "drill down" into related data and execute the available menu actions.



The following configuration units are used to configure the QuickView:

- To define a specific Details control for a QuickView: Create a field group `<InfoAreaId>.QuickView`. The Details control of this field group is used.
- To customize the entire QuickView: Create an Expand configuration `<InfoAreaId>.QuickView`. The context menu (i.e. left panel) and the Details control from this Expand configuration are used. The related info areas (i.e. the right panel) are defined by the `ExpandChildren` header from the header group specified in this Expand configuration.

To hide individual menu items from the QuickView enter `-QuickView` in the menu actions **Info Area** field. Example in `UPDATE_DEFAULT: A_Delete`. For further details, see [Menu Actions](#) on page 295.

To hide individual buttons, check **Hide button in Quick View** in the button's definition. For further details, see [Configuring a Button](#) on page 305.

## Alternative Expand Configurations

Alternative expand configurations allow different settings of the Expand action template depending on the content of the displayed record.

This feature of Expand action template is commonly referred to as "filter masks".

In `UPDATE_DEFAULT` alternative expand configurations are used as follows:

- for **Activity** records (MA) depending on the **Activity Type**.
- for distinguishing between newly created (`B7New`) and existing (`B7`) **Stock Control** records (`B7New` does not contain the **Quantity factor** field because otherwise it would not be updated according to the selected **St.V.Type**).

To define alternate expand configuration, at least two Expand configurations for an info area must exist (e.g. MATel and MAlLetter).

1. Select a field from the **Add Condition for this Field** drop-down list.
2. Click + to define the conditions to be met and assign the alternative Expand configuration.

Use the variables \$curRep, \$curDay and \$curTenant as comparison values for rep, date and tenant fields.

Use the **+ AND** button to combine multiple criteria.

The conditions are checked in the order they are defined.

- Use the and buttons to change their order.

The following operators are available (**Cmp** drop-down list):

<p>=, &gt;=, &lt;=, &lt;, &gt;, &lt;&gt;</p>	<p>These operators support neither wildcards nor regular expressions (thus a condition company name = u* does not return all companies starting with u but only companies where the name literally is u*).</p> <p>If you need to use wildcards, you have to use LIKE with regular expressions.</p>
<p>LIKE</p>	<p>LIKE supports only regular expressions (attention: case-sensitive syntax).</p> <p>E.g. to find all companies starting with an uppercase U, you have to specify the regular expression LIKE ^U.*. To find all company names starting with U, regardless of case, use LIKE ^[uU].*</p> <p>For the valid regular expression syntax. For further details, see <a href="http://msdn.microsoft.com/en-us/library/ae5bf541(v=vs.100).aspx">http://msdn.microsoft.com/en-us/library/ae5bf541(v=vs.100).aspx</a>.</p>

**Note:** If you create "circular" alternative definitions (i.e. configurations referencing each other), the configuration where the circle is closed is the one that is applied. Example: If the definitions are "in configuration FI: if country = Austria use the FIAustria configuration" and "in configuration FIAustria: if company name starts with U use the FI configuration", there is a circular reference for companies from Austria whose name starts with U. In this case, the loop closes with the second condition, thus the FI configuration is applied to the mask.

## Expand Action Parameters

Learn about the input arguments supported by the Expand Action Parameters.

The Expand action supports the following input arguments:

## expandName

The name of the Expand configuration. For further details, see [Expand Configuration](#) on page 352.

If undefined, the Expand configuration `<infoAreaId>` is used. This is the default Expand configuration for the info area from which the action is called.

## uid

The unique ID of the record to be displayed.

The `uid` consists of a record's `infoAreaID` plus `recordID`, thus uniquely identifying the record in the database.

Possible values:

- `Record`: Displays the current record.
- `Linked`: Displays the parent record of the current record.

## Example

Selecting a person from the sub-list of a company record (**Related Data**):

`Record` displays an Expand view of the selected person record.

`Linked` displays an Expand view of the selected person's parent company.

## mode

The mode of the Expand view:

- **New**: Expand view for creating a new record (as used in the `ExpandNew` action template). For further details, see [ExpandNew Action Parameters](#) on page 362.
- **NewMy**: Depending on the parent record (see [link/linkId/link2/linkId2](#)) the new record is created as follows:
  - parent FI: The new record is created as a child of the current user's company record (`u8.session.identity.companyUid`)
  - parent KP: The new record is created as a child of the current user's person record (`u8.session.identity.personUid`)
  - any other parent: fallback to `mode=New`

Example in `UPDATE_DEFAULT: D_NewAppointment_My` and `D_NewAppointment_My_FI` in `M_CalNew` called by the **New Calendar Entry** button. These menu actions create new activities for the current user and the current user's company respectively.

- **NewOrShow**: Expand view for creating a new record or updating an existing one. Use this mode for 1:1 links only, in situations where only one linked record is allowed, example - for a marketing activity (AK) only one execution record (AD) may exist. In addition, you need to pass a link record to the expand action. For further details, see [link/linkId/link2/linkId2](#).

- **Show:** Expand view that allows users to update an existing record (default). For further details, see [Expand Action Parameters](#) on page 356
- **Update:** Expand view for updating an existing record with values specified in the action. Use the following parameters to create `Expand` actions with `mode = Update`:
  - [copySourceFields/copyTargetFields](#)
  - [defaultFields/defaultValues](#)
  - [emptyFields](#)
  - [autoSave](#)

---

**Note:** For `Expand` actions with `mode = Update` CRM.Core default values (i.e. default values defined in the Aurea CRM win **Rights** module) are not applied and plausibility check triggers are not executed.

---

### relationName

The link relation used to determine the record to be displayed.

Enter `$Parent` to display a record's parent record.

Enter `$GenericLink` to display a generic link record.

Other possible values: `$RelatedCompany`, `$RelatedHuman`, `$RelatedPerson`, `$RelatedAppointment`. For further details, see [Link Names in JavaScript](#) on page 22.

### scripts

A comma-separated list of one or more (local) JavaScript files to be loaded to the page.

### gotoAfterSave

Defines the view to be displayed or menu action to be performed after saving. This overrides default.

Syntax - `Menu:<menu action>`

Example: `Menu:A_ExpandTree` opens a record's Tree view after saving.

### history

If set to `none`, the page is not added to the (page-related) browsing history.

**Default:** None

### saveProvider

The name of the provider that is used to save the records in the Expand view. Use this parameter to specify your own ASPX-page if you want to use a different channel.

## emptyFields

Only for `Expand` pages with `mode=Show` or `Update`. A comma-separated list of fields (`<field name or number>`) whose contents are to be cleared upon updating the record.

## createNewTab

Controls whether an action is executed in a new tab. The following values are available:

- `false`: To create a new tab `Ctrl+Click` has to be used.
- `true`: The action (example - displaying a record or opening the calendar) is always executed in a new tab.

The tab title is retrieved from the header as defined in the action (example - `headerGroupName` in a `Tree` action). If no header is found, the label of the button or menu item that called page is used instead. For more details, see the article “How to set the Title of a Tab”, see <https://support.aurea.com>.

To specify if and how the new tab should be reused if the same action is called again, use the following syntax (JSON):

```
{"reuseId": "<tabId>", "reuseRecord": true, "refresh": true}
```

`reuseId`: Executes the action in a tab with the specified ID. `<TabId>` can be any name you choose. If no tab with such an ID exists, a new tab is created, otherwise Aurea CRM.Web switches to the existing tab.

`reuseRecord`: For generic menu actions like `A_Expand` or `A_ExpandTree`, where you need a different ID for each record, you can specify `"reuseRecord": true`. Example: For `A_Expand` in `M_Company` this would open each company in a separate tab. But if the user tries to expand a company which is already open, the respective tab is activated instead.

`refresh`: Defines whether the re-used tab is not only brought into the foreground but also refreshed. Example: `{"reuseId": "ExpandView", "refresh": true}` for an expand action refreshes the tab to always display the current record.

**Default:** False

---

**Note:** If a menu action with a definition for `createNewTab` (example - `{"reuseId": "myTabId", "refresh": true}`) is executed via `Ctrl+Click` these parameters are used as well.

---

**Note:** To automatically close a tab, define the parameter `onProceed` as input argument and set it to `$back, >> onProceed`. This closes the tab if the user clicks **Save** after editing a record in a new tab, and automatically navigates back to the previous tab.

---

## createNewPopup

If set to `true`, the action (example - `A_Expand`) is executed in a popup. If set to `false` or undefined, the action is executed on the current page.

**Default:** undefined

Example in UPDATE\_DEFAULT: If the user clicks to edit an item in the calendar, the menu action `A_Edit<InfoAreaId>_Calendar` (configured for A1, A3 and MA) is executed. These menu actions are by default configured to create a new popup.

Instead of `true` you can specify an options string defining the popup's visualization (example - `createNewPopup = title(My Popup);width(0.5);height(0.2)`).

The following parameters are supported:

- `title`: The header text for the popup as a string. Alternatively, you can specify any CRM.Designer text using the syntax `TXT_<textgroup>_<number>`. If undefined, the page's name is used as title (example - "Expand").
- `width`: The width of the popup. Integers are interpreted as pixels, floating-point values (between 0.0 and 1.0) are interpreted as a percentage of the browser windows's size. If undefined, 60% of the window width are used as default.
- `height`: The height of the popup. Possible values and default as for `width`.

The screenshot shows a CRM application interface. At the top, there's a search bar for 'Search Companies' with fields for 'Company up\*', 'Country', 'From PC', and 'To ZIP'. Below this, a search results popup is open, titled 'Expand' and 'Company update Software AG'. The popup has a menu bar with 'Menu', 'Edit', 'Save', 'Cancel', 'Copy', and 'Print'. The main content area shows company details for 'Company update Software AG' with fields for 'Street', 'Postal Code', 'City', and 'Country'. Below this is a table with columns 'Address', 'Type', 'Classification', and 'Event'. The table contains several rows of data, including 'Country C...', 'Area Code', 'Telephone', 'Fax', 'E-mail 1', 'E-mail 2', 'Rep 1 ID', 'Rep 2 ID', and 'Website'. Below the table is a 'Related Data' section with a tabbed interface showing 'Persons', 'Activities', 'Opportunities', 'Offers', 'Orders', 'Installed Bases', and 'Service C...'. The 'Persons' tab is active, showing a table with columns 'Las...', 'Firs...', 'Fun...', 'Dep...', 'Are...', 'Tel 1', 'E...', 'Co...', 'Cou...', and 'City'. The table contains several rows of data, including 'Angerer', 'Simeon', 'Kaindl', 'Christoph', 'Kellerm...', 'Gerald', 'Özaltin', 'Ramazan', 'Yuecel', 'Iteris', and 'heyhey'. At the bottom of the popup, there's a 'Last Name' field with a dropdown menu showing 'Angerer' and 'Kaindl'.

There can only be one popup at a time: If a navigational action is executed in the popup (example - `A_AddPerson`), the popup automatically closes and the action is executed in a new tab. Therefore, suppose if you execute a `Search` action in a popup and provide an `ExpandNew` action within the search, the search popup is closed and the `ExpandNew` action is executed in a new tab. To implement this scenario using popups only, you need to define a process containing the `Search` and `ExpandNew` actions as process steps. For further details, see [Processes](#) on page 547.

To close a popup, define the parameter `onProceed` as input argument and set it to `$back, >> onProceed`. This closes the popup if the user clicks **Save** after editing a record in a popup.



---

You can define `createNewPopup` for all navigational actions, i.e. any action referring to a URL.

---

**Note:** Creating a new tab via `Ctrl+Click` or `createNewTab` always takes precedence over `createNewPopup`.

---

### viewMode

Determines the view mode of the page.

Available values: `Edit`, `Read`.

**Default:** `Read`.

Users can switch from View to Edit mode by clicking the **Edit** button. For further details, see [Edit](#) on page 41 .

### initialTab

Determines the active tab of the Details control when initially displaying a record in Expand view.

Specify the tab number (0-based). Example - `initialTab = 3` activates the fourth tab of the Details control.

### initialSubListTab

Determines the active sub-list tab when initially displaying a record in Expand view.

You can specify either the tab number (0-based) or the info area ID: `initialSubListTab, 3` activates the fourth sub-list tab upon opening the page, `initialSubListTab, MA` activates the **Activities** sub-list.

---

**Note:** It is recommended to use the info area ID since this continues to work even if the tab order is changed at some later point in the project.

---

### fieldGroupName

Allows you to use a different field group than configured in the Expand configuration.

### proceed

Set this argument to `off` in order to prevent the page from proceeding after a successful save or delete.

### onProceed

Use/define the `onProceed` parameter to control a page's behavior when its `proceed()` method is called.

Defining `onProceed = $back` results in the following behavior:

- If `createNewPopup = true` is defined for the action, the popup closes. For further details, see [createNewPopup](#).
- If `createNewTab = true` is defined for the action, the tab closes and Aurea CRM.Web navigates back to the previous tab.
- If neither `createNewPopup` nor `createNewTab` is set to true, the last page of the current tab is loaded again.

### validationSummaryPosition

Use this parameter to control the positioning of a validation message for an action. For example, saving a record results in a validation message about the save action. You can select one of three positions: **Top**; **Bottom**; **Hidden**.

---

**Note:** This parameter is used to control action-specific validation messages. For global control of validation message for all actions use the web configuration parameter [Expand.ValidationSummaryPosition](#).

---

Type: Value-Optional

### Expand.RefreshCurrentTabOnSave

When the content of an Expand template is saved, it reloads the current tab. Thus, all related child data is refreshed automatically without the need to reload the page.

**Type:** Boolean

**Default:** False

## ExpandNew Action Parameters

Learn about the ExpandNew Action Parameters.

In addition to the parameters listed under [Expand Action Parameters](#) on page 356, the `ExpandNew` action (`mode=New`) supports the following input arguments:

### infoAreald

The info area where the record is created.

This parameter is mandatory for `ExpandNew` actions.

### link/linkId/link2/linkId2

`link`: Record to which the new record is linked (`Record` or `Linked`).

`linkId`: ID of the link used for `link`. This parameter is only relevant if multiple links between two info areas exist in the data model (example - multiple FI-links in KM). For further details, see [Link IDs](#) on page 19.

Use `link2` and `linkId2` to specify a second link record.

**Default**=-1.

## linkRelationName

The name of a relation used to determine the link record, example - `$Parent`. For further details, see [Link Names in JavaScript](#) on page 22.

## copyUid

Defines the record from which data is copied to the new record (`Record` or `Linked`). `copySourceFields` and `copyTargetFields` specify which field values are actually copied and where to.

Example in `UPDATE_DEFAULT`: creating follow-up activities (`ExpandNew` action in `stepCreateFollowUpContact->UpdatePredecessorContact` of `processNewFollowUpAppointment`).

---

**Note:** If `copySourceFields` and `copyTargetFields` are specified but `copyUid` is not specified, the data is taken from the record specified in `link`.

---

## copySourceFields/copyTargetFields

Comma-separated lists of fields whose values are copied to the details view of the new record. `copySourceFields` contains the fields from the source record (`copyUid`), `copyTargetFields` contains the target fields.

## Syntax

`copySourceFields`: A comma-separated list of fields (`<field name or number>`) referring to the info area specified in `copyUid`) or the syntax `<infoAreaId>.<fieldId>` (example - `FI.City`) to copy fields from a parent info area.

`copyTargetFields`: A comma-separated list of fields that receive the values from `copySourceFields`.

## Examples

```
copySourceFields = City
```

Copies the value of the **City** field if `copyUid` refers to a related company record.

```
copySourceFields = FI.City
```

Copies the value of the **City** field from FI no matter whether `copyUid` (or `link`) refer to a related company record.

Creating a new person (KP) in a company (FI):

```
copySourceFields = !Country
```

```
copyTargetFields = FreeC1
```

This copies the text value of the **Country** catalog into the new person's **FreeC1** field.

---

**Note:** When using field IDs: To copy the text of a catalog value, add a "t" in front of the field ID (example - `FI.t5`).

---

---

**Note:** When using XML names: To copy the text of a catalog value, add a "!" in front of the field name (example - `FI.!Country`).

---

---

**Note:** To copy a catalog value from one catalog field to another catalog field (that uses the same catalog), enter just the field name or ID (example - `FI.Country`)

---

---

**Note:** Copying a catalog value to a different catalog is not supported.

---

## autoSave

If set to `true` the record is automatically saved after the Expand view has been created.

---

**Note:** The match-up is not executed if this option is enabled.

---

---

**Note:** You have to define default values for any mandatory fields, otherwise the record cannot be saved.

---

**Default:** `false`

## onCancel

Use the `onCancel` parameter to control a New page's behavior when the user cancels the creation of a new record (by clicking **Cancel** or pressing the ESC key).

Defining `onCancel = $back` (default) results in the following behavior:

- If `createNewPopup = true` is defined for the `ExpandNew` action, the popup closes. For further details, see [createNewPopup](#).
- If `createNewTab = true` is defined for the action, the tab is closed.
- If neither `createNewPopup` nor `createNewTab` is set to `true`, the last page of the current tab is loaded again.

You can use this parameter for `ExpandNew` pages in Tree views as well.

## Search

The Search action template allows users to search for records in an info area.

The `search` action template (`search.aspx`) allows users to search for records in an info area, either linked or not linked to a parent record. The search results are displayed in a list offering a context menu for each record. If configured, records can be edited directly in the list, and the page displays a sub-list of info areas that are dependent of the selected record.

The `Search` action template can be used for actions defined for menu entries, buttons and within processes. For further details, see [Defining an Action](#) on page 287. The contents and functionality of a Search page is configured via the Search&List configuration referenced in the action. For further details, see [Search&List Configuration](#) on page 365.

The screenshot displays the 'Search Companies' interface. At the top, there is a 'Search header' with search controls for 'Company', 'Country', 'From PC', 'To ZIP', and 'With me'. Below this is a 'Search Results' section with a 'SearchResults header' and a table of results. The table has columns for Company, Lead Status, Country, Postal Code, City, and Street. The first result is 'AkiComp' with details like 'Synonym Aki', 'City Wien', 'Country Austria', etc. Below the table is a 'Related Data' section with a 'SearchSubList header' and a list of related items like 'Persons', 'Activities', etc. The bottom part shows a table with columns for Last Name, First Name, Function, Department, Area Code, Tel 1, Company, and E-mail 1.

## Search&List Configuration

Learn about the Search&List Configuration in CRM.

The screenshot shows the 'Configuration for 'FI'' dialog box. It has a title bar 'Configuration for 'FI'' and a close button. The main area contains several fields: 'Infoarea' set to 'Company (FI)', 'Configuration' set to 'my\_config', 'Field Group' set to 'FI', 'Header Group' set to 'FI', 'Context Menu' set to 'M\_Company', and 'Default Action' set to 'not selected'. There is a 'Description' field with the placeholder text 'Enter description here.' and a 'Help ID' field. A 'Save' button is at the bottom right.

A Search&List configuration consists of the following items:

- **Field Group:** Defines the layout of the search mask and list rows. The Search and List controls of the specified field group are used. For further details, see [Search control](#) and [List control](#).
- **Header Group:** Defines the header group of the headers used on the page. For further details, see [Standard Headers](#) on page 337.

- **Context Menu:** Defines the context menu displayed for the search result rows. For further details, see [Context Menus](#) on page 292. The menu specified here is also used as a fallback for records displayed in sub-lists.
- **Default Action:** Defines the default action used for a double-click on a row in the search results list. For further details, see [Default Action](#) on page 289. If undefined, the **InfoArea Default Action** specified in the info area's default Expand configuration is used as a fallback.

---

**Note:** If used within a process, the context menu and the default action settings defined here are replaced by the defaults for processes: default context menu `M_Wizard` and default action `A_Select` configurable via the `Process.DefaultMenu` and `Process.DefaultAction` Web Configuration parameters. For further details, see [Process.DefaultMenu](#) and [Process.DefaultAction](#).

---

- **Description:** An internal description (only visible in CRM.Designer).
- **Help-ID:** Allows you to define a name which can be referenced in the `helpmap.xml` for the context-sensitive help (example - `<HelpID>Details</HelpID>`).


For linked searches a special logic is implemented for determining the Search&List configuration:

If a Search&List configuration named `xx` is called and info area `xx` is linked to the info area `yy`, the Search&List configuration is determined as follows: `xx-yy` (example - `MA-AK`), fallback `xx`.

If you explicitly enter a name (example - `MyMA` or `MA.Tree`), this Search&List configuration is used. However, if the specified Search&List configuration is not found, the `xx-yy` (fallback `xx`) logic is applied.

In the table caption for the search page the default table caption for the linked info area is used. For MiniDetails, the MiniDetails control of the default field group is used.

## Remember Search Settings

By default, the user's search settings are saved between sessions in the user's roaming settings. For further details, see [User Settings](#) on page 25. Users can reset the search settings for a specific search by clicking on  (**Settings**) > **Clear/Reset search settings**. For further details, see [Resetting a Search](#).

To globally turn off the "remember search settings" feature, uncheck the `Search.RememberDynamicFields` Web Configuration parameter. For further details, see [Search.RememberDynamicField](#). To override this setting for a specific search action, use the `rememberSearch` parameter. For further details, see [rememberSearch](#). To ensure that values set in the search mask are remembered enable the `Search.RememberSearchValues` web configuration parameter, see [Search.RememberSearchValues](#).

The application-wide saving of search settings is determined by the context of the search. The context is defined by the Search&List configuration and the assigned `flavor`, an optional action parameter. By default, only one flavor "Select" is available. It is automatically assigned to a Search&List configuration when called from within a process.

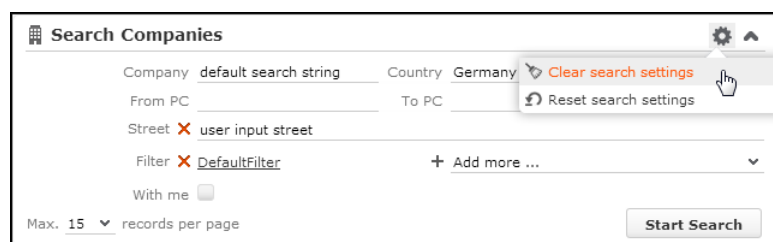
### Example:


The search settings for the person search called via `ID_NET_PERSON_SEARCH` are saved as "Search/KP".

The search setting for the person search called from within the **New Activity** process are saved as "Search/KP/Select".

You can assign any other flavor to a custom search action (by defining the `flavor` action parameter) to save the search settings for this action in a separate container.

## Resetting a Search



By default, the info area-specific searches (i.e. searches for which a Search&List configuration and/or `Search` action exists) can be reset by clicking on  (**Settings**) > **Clear/Reset search settings**:

**Note:** In addition to these options you can also provide **Clear Search** and **Reset Search** buttons beside the **Start Search** button by enabling the `Search.ShowClearSearchButton` and `Search.ShowResetSearchButton` web configuration parameters. These are global options and are applied across all info areas providing search capabilities. For more information, see [Search.ShowClearSearchButton](#) and [Search.ShowResetSearchButton](#).

Generic searches are searches without info area context. Example - Queries, Analyses, etc. They only offer the **Clear search settings** option since you can not specify predefined search settings like `defaultFilterName` etc. Therefore there is nothing to reset a generic search to.


The reset/clear options work as follows:

- **Clear search settings** (`A_ClearSearch`): Selecting this option **clears all search criteria** (default and user-defined search values and filters) from the search mask. Additional search fields are not removed but the search values are cleared.
- **Reset search settings** (`A_ResetSearch`): Selecting this option **restores the settings** as defined in the `Search` action (search fields, `defaultFilterName`, `defaultFieldValues`).

Additional search fields and filters are removed, search values are cleared or reset to `defaultFieldValues` if applicable.

In both cases fixed search criteria (`fixedFilterName` and `fixedFieldValues`) are retained.

The availability of the reset/clear options is controlled via Web Configuration parameters:


Enabling the `Search.ShowSettingsButton` Web Configuration parameter adds a `SearchSettings` button  to all info area-specific search pages and a `SearchSettingsGeneric` button to all other search pages. For further details, see [Search.ShowSettingsButton](#).

---

**Note:** The `SearchSettings` and `SearchSettingsGeneric` buttons are hard-coded; you can not remove such a button from a specific search since it is not visible in the header definition.

---

---

**Note:** If a search page's header already contains a , **Settings** button (i.e. a button with a name ending with `_Settings`), `SearchSettings` / `SearchSettingsGeneric` is not used.

---

The `SearchSettings` and `SearchSettingsGeneric` buttons call the menus `M_SearchSettings` and `M_SearchSettingsGeneric` respectively.

`M_SearchSettings` contains the menu actions `A_ClearSearch` and `A_ResetSearch`, `M_SearchSettingsGeneric` only contains `A_ResetSearch`.

Enabling/disabling the options of the `Search.ClearSearch` Web Configuration parameter determines the availability of `A_ClearSearch` and `A_ResetSearch`. For further details, see [Search.ClearSearch](#).

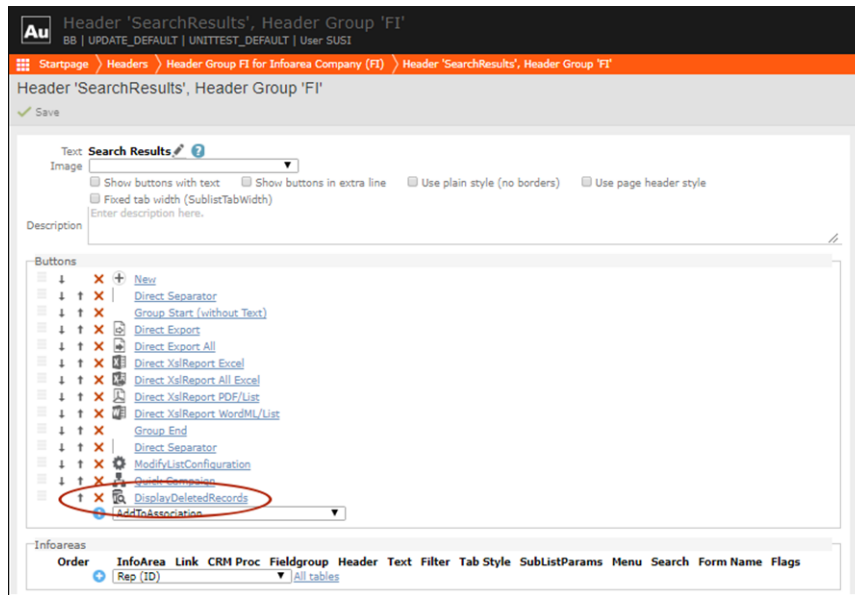
## Displaying Deleted Records

Deleted records are not displayed by default for all info areas.

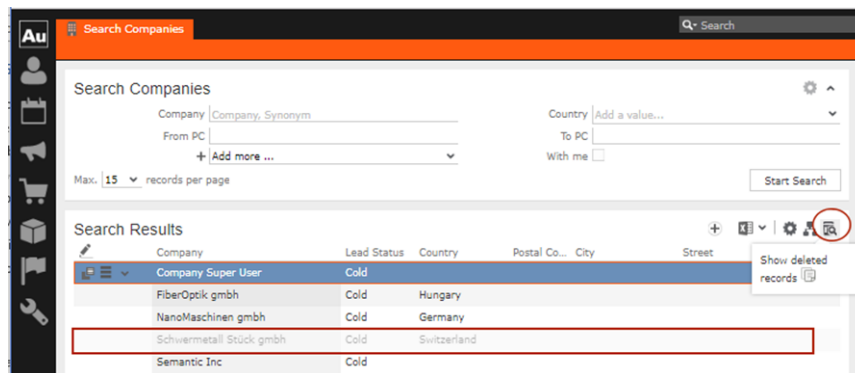
To allow users to view deleted records in a search list perform the following:

1. Ensure that the required info area records are marked as deleted. This ensures that the deleting a record from the info area is only marked as deleted in the database, but is not physically deleted from the database. For more information on this task, see **Rights Module > Rights > Mark as Deleted** in the *Aurea CRM Win Administration Guide* ([https://help.aurea.com/crm/crm-win-admin-guide/crm-win-admin-guide/dita/topics/c\\_win\\_adm-1\\_1-Intoduction\\_crm.html](https://help.aurea.com/crm/crm-win-admin-guide/crm-win-admin-guide/dita/topics/c_win_adm-1_1-Intoduction_crm.html)).
2. Add the **DisplayDeletedRecords** button to the search results header for the required info area. For information on adding and configuring an action button, see [Buttons](#) on page 304 **Navigation Customization > Buttons** in the *Aurea CRM Web Administration Guide* (<https://help.aurea.com/crm/crm-web-administrator-guide/index.html>). The following screenshot shows the button added to the **Search Results Header** group of the Company (FI) info area.





3. The Search results list pane for the configured info area displays the **DisplayDeletedRecords** button. Clicking the button displays the deleted records. The deleted records displayed are greyed out and cannot be viewed or edited. The following screenshot shows the button and the deleted record (**Schwermetall Stück gmbh**) visible in the result list.



## Search Action Parameters

Learn about the input arguments supported by the Search Action Parameters.

The `Search` action supports the following input arguments:

### fixedFilterName

Allows you to specify a fixed filter which is applied to the search and cannot be changed by the user. The filter name is not visible for the user.

---

**Note:** Filters have to be referenced as `<InfoAreaId>.<FilterName>` in the action definition (e. g. `KP.Filter1`).

---

## fixedFieldValues

Fixed field values for the search. This cannot be changed by the user.

**Syntax:** <fieldId>:<value>

- You can specify either field IDs or XML names as <fieldId>.
- Separate multiple items by semicolons.
- If <value> is a string, it must be enclosed in double quotes. Do **not** use double quotes when specifying catalog codes.

**Example(for a F search):** Company:"my search string";Country:#2;Rep2ID:"\$cur-Rep"

- The specified fields must be part of the respective Search control, otherwise they are ignored.
- To enter the current rep: <fieldId>:"\$curRep"
- To enter the current rep's group: <fieldId>:"\$curOrgGroup"
- To enter the current date: <fieldId>:"\$curDay"
- To enter the number of the current user's login language: <fieldId>:LOGINLANGUAGE
- To set a value for a catalog field: <fieldId>:#<catalogCode>

For hierarchical catalogs you can enter <fieldId>:#<childCode>~#<parentCode>

**Example:** For a **Ticket (KM)** Search action you can define

Problem:#3~#1

instead of

ProblemGroup:#1;Problem:#3

- To set a value for a variable catalog field using the external key: <field-Id>:EXT:<external key>

**Example:** ProblemGroup:EXT:1\_324\_2

- To set the value of a Boolean field enter `true` or `false`.
- If you want to enter a value from a linked record into a field in the search mask, use the syntax <targetFieldId>:<infoAreaId>.<fieldId>

**Example:** When searching for all mailings (BR) that were sent to a person (KP) and want to specify the mailing channel based on the person's preferred channel, enter `Channel:KP.PreferredChannel` in the Search action for mailings used in the person context menu.

- If you have a field from a linked info area in your search mask, you can automatically enter a value using <infoAreaId>.<fieldId>:<value>

**Example:** To search only for persons with specific interests, enter `IT.Interest-Grp:#2` in the person Search action

## searchAndListName

The name of the Search&List configuration to be used.

---

If undefined or not found, the Search&List configuration `<infoAreaId>` is used. This is the default Search&List configuration for the info area from which the action is called. For linked searches a special logic is implemented for determining the Search&List configuration. For further details, see [Search&List Configuration](#) on page 365.

### defaultFilterName

Allows you to specify a filter which is applied to the search by default. Users are still able to select another filter unless `disableFilter = true`.

---

**Note:** Filters have to be referenced as `<InfoAreaId>.<FilterName>` in the action (e. g. "KP.Filter1"). In Aurea CRM.Web they are then displayed as `<FilterName>` (e. g. "Filter1").

---

### createNewTab

See [createNewTab](#) in [Expand Action Parameters](#) on page 356.

### filterNameSpace

Defines a namespace for the filters available on the Search page.

To use an info area multiple times, it is common to apply a fixed filter when calling Search pages. If you do not want users to see all filters defined for an info area but only those applicable to the context of the current Search page, you can define a prefix in `filterNameSpace`. Users only see filters whose name begins with this prefix in the **Add more** drop-down list.

**Syntax:** `<InfoAreaId>.<Namespace>.<FilterName>`

**Example:** If you have two filters "FI.usage1.myFilter" and "FI.usage2.anotherFilter", and specify "usage1" as the `filterNameSpace` of a Search page, the user only sees "myFilter" in the **Add more** drop-down list of this Search page.

---

**Note:** If you work with filter namespaces, you should define `filterNameSpace` for all your Search pages, otherwise filters are displayed including the namespace.

---

### disableFilter

Disables the filter selection and definition functionality for the user. Both the predefined filters and the **More Filters** entry are removed from the **Add more** drop-down list of the Search page.

---

**Note:** To globally restrict working with filters to predefined filters only, create and check a Web Configuration parameter called `Search.NoFilterDefinition` (type: Checkbox): This removes the **More Filters** entry from the **Add more** drop-down list on all Search pages. For further details, see [Search.NoFilterDefinition](#). Users may only select predefined filters and can no longer create or edit any filters in Aurea CRM.Web.

---

## link/linkId


`link`: Optional parent record of the search results (for linked searches, example - all MA records linked to a KP): `Record or Linked`.

`linkId`: ID of the link used for `link` (default=-1 i.e. the default link). This parameter is only relevant if multiple links between two info areas exist in the data model (example - multiple FI-links in KM). For further details, see [Link IDs](#) on page 19.

## scripts

A comma-separated list of one or more (local) JavaScript files to be loaded to the page.

## showSearch

If set to `true`, the search mask is visible. If set to `false`, it is displayed collapsed and can be expanded by clicking .

**Default:** `true`

## relationName

Name of the relation to the parent/child records, for example `$Children` for hierarchical info areas like OJ in the BTB vertical.

Possible values:

- `$Parent`: Reads the "logical" parent record, example - the parent record of a sub-ticket is a ticket etc.
- `$ParentLink`: Reads the parent record as defined in the data model.
- `$Children`: Reads the children of a record, example - all MA records of a KP.

## selectionMode

Determines if users can select multiple rows from the list and thus perform action on multiple list items at once. The following values are available:

- `Single`: Selecting multiple list items is disabled.
- `Multi`: Selecting multiple list items is enabled (default).

**Default:** `Multi`

## listParams

Additional list parameters separated by semicolon: `key,value;key,value;` etc. For further details, see [ListParams](#) on page 374.

## autoSearch

If set to `true`, the search is immediately executed when the user opens the page.

Auto-search does not happen when the user updates a filter in the search mask. If you want auto-search to be activated when a filter is updated in the search mask, you have to set [autoApplyFilter](#) to `true`.

**Default** = `false`

### defaultFieldValues

Default field values for the search that can be changed by the user. Syntax. For further details, see [fixedFieldValues](#).

Syntax for entering ranges and functions: `<fieldId>:<value>:<operator>`

**Example:** `MA.Date:TODAY-7:>=;MA.Date:TODAY+14:<=`

To avoid that a default value is overwritten by the user's search settings (see [Remember Search Settings](#)), use the following syntax: `!<fieldId>:<value>`

**Example:** `!Street="Default Street"`

### defaultsUid

The uid of the record from which `defaultFieldValues` are read: `Record` or `Linked`. If undefined, the `link` argument is used as a fallback.

### maxRows

Determines the maximum number of results returned by the search. This parameter overrides the global value specified in the `Search.MaxResults` Web Configuration parameter. For further details, see [Search.MaxResults](#).

---

**Note:** The value specified here is displayed as default in the search mask's **Max. Records/Page** drop-down list.

---

### subListParams

Additional list parameters for sub-lists separated by semicolon: `key,value;key,value;` etc. For further details, see [ListParams](#) on page 374.

---

**Note:** The `subListParams` defined in the `Search` action override the corresponding **SubListParams** defined for individual sub-lists in `SearchSubList` headers. For further details, see [Related Data area \(Header Info Areas\)](#) on page 329.

---

### createNewPopup

See [createNewPopup](#).

### autoExpandOnSingleHit

If set to `true` and the search returns exactly one record, the search results list is not displayed but the default action is automatically executed for the record. For further details, see [Default Action](#) on page 289.

This setting overrides the global setting defined in the `Search.AutoExpandOnSingleHit` Web Configuration parameter. For further details, see [Search.AutoExpandOnSingleHit](#).

**Default** = `false`

### sourceCopyFieldGroupName

The name of the field group used to define a dynamic filter for the search. The filter values are defined as functions in the List control. For further details, see [Dynamic Filters](#) on page 509.

### flavor

Use this input argument to customize the behavior (example - default action, menu etc.) of a specific search view (example - in a process).

Available values:

- `Auto`
- `Select`: A search view with flavor `Auto` gets the flavor `Select` if used within a process or record selector. If used within a record selector, the buttons configured in the `SearchResults` header are hidden.
- `Search`: You can use this flavor to display the buttons configured in the `SearchResults` header when using the search view in a process or form.

Alternatively you can configure the buttons in the `SearchResultsSelect` header. In this case the buttons are available both in processes and record selectors that call this search view.

To make sure header buttons are displayed in a search view configured in a form, you can set **SearchView > resultHeader > view > showButtons** to true.

Default: `Auto`

### sourceUid

The record from which values are read for dynamic filters. For further details, see [Dynamic Filters](#) on page 509. If undefined, `link` is used as a fallback.

### rememberSearch

Add this argument and set it to `true` or `false` to override the global setting of the `Search.RememberDynamicFields` Web Configuration parameter. For further details, see [Remember Search Settings](#).

## ListParams

This topic has information on how to specify additional parameters to control the behavior of lists and sub-lists.

The `listParams` and `subListParams` arguments allow you to specify additional parameters to control the behavior of lists and sub-lists.

### Syntax:

---

<Name>, <Value>; <Name2>, <Value2>; etc.

### Example:

```
canEdit, false; cachedMode, true;
```

To use `listParams` with JavaScript in a form, use the syntax `oParameter.ListParams = "singleClickEdit, true";` (no spaces inside the quotes).

---

**Note:** If not stated otherwise, the parameters described in this chapter can be used for configuring both lists and sub-lists.

---

**Note:** Since lists are also used in contexts other than the `Search` action template, some parameters are only useful when applied to these other contexts.

---

Use the Web Configuration parameters `List.ListParams` and `List.SubList.ListParams` to define global settings applying to all lists/sub-lists. For further details, see [List.ListParams/List.SubList.ListParams](#).

The `ListParams` action supports the following input arguments:

#### **allowColumnReordering**

If set to `false`, users cannot change the order of list columns.

#### **allowColumnResizing**

If set to `false`, users cannot resize list columns.

#### **allowEditOnClick**

If set to `false`, users cannot switch to Edit mode by "slow double-click".

#### **allowSorting**

If set to `false`, users cannot sort the list.

#### **autoSelect**

If set to `true`, the first list row is automatically selected.

#### **autoApplyFilter**

If set to `true`, the search is immediately executed when the user updates a filter in the search mask. It is dependent upon on the [autoSearch](#) search action parameter and is active only when the `autoSearch` function is set to `true`.

This parameter can be applied to an action template and specific search masks. It can also be applied globally using the Web Configuration parameters `List.ListParams` to define global settings across all lists. For more information, see [List.ListParams/List.SubList.ListParams](#).

**Default** = false

### cachedMode

If set to `true`, lists are cached on the client. This allows results to be sorted by decoded fields (Z-fields). Example - the company or person link fields in the **Target Group (Marketing Activity)** info area.

In addition, since all results are transferred to the client, the total number of matches is displayed in the search results header (example - "1 - 10 of 54") thus overruling the `showCount` setting. For further details, see [showCount](#).

---

**Note:** Use `cachedMode, true` only if your users really need this kind of sorting which is not supported by the server. When a list is cached on the client, you cannot selectively export data since the server does not know about the client's sorting. In addition, with client-side caching enabled, **all** search results are transferred to the client at once. Thus, only activate this feature for info areas where you are sure that searches returns a limited number of results.

---

If a user changes the sorting of a list cached on the client, changes are not applied immediately. In that case the search needs to be started again.

In `UPDATE_DEFAULT` `cachedMode` is set to `true` for the **Target Group (Marketing Activity)** info area (AG) (`Search` action defined for menu `A_GotoAG`).

### canEdit

If set to `false`, records cannot be edited in a list. If set to `true`, the list is editable, also in processes.

**Default:** `true` for "standard" lists, `false` for lists within processes.

### collapseInvisibleRows

If set to `true`, the list height shrinks to its actual size depending on the number of result records. If set to `false`, the list keeps its size regardless of the actual number of displayed records.

### Example

A list containing two result rows and **max. Records per page** is set to 10:

- `collapseInvisibleRows, true` results in a list that is 2 rows high
- `collapseInvisibleRows, false` results in a list that is 10 rows high.

**Default:** `false`

### defaultAction

Use this parameter to execute another than the default action, when the user double-clicks a row. You can:

- specify the name of a menu action. Example - `A_ExpandTree`
- specify the name of a custom JavaScript function



---

## fieldgroupName

Defines the field group to be used.

---

**Note:** Using this parameter is only recommended for pages other than the `Search` action template.

---

## headerWidth

Determines the width of the list header. This overwrites default. You need to specify the measuring unit and a ";" at the end.

### Examples:

```
headerWidth,180px;
```

```
headerWidth,10cm;
```

```
headerWidth,40%;
```

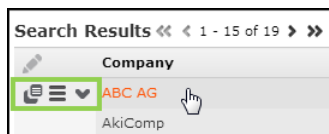
## hideCaption

If set to `true`, the text of the list header is hidden.

To hide all elements of a list header, arrow buttons, direct buttons and header text, enter `showPaging,false;showButtons,false;hideCaption,true`.

## hideControlColumns

If set to `true`, the first column of the list containing the Quick View, Context Menu and Minidetails buttons, or Save and Cancel when in Edit mode is hidden.



## includeConditionalRights

If set to `true`, conditional field rights are represented in the user interface, which means:

- Conditional Field Rights - **Deny Access:** If data is not accessible, the label for the field is displayed but the field value is locked.
- Conditional Field Rights - **Deny Update:** If data is not editable, the field is displayed as read only.

For lists, the display of conditional field rights is enabled by default.

---

**Note:** `includeConditionalRights` is set to `true` for lists on quick add pages by default and cannot be changed. For further details, see [SerialEntry](#) on page 454.

---

---

**Note:** Important information regarding `includeConditionalRights` and **Performance:**

- Conditional field rights affects the performance of list creation (because for each record in the list, an extra round-trip to the server is required). Thus, Aurea recommends disabling the display of conditional field rights in case you have unexpected performance degrade when loading large amount of data. To do so, add `includeConditionalRights, false` to the `listParams` argument of the Search action and/or to the corresponding header definition for sub-lists.
  - Conditional field rights are applied in any case, independent of their representation in the user interface using the above-mentioned parameter, i.e. if `includeConditionalRights` is set to `false`, the user cannot save field values in fields where conditional field rights apply.
  - To set the `includeConditionalRights` parameter, when you are using `SearchView` widget on a **Form**, go to “SearchView -> resultModel” to find and set the parameter.
- 

### initialTab

Determines which sub-list tab is active when a page is initially called. You can specify either the tab number (0-based) or the info area ID: `initialTab, 3` activates the fourth sub-list tab upon opening the page, `initialTab, MA` activates the **Activities** sub-list.

---

**Note:** This parameter is intended for search sub-lists only. For sub-lists in Expand views you need to use the `initialSubListTab` action parameter. For further details, see [initialSubListTab](#).

---

**Note:** It is recommended to specify the info area ID instead of the tab number since this continues to work even if the tab order is changed at some later point in the project.

---

### maxRows

Set `maxRows, <number>` to define the maximum number of records shown in a sub-list. For further details, see [Related Data area \(Header Info Areas\)](#) on page 329. If `maxRows` is defined in a header, the Web Configuration parameter `Search.MaxResults` does not have any effect. For further details, see [Search.MaxResults](#).

---

**Note:** This parameter is intended for sub-lists only.

---

### selectionMode

Determines if users can select multiple rows from the list and thus perform action on multiple list items at once. The following values are available:

- `Single`: Selecting multiple list items is disabled.
- `Multi`: Selecting multiple list items is enabled.

**Default:** `Multi`

---

---

## showButtons

If set to `false`, the buttons in the list header are not displayed.

## showCount

Allows the number of search results to be displayed. If more records than specified with `showCount` are found, the user is informed that more than `<limit>` results exist. Example - "Search Results 1 – 10 of > 100".

Specify `showCount, <limit>` (example - `showCount, 50`) in the `ListParams` of any desired list or sub-list or globally using the `List.ListParams/List.SubList.ListParams` Web Configuration parameters. For further details, see [List.ListParams/List.SubList.ListParams](#).

In `UPDATE_DEFAULT` `showCount` is globally set to "50" via the `List.ListParams` Web Configuration parameter.

---

**Note:** The maximum limit is `showCount, 100`. If a larger number is specified, 100 is used as the limit and a message is written to the client log.

---

---

**Note:** Performance considerations: To determine the number of search results, `<limit>` records are read for each search even if only, say, 10 records are displayed to the user. If you experience performance issues using this feature, try lowering the limit.


---

## showLinks

If set to `true`, link fields pointing to other info areas are displayed as hyperlinks in lists.


**Default:** `true`

## showMenu

If set to `true`, the **Display Menu** button  is displayed in sub-list rows calling the context menu defined in the Search&List configuration, fallback: **InfoArea Default-Menu** from the info area's default Expand configuration. If set to `false`, the **Display Menu** button is hidden, i.e. no context menu can be displayed for sub-list rows.

Specify `showMenu, <MenuName>` to display another than the default context menu (example - `showMenu, M_Person`).

## showMiniDetails

If set to `false`, the **Display Details** button  is hidden from the sub-list row, i.e. no MiniDetails can be displayed for sub-list rows.

## showPaging

Determines the display of number of records plus arrow buttons to browse through the list. By default, number of records plus arrow buttons are hidden. Available values:

- `showPaging, true`: The number of records plus arrow buttons are displayed above and below the list.
- `showPaging, top`: The paging is displayed only above the list.
- `showPaging, bottom`: The paging is displayed only below the list.

---

**Note:** If defined as `listParam`, only `true, false` and `bottom` are supported.

---

To show the paging for query snapshots:

You need to define `listparams = showPaging, true`; directly in the `QueryResultSetOpen` action used by `A_QueryResultSetRun` in `UPDATE_DEFAULT`. The `List.ListParams` Web Configuration parameter is not taken into account by the `QueryResultSetOpen` action.

## showQuickViewAction

If set to `false`, the **QuickView** button  is hidden from the sub-list row.

Define `showQuickViewAction, false` in the `List.SubList.ListParams` Web Configuration parameter to hide the **QuickView** button from the **Parent Data** list.

## showSumLine

If set to `true`, the sum total of the list is displayed immediately when opening a `Search` page.

**Condition for showing the sum total:** At least one result with at least one field that can be totaled is returned. The field has to be of data type S or L without the "no sum" property being set in the data model.

Values are summed per page. If a value is changed, the sum total is recalculated immediately.

In `UPDATE_DEFAULT` `showSumLine` is set to `true` for the **Offer** and **Order** info areas (menu actions `ID_NET_OFFER` and `ID_NET_ORDER_SEARCH`).

## singleClickEdit

If set to `true`, users can switch to the list's Edit mode with a single click instead of the default double click.

## virtualInfoArea.ReadMode

Determines whether virtual info areas are used for context menus of list rows (static | dynamic). For further details, see [virtualInfoAreaReadMode](#) on page 285.

## Tree

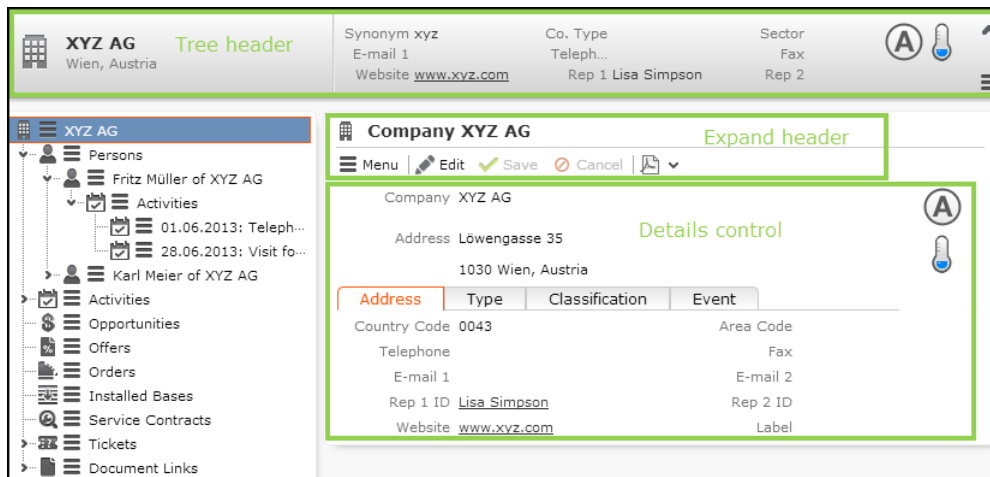
The Tree action template allows user to display a Tree view of a record and its children.

The `Tree` action template (`tree.aspx`) allows user to display a Tree view of a record and its children. The tree displays all existing sub-levels.

The `Tree` action template can be used for actions defined for menu entries, buttons and within processes. For further details, see [Defining an Action](#) on page 287.

Example in `UPDATE_DEFAULT`: `D_ExpandTree` in `M_Company`.

The contents and functionality of a Tree view is configured via the `TreeView` configuration referenced in the action. For further details, see [TreeView Configuration](#) on page 381.



The Tree view uses the Details control for displaying record details. For further details, see [Details control](#). Therefore the Tree view supports the same functionality as the Expand view of a record, example - displaying activity participants, sub-lists, filter masks, match-up for new XYZ records, exports, reports, etc.

**Note:** Filter masks in a Tree view do not support different CSS styles. For further details, see [Alternative Expand Configurations](#).

## TreeView Configuration

Defines the nodes of a tree.

The `TreeView` configuration defines the nodes of the tree, which sibling or child info areas are displayed in the tree view of a record.

TreeView 'FI.Report'\*

✓ Save ✗ Delete ⓘ Recent Changes

Description

+ Add ◀ Decrease Indent ▶ Increase Indent

**Company (FI)**

- Person in Company (KP)
- Activity (MA)
- Activity (MA)**
- Opportunity (Y1)
- Order (AU)
- Offer (PR)
- Offer Item (AP)
- Ticket (KM)
- Document link (D3)
- Document (D1)
- Sales Group (SB)

**Info Area Settings**

\*Info Area **Activity (MA)**  show linked  show all

Link ID

Relation Name

CRM Processes **all** 🔍

Info Area Label

Record Filter

Info Area Filter

Sorting **d2**

Record Count **10**

Node Menu **M\_TreeChildren**

Search&List **MA.Tree**

Widget Options

Form

**Node Settings**

Check Children  Company Related  Disable Tweaks

Expand  Readonly  Expand Records

Hide Records  No Records  Hide Groupnode

Show Count  Hide Groupnode Without Children

**Record Settings**

Context Menu **M\_Appointments**

Expand **MATree**

Table Caption **MA.Report**

Widget Options

Form

To define a Tree view (see [Designing Configuration Units](#) on page 64):


1. Click **+ Add** to add a new node (i.e. sub-info area) to the tree.
2. Select the desired info area under **Info Area Settings > Info Area**.
  - **show linked:** Lists all info areas linked to the new node's current parent info area.
  - **show all:** Lists all available info areas.

Enable the **Link ID** check box and select/enter a link ID, if you want to use another than the default link. For further details, see [Specifying Link IDs](#) on page 20.

3. Click on an info area to define settings for this node. All available settings for the selected node are displayed in the right part of the page:

- general settings for the selected node. For further details, see [Info Area Settings](#).
- settings configuring the appearance and behavior of the node. For further details, see [Node Settings](#).
- settings for the records listed for the node. For further details, see [Record Settings](#).

4. Arrange the sub-info areas of your tree:

- Use the **Decrease/Increase Indent** buttons to move nodes to another hierarchy level, i.e. change the parent info area. Example: Create a node for the **Activity** info area below the **Person in Company** info area to display person-related activity records.
- To reorder nodes, click  (**Drag&Drop**) in the respective row and drag it to the desired position. Child nodes of the selected node are automatically moved as well.

While dragging the selected node, valid drop locations are displayed in blue. Nodes can only be dragged to the same hierarchy level. To change the indentation, use the **Decrease/Increase Indent** buttons as described above.

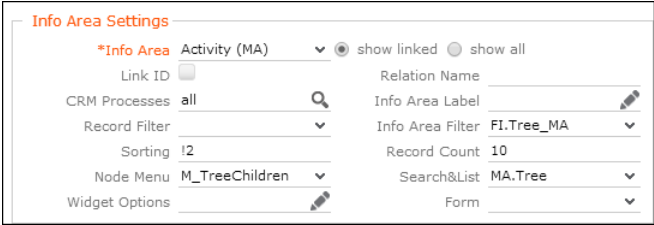
The topmost node (i.e. the tree's parent) is displayed in bold and cannot be moved or deleted.

- Click  (**Delete**) to delete a node. Child nodes are deleted as well.

## Note:

- If a sub-info area has no valid link to its parent, a message is displayed on mouse-over and the info area ID is displayed with a wavy red underline. You can still save the definition. Drag the info area to a valid position or create the required link in the data model.
- For sub-info areas with no valid link to their parent node the tree displays all available records in Aurea CRM.Web.

## Info Area Settings



**Info Area Settings**

\*Info Area: Activity (MA)  show linked  show all

Link ID:

Relation Name:

CRM Processes: all

Info Area Label:

Record Filter:

Info Area Filter: FI.Tree\_MA

Sorting: !2

Record Count: 10

Node Menu: M\_TreeChildren

Search&List: MA.Tree

Widget Options:

Form:

The general definition of a tree node (i.e. sub-info area) consists of the following settings:

- **Info Area:** Select the info area you want to define as a tree node. For further details, see [TreeView Configuration](#) on page 381.

With the tree view you can also navigate up and down hierarchies. For further details, see [walkUpToRootRecord](#). Hierarchy trees are implemented for the info areas A1, OJ, KM, IB, and MA. Select "Special: Parent" as child info area to navigate up the hierarchy, and "Special: Children" to navigate down. The configured table caption and menu settings for this node are ignored for the "Special" settings, they are taken from the parent instead.

- **Link ID:** Enable the **Link ID** check box and enter the link ID, if you want to use another than the default link. For further details, see [Link IDs](#) on page 19.
- **Relation Name:** If you have defined a custom link between two info areas, you can enter the name of this virtual relation, which is then displayed in the tree.
- **CRM Processes:** Assign one or more CRM processes to the tree node. For further details, see [Roles & CRM Processes](#) on page 727.
- **Info Area Label:** You can specify a label for the node (default: the plural of info area's name).



Table captions for the records listed in the tree are defined in the **Record Settings** area. For further details, see [Record Settings](#).

- **Record Filter:** Select a filter to filter the records displayed under this node.
- **Info Area Filter:** Select a filter to hide tree nodes based on field values from the parent record. Example: Define and select a filter to only show the Activity (MA) node if the parent company meets the condition `(RepID='$curRep' OR Rep2ID='$curRep') AND ABC='A'`.



All filters defined for the parent info area are listed.

---

**Note:** Filters with more than one info area and parameterized filters (i.e. filters requiring user input) are not supported. If such a filter is used, it is ignored and a message is written to the client log.

---



---

**Note:** If the value of a criteria field is changed by the user, he has to manually refresh the page in order to see the refreshed tree.

---

- **Sorting:** Defines the sort order for the records in the node. Use the following syntax:
  - Sort ascending: `<field name or number>` example - `Date` for the **Starts on** field of an activity (MA)
  - Sort descending: `!<field name or number>` example - `!Date` for the **Starts on** field of an activity (MA)
  - Comma-separate multiple sort fields: example - `!Date,Contact`
- **Record Count:** Defines the number of records displayed for the node, i.e. how many records are displayed when initially expanding an info area node. If more records exist, an additional node "..." is shown below. Clicking on "..." displays the next `<Record Count>` records.
- **Node Menu:** Defines the info area node's context menu. The following menus and menu actions are available:
  - `M_TreeChildren`: Contains all three menu actions, `A_ShowALL`, `A_TreeList` and `A_TreeNew`.
  - `M_TreeChildrenNoNew`: Contains `A_ShowAll` and `A_TreeList`.
  - `M_TreeChildrenNewOrEdit`: Contains `A_TreeNewOrEdit`. This is used for 1:1 children analogous to an Expand view with `mode = NewOrShow`. For further details, see [mode](#). Example in `UPDATE_DEFAULT: AD` node in `TreeView AK`.
  - `A_ShowAll`: Expands the info area node, thus displaying all child records.
  - `A_TreeList`: Leaves the tree view to display the node's child records on a Search&List page.
  - `A_TreeNew`: Allows users to add new records as children of that node linked to the parent record without leaving the tree view. The `SearchResults` header contains a `NewCommand` button doing the same. For further details, see [NewCommand](#) on page 317.
- **Search&List:** Defines the Search&List configuration to be used when clicking on the info area node. For further details, see [Search&List Configuration](#) on page 365. By default, the search area is collapsed, only the actual results are visible. Clicking a list item displays the corresponding sub-list.
- **Widget Options:** Use this field to pass additional parameters to the displayed widget (`SearchView` or the form specified under **Form**).

All properties for the selected widget as available in the Form Designer can be used. Define the widget with all desired properties in the Form Designer and copy the properties from the source view to this field.

## Examples:

```
{ "viewOptions": { "list": { "grid": { "showMenu": true, "showMiniDetails": false
  }}} }
{ "form": { "useAutoSearch": false }, "showSearchForm": true }
```

- **Form:** Allows you to provide the name of a custom widget or form to be displayed when clicking on the node (with `Record` representing the current record).


## Node Settings

This setting is for appearance and behavior of a tree node.

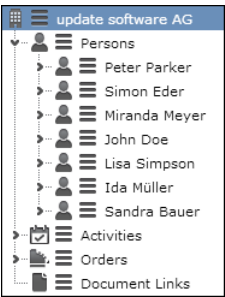
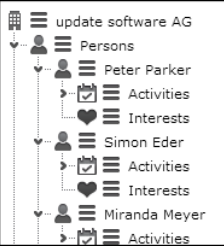
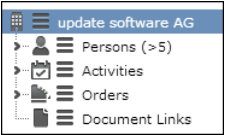
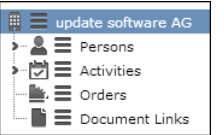

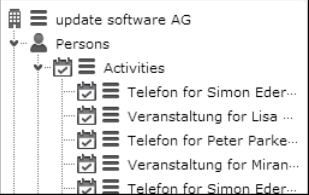
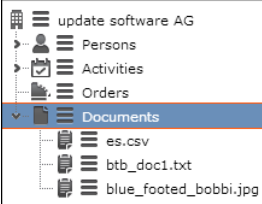


Node Settings		
<input checked="" type="checkbox"/> Check Children	<input type="checkbox"/> Company Related	<input type="checkbox"/> Disable Tweaks
<input type="checkbox"/> Expand	<input type="checkbox"/> Readonly	<input type="checkbox"/> Expand Records
<input type="checkbox"/> Hide Records	<input type="checkbox"/> No Records	<input type="checkbox"/> Hide Groupnode
<input type="checkbox"/> Show Count	<input checked="" type="checkbox"/> Hide Groupnode Without Children	

Use the following settings to configure the appearance and behavior of a tree node:

- **Check Children:** The  symbol for expanding a node is only displayed if the branch actually contains records. (To gather this information, the tree view tries to load the first record of the branch).
- **Company-related:** Only company-related records are displayed for this node. For example, use this flag to show only activities linked directly to the company instead of appointments linked to persons of that company.
- **Disable Tweaks:** This flag is only relevant when listing activities (MA) for a company or person: If checked, only MA records linked directly to the parent record are displayed but activities linked via the **Activity Participants** info area (MB) are ignored.
- **Expand:** When the tree is loaded, all branches with this flag are automatically expanded (i.e. `<Record Count>` records are read and displayed).
- **Expand Records:** When the tree is loaded, all branches with this flag plus the records within are automatically expanded (i.e. their sub-info areas are displayed).
- **Hide Group Node:** The group node is hidden, and records contained in this node are displayed directly under the parent node.
- **Hide Records:** The records of this node are not displayed.
- **No Records:** Allows you to define group nodes that contain group nodes themselves it is recommended to assign no menu if this flag is set.
- **Readonly:** Defines whether the details mask of a tree's record is editable or not. For further details, see [viewMode](#).
- **Show Count:** Displays the number of records for that node. If more than `<Record Count>` records are found, the user is only informed that there are more than `<Record Count>` records.
- **Hide GroupNode Without Children:** Hides this group node in the tree view, if the group node does not have any child record to display. For example, a Ticket, Sales Group, or Activity node is hidden if it does not have any records to display.

## Examples

Flag	Result	Flag	Result
<p><b>Expand</b> for persons</p>		<p><b>Expand Records</b> for persons</p>	
<p><b>Show Count</b> for persons</p>		<p><b>Check Children</b> example - for orders</p>	
<p><b>Hide Groupnode</b> for persons</p>		<p><b>No Records</b> for persons</p>	
<p><b>Hide Records and Hide Groupnode</b></p> <p>Use this combination to show documents linked to a record via D3 as if they were directly associated with the record.</p> <p>Example in UPDATE_DEFAULT: Treeview FI</p> <p>D3: Parent: FI   Link-ID: 127   Flags: <b>Hide Records</b> D1: Parent: D3   Flags: <b>Hide Groupnode</b></p> <p>You can define an <b>InfoArea Label Text</b> for the D3 node renaming <b>Document Link</b> example - to "Documents".</p>			

## Record Settings

Record Settings	
Context Menu	M_Person
Table Caption	KP
Widget Options	
Expand	KPTree
Form	

Use the following settings to configure the appearance and behavior of the records listed under the selected node:

- **Context Menu:** Defines the menu displayed for each record of this info area. You can use the same menu as specified in the Expand configuration for that info area. Alternatively, you can use `M_DefaultTree` or define a special menu e.g. to avoid leaving the tree view. If undefined, the user can only view or edit a record depending on the node settings. For further details, see [Node Settings](#).
- **Expand:** Defines which Expand configuration is used for displaying a record. If undefined, the default Expand configuration `<infoAreaId>` is used.
- **Table Caption:** Select the table caption to be used for all records of this info area and this branch (**Default** = `<InfoAreaID>`).

You can use any table caption defined for the info area – this allows you to use another than the default table caption, e.g. if the default table caption is too long or contains redundant information already visible in the tree like the name of the company in a KP table caption.

- **Widget Options:** Use this field to pass additional parameters to the displayed widget, `ExpandView` or the form specified under **Form**.

All properties for the selected widget as available in the Form Designer can be used. (Hint: Define the widget with all desired properties in the Form Designer and copy the properties from the source view to this field.)

Example: `{ "viewMode": "Edit" }`

- **Form:** Allows you to provide the name of a custom widget or form to be displayed when clicking on the node (with `$Record` representing the current record). The **Form** parameter allows to configure the tree view e.g. to open an external website in the area next to the tree.

## Tree Action Parameters

Learn about the Tree Action Parameters.

To call a Tree view from a context menu or a button, define the `Tree` action with its `treeViewName` and the record used as root (`uid`). If no TreeView configuration is specified, the default TreeView configuration `<infoAreaId>` is used.

If a TreeView configuration belongs to an info area other than the info area of the record passed to the tree, not the tree root but the record passed to the tree is automatically highlighted and expanded if configured that way. This allows you e.g. to call a company tree view configuration from the person context menu, and have the tree expanded so that the person is highlighted within the tree.

The `Tree` action supports the following input arguments:

## treeViewName

The name of the TreeView configuration. For further details, see [TreeView Configuration](#) on page 381.

If undefined, the TreeView configuration `<infoAreaId>` is used. The TreeView configuration for the info area from which the action is called.

You can use the prefix `%%` as a placeholder for the info area: Thus, e.g. instead of defining different actions for `IBHierarchyTree` and `KMHierarchyTree`, you can define one action (`A_HierarchyTree` in `UPDATE_DEFAULT`) specifying `%%HierarchyTree` as `treeViewName`.

## uid

The record to be displayed as the tree's root node.

This parameter is mandatory.

**Record:** The current record is displayed as the tree's root node.

**Linked:** The parent record of the current record is displayed as the tree's root node.

## link

Record to which the tree's root record is linked (`Record`, `Linked` or undefined).

Defines the link to the parent record, only used if the specified record acts as a link record to the tree's root record.

## linkId

The ID of the link used for `link`. This parameter is only relevant if multiple data model links between two info areas exist (example - multiple FI-links in KM). For further details, see [Link IDs](#) on page 19.

**Default**=-1, which corresponds to the default link

## headerGroupName

Defines the header group for the `Tree` header.

You can use the prefix `%%` as a placeholder for the info area: Thus, for example instead of defining different actions using `IBHierarchyHeader` and `KMHierarchyHeader`, you can define one action (`A_HierarchyTree` in `UPDATE_DEFAULT`) specifying `%%HierarchyHeader` as `headerGroupName`.

The `Expand` headers for the tree's sub-info areas are taken from the header group specified in the sub-info areas' `Expand` configurations. For further details, see [Record Settings](#).

## viewMode

The initial view mode of the `Expand` views accessed from the tree. Available values: `Edit`, `Read`.

This parameter overrides the standard behavior, i.e. the Expand view of tree node records being displayed read-only or editable according to the TreeView configuration (`ReadOnly` flag). For further details, see [Node Settings](#).

If `viewMode=Read`, all details masks are displayed read-only, regardless of the TreeView configuration. However, if the `Expand` header contains an **Edit** button, the user can manually switch to Edit mode if the tree view configuration does not also define the details as read-only.

**Default:** Read

### **expandOptions**

Determines the active Details control and sub-list tabs when initially displaying a record in Tree view.

#### **Syntax:**

```
<Name>, <Value>; <Name2>, <Value2>;
```

The following settings are available:

- `initialTab, <0-based number>`: Determines the initially active Details control tab.
- `initialSubListTab, <0-based number or info area ID>`: Determines the initially active sub-list tab.

---

**Note:** It is recommended to use the info area ID for `initialSubListTab` since this continues to work even if the tab order is changed at some later point in the project.

---

#### **Examples:**

- `initialTab, 2; initialSubListTab, 3`; initially activates the 3rd tab of the Details control and the 4th sub-list tab.
- `initialTab, 2; initialSubListTab, MA`; initially activates the 3rd tab of the Details control and the (first) **Activities** sub-list.

### **walkUpToRootRecord**

If set to `1`, the tree is defined as a hierarchy tree to display e.g. a ticket with its sub-tickets: The root record of the tree is determined from the chain of parent records starting at the record defined by `uid`.

Hierarchy trees are supported for the following info areas: A1, OJ, KM, IB, and MA.

---

**Note:** If an item references itself, it is only displayed once in the hierarchy.

---

### **createNewTab**

See [createNewTab](#) in [Expand Action Parameters](#) on page 356.

### **createNewPopup**

See [createNewPopup](#).

## history

The page is not added to the page-related browsing history. This parameter cannot be changed.

## relationName

See [relationName](#).

## JavaScript

Use the `JavaScript` action template to implement custom functions on the client (buttons, menu actions, process steps).

The JavaScript action template is a virtual page not based on a physical file. The following JavaScript action templates are available:

- `JavaScript`: Calls a JavaScript function.
- `JavaScriptWithCallback`: Calls a JavaScript function. On completion the callback is called.

For information on the supported input arguments, see the article “How to Use the Actions Service” at <https://support.aurea.com>.

In addition, the `JavaScript` and `JavaScriptWithCallback` action templates support the `recordSet` argument, see the article “How to execute JavaScript on a set of records” at <https://support.aurea.com>.

For information on loading scripts into Aurea CRM web at startup, see the article “Scripts” at <https://support.aurea.com>.

## Out-of-the-Box JavaScript Functions

Learn about the Out-of-the-Box JavaScript Functions.

Following are the list of Out-of-the-Box JavaScript Functions:

### **u8.services.businessObjects.copy**

Use this JavaScript function to copy information between info areas. In `UPDATE_DEFAULT` this function is used in `D_CopyOfferToOrder` (BTB and OTC), `A_CopyOfferToWR` and `A_CopyOfferFromServiceContract` both BTB only. For further details, see [D\\_CopyOfferToOrder](#).

Use the **Copy Fields** options in the **Configuration** info area (MC) to determine which fields are copied, see [Configuration Info Area](#) in the *CRM.Core Administrator Guide*.

### **u8.services.lru.clear();**

Use this JavaScript function to delete Aurea CRM.Web's browsing history. an example of doing this is by defining an application menu entry.

`u8.services.lru.clear();` deletes the complete history (pages and records).

`u8.services.lru.clear(type)`; deletes the history for the type specified:

- `u8.services.lru.clear("Page")`; for deleting the pages
- `u8.services.lru.clear("Record")`; for deleting the records from the browsing history

### **u8.session.identity**

Use the following functions to identify the current rep (example - for render hooks, process metrics etc.):

- `u8.session.identity.repId` gets the current user's **Rep ID** as an integer value.
- `u8.session.identity.repName` gets the current user's **Rep** name (e.g. "John Doe").
- `u8.session.identity.userName` gets the current user's login name (e.g. "JOHN").
- `u8.session.identity.isSuperUser` is true if the current user is the SU.

## **Reloading Custom Javascripts in the CRM Client**

Learn how to reload custom javascripts from cache in CRM client.

CRM client javascript files are cached in the browser until the CRM application is upgraded to the next version. You can also ensure that your custom CRM Client javascript files are cached in user browser only for the duration of the IIS application pool lifecycle and are forcefully reloaded upon IIS application pool recycle. This is an optional feature and is disabled by default.

To enable this feature enable the `<JavaScriptVersioning>` parameter in the `settings.xml` file, see the sample configuration below:

```
<update.web>
  <JavaScriptVersioning>ApplicationVersion</ JavaScriptVersioning >
</update.web>
```

`ApplicationVersion (default)`—ensures that the javascript files are reloaded when their version number changes. Use this option if you do not change the custom javascript frequently.

---

**Note:** The javascript files have the version number in the `ApplicationVersion` tag. For example, `ApplicationVersion: script.js?v10.2.0`.

---

`ApplicationStartTime`—The start time is set to the time when the application pool is reset. Use this option if you often change your custom javascripts.

---

**Note:** The javascript files have the start time in the `ApplicationStartTime` tag. For example, `ApplicationStartTime: script.js?123456789`.

---



## ShowForm

ShowForm action template is used to display custom forms.

The `ShowForm` action template (`form.aspx`) allows you to display custom forms. For further details, see [Forms](#) on page 621.

## ShowForm Action Parameters

List of input arguments for ShowForm Action Parameters.

The `ShowForm` action supports the following input arguments:

### **createNewTab**

See [createNewTab](#) in [Expand Action Parameters](#) on page 356.

### **createNewPopup**

See [createNewPopup](#).

### **FormName**

The name of the form you want to display. For further details, see [Forms](#) on page 621.

This parameter is mandatory.

### **HeaderName**

An optional header for the form. You can specify any special header of the type specified in `HeaderType`.

### **HeaderType**

The type of the header specified in `HeaderName`.

Enter the name of a header group or "Special" for an info area-independent special header.

### **InfoArea / RecID**

The `uid` of a record (`Record` or `Linked`). For further details, see [uid](#).

### **Options**

Option string for JavaScript evaluation.

### **scripts**

A comma-separated list of one or more (local) JavaScript files to be loaded to the page.

## Textgroups

Additional text groups (comma-separated).

## pageTitle

The title of the page. This page title is only used if `HeaderName` is undefined. For further details, see [HeaderName](#). The `pageTitle` action parameter overrides the `pageTitle` attribute defined in the form.

For more information, see [Defining a Header/Title for a Form](#) on page 630.

## WaitUntilCreated

If set to `true`, an empty page is display until the form is entirely created and all data loaded. If set to `false`, the form is immediately displayed and the loading process is visible for the user.

Define `WaitUntilCreated = false` for complex forms with lots of widgets and data to load.

For best user experience, complex widgets contained in e.g. a `DashboardPanel` should have a custom width set via CSS properties; this prevents a 'jumpy' loading process.

**Default:** `true`

## AnalysisEdit

The `AnalysisEdit` action template (`analysis.aspx`) is used to create, edit and execute analyses. Example in `UPDATE_DEFAULT`: The menu action `ID_NET_MAN-AGESTATISTICS (New Analysis)` in `ID_NET_QUERIES_AND_STATISTICS`.

**New:Analysis \***

✓ Save ✗ Delete ▶▶ Run Analysis

Analysis name   Private  Public

Query name  🔍

Categories, Values, Filter | List

Categories		Hide unused		Values		Hide unused	
	Disable ...	Show all...			Currency-dep...	Percentage-d...	
<input type="checkbox"/> Company	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Company	Count		
<input type="checkbox"/> Synonym	<input type="checkbox"/>	<input type="checkbox"/>					
<input type="checkbox"/> Country	<input type="checkbox"/>	<input type="checkbox"/>					

**Filter**  Hide unused

	Filter must be selected
<input type="checkbox"/> Company	<input type="checkbox"/>
<input type="checkbox"/> Synonym	<input type="checkbox"/>
<input type="checkbox"/> Country	<input type="checkbox"/>

**Default Values**

Maximum number of bars  Default for category

Currency field  Percentage field

▶▶ Run Analysis

## AnalysisEdit Action Parameters

List of input arguments for AnalysisEdit Action Parameters.

The `AnalysisEdit` action supports the following input arguments (all optional):

### **name**

The name of the analysis.

### **scope**

Defines which kind of analysis is to be used (if two analyses with the same name exist): `Public` or `Private`.

### **autostart**

If set to `true`, the analysis is automatically executed.

### **link/linkId**

Defines the context for the analysis' execution: The uid of a record to be used as link record when executing the analysis.

Use `linkId` (optional) to specify another than the default link.

### **autoLoadSettings**

Determines if the analysis settings are stored and loaded when executing the analysis and how it is done.

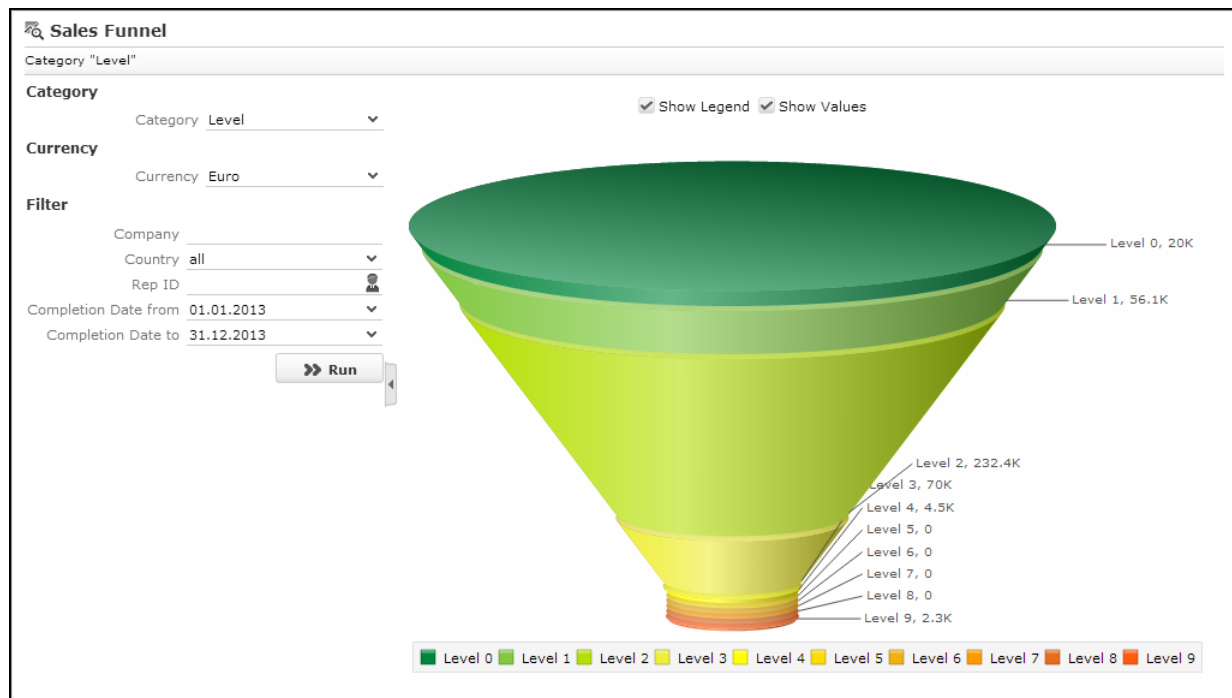
Available values: `"all"`, `"defaultValues"`, `"category"`, `"fields"`. Separate multiple entries by comma.

If defined, a copy of the analysis containing the most recent settings selected in the settings editor is saved. When the analysis is executed later, these settings is re-stored.

## AnalysisRun

The `AnalysisRun` action template (`analysisrun.aspx`) executes a predefined analysis.

Example in `UPDATE_DEFAULT`: The menu action `ID_SALESFUNNEL` in `ID_NET_PROCESSES` displays the **Sales Funnel** analysis.



## AnalysisRun Action Parameters

List of input arguments for AnalysisRun Action Parameters.

The `AnalysisRun` action supports the following input arguments:

### name

The name of the analysis to be executed.

This parameter is mandatory.

### scope

Defines which kind of analysis is to be used if two analyses with the same name exist: `Public` OR `Private`.

### autoStart

If set to `true`, the analysis is automatically executed.

### chartType

Specify the name of a chart type (e.g. `Bar3D`) determining the default chart type selection on the Analysis page. It can be changed by the user. For further details, see [Chart Types](#) on page 545.

### fixedChartType

Determines the chart type to be displayed. It cannot be changed by the user. For further details, see [Chart Types](#) on page 545.

**link/linkId**

Defines the context for the analysis' execution: The uid of a record to be used as link record when executing the analysis.

Use `linkId` (optional) to specify another than the default link.

**headerName**

Name of an optional special header to be displayed.

**showGrid**

If set to `false`, the **Show Table** button is hidden from the Analysis page.

**showChart**

If set to `false`, the **Show Chart** button is hidden from the Analysis page.

**showSettings**

If set to `true`, the settings editor is displayed on the Analysis page allowing users to modify the analysis. Use `showCategory`, `showValues`, `showCurrency` and `showFilters`. Ssee below sections to show/hide single elements of the settings editor.

`showSettings` overrules `collapseSettingsEditor`. For further details, see [collapseSettingsEditor](#).

**view**

Determines the initial view of the Analysis page: `grid` or `chart`.

**showCategory**

If set to `false`, the **Category** area is hidden from the settings editor.

**showValues**

If set to `false`, the **Values** area is hidden from the settings editor.

**showCurrency**

If set to `false`, the **Currency** area is hidden from the settings editor.

**showFilters**

If set to `false`, the **Filter** area is hidden from the settings editor.

**showChartLegend**

If set to `true`, the **Show Legend** check box is checked by default.

### **showChartValues**

If set to `true`, the **Show Values** check box is checked by default.

### **hideSelector**

If set to `false`, the chart selector is hidden from the analysis page.

### **collapseSettingsEditor**

Defines the initial state of the settings editor: If set to `true`, the settings editor is initially hidden.

`showSettings` overrides `collapseSettingsEditor`. For further details, see [showSettings](#).

**Default:** `false`.

### **pageSize**

Use/add this parameter to determine the number of result rows displayed in the table view of an analysis result page.

**Default:** `10`.

## **AnalysisSelect / QuerySelect**

The `AnalysisSelect` and `QuerySelect` action templates are used to select and execute analyses and queries respectively.

Examples in `UPDATE_DEFAULT`: `ID_NET_QUERIES_AND_STATISTICS` contains the menu action `ID_NET_SELECTANALYSIS` and `ID_NET_QUERY_FORMAT` allowing users to search for, select and execute analyses and queries.

The `AnalysisSelect` and `QuerySelect` actions support the following input arguments:

- an optional link record, see [link/linkId](#).
- Unlike for other `Search` actions, `autoSearch` is enabled for the Query and Analysis searches. Set `autoSearch` to `false` to disable the automatic search.

---

**Note:** Auto-search does not happen when the user updates a filter in an Analysis or Query. If you want auto-search to be activated when the user updates a filter, you have to set [autoApplyFilter](#) to `true`.

---

# BuyingCenter

The `BuyingCenter` action template (`BuyingCenter.aspx`) is used to create a "Buying Center Wall" for opportunities providing a graphical overview of all opportunity participants, their roles and their attitude.

For general information on opportunity participants, see [Opportunities](#) in the *Web Business Logic User Guide*.

The screenshot displays the BuyingCenter interface for an opportunity named "Opportunity Easter Sales". The interface is organized into several sections:

- Header:** Shows "Level 3", "Realization Prob... 100,00%", "100.000,00 Euro", and "Open". The responsible person is "Sally Seller".
- Attitude Columns:** A horizontal bar at the top categorizes participants by attitude: Hostile, Indifferent, Neutral, Friendly, and Superfriendly.
- Participant Cards:** Sticky notes representing participants are placed on the wall. For example:
  - Mr. Alex Alpha (Indifferent):** Consultant. "Not interested in phone contact or product samples".
  - Mr. Ronny Review (Indifferent):** Consultant. "Pure business consulting, shows no preference towards any vendor".
  - ACME (Neutral):** "Need to find the right contact person there!".
  - Ms. Miranda Meyer (Neutral):** Contact Person. "Sent sweets, hope that helps".
  - Chris Chros (Neutral):** End Customer. "Needs some more convincing...".
  - Ms. Irmgard Sabet (Friendly):** Influencer.
  - Ms. Anda Allner (Superfriendly):** Decider. "Loves our products Regular feedback loop".
  - Mr. Joe Bam (Superfriendly):** Influencer. "Related to one of our board members".
- Left Sidebar:**
  - New Participant:** A button to add new participants.
  - Suggested Participants (from activities):** A list of suggested participants including "Ac Ipsum Phasellus Industries", "Ms. Anne Anninger", "John Doe", and "Mr. Jasenko Ramiak".
  - Missing required roles:** "2 Influencers of 4 (2 to go)".
  - Missing optional roles:** "2 Consultants of 3 (1 to go)".
- Bottom Section (Related Data):**
  - Opportunity Competitor:** A table with columns for Opportunity Competitor, Strength, Weakness, and Product. Data includes "Huge Food Supplier" with Strength "Cheap" and Weakness "Mediocre Quality".
  - Activities:** A table with columns for Type, Starts on, Person, and Subject. Data includes "Visit" on "02.03.2015" by "Ms. Irmgard Sabet" (Subject: "Easter Bunny Tasting"), "Letter" on "23.02.2015" by "Ms. Miranda Meyer" (Subject: "Chocolate Samples"), and "Visit" on "14.01.2015" by "Mr. Jasenko Ramiak" (Subject: "Check-up").

In `UPDATE_DEFAULT` the `BuyingCenter` action is called by the `A_BuyingCenterY1` context menu action (**Buying Center**) configured in `M_Opportunity`.

All **Opportunity Participant** (Y2) records currently linked to the opportunity are displayed as sticky notes on the "wall". The available columns and sorting are defined in the action, see [BuyingCenter Action Parameters](#) on page 401. The (collapsible) related data area at the bottom of the page displays further records linked to the opportunity (competitors and activities in `UPDATE_DEFAULT`).

While in View mode, users can drag&drop the items on the wall. In addition they can add items using the options configured in the (collapsible) side bar. Items can be edited directly on the wall. Changes are applied immediately, example - changing the attitude of a participant moves the item to another column, changing the role updates the color of the item etc.

Use the **SDK** page to test all elements of the Buying Center Wall.

## Configuration Units

Learn about the configuration units of Buying centre wall.

The Buying Center Wall consists of the following configuration units:

- The context menu action `A_BuyingCenterY1` with action `BuyingCenter`. For details on the available action parameters, see [BuyingCenter Action Parameters](#) on page 401.

`A_BuyingCenterY1` is assigned to CRM process 1.2.4. "Opportunity Management" (CRM process ID 70000100).

- The header group `Y1.BuyingCenter` contains the following definitions:
- the `Expand` header configures the page header using the MiniDetails control of field group `Y1.BuyingCenter`.
- the `ExpandChildren` header configures the page's related data area displaying records from info areas **Opportunity Competitor** (Y3) and **Activity** (MA) in UPDATE\_DEFAULT.
- The side bar contains the following elements:
- A (hard-coded) **New Participant** button launching the `NewParticipantForOpportunity` process. This process is part of UPDATE\_DEFAULT and allows users to search for a company/person and add it as an opportunity participant.
- The **Suggested Participants** drop-down list allowing the user to select from suggested participants. Example - persons from the company linked to the opportunity, participants from activities linked to the opportunity etc. The suggestions are created by executing queries.

The available options are configured in UPDATE\_DEFAULT as menu actions in the `M_BuyingCenterParticipantSuggestionsY1` context menu. You can customize the options by adding queries that filter the results.

For a complete description of these options and their configuration, see the article "How to configure suggested participants in the Buying center" at <https://support.aurea.com>.

- If configured, the side bar also lists missing required and optional roles. The record count is updated as the user adds/removes/changes participant items.

Missing roles are configured via a JSON file on the server. The location of the file must be specified in the `BuyingCenter.MissingRoles.Y1.Definition` Web Configuration parameter. For further details, see [BuyingCenter.MissingRoles.Y1.Definition](#).



---

If undefined (default), this element is hidden from the side bar.

---

**Note:** No missing roles JSON file is delivered with Aurea CRM.Web since the "Participation" catalog determining the roles is a variable catalog and therefore different for every project.

---

For an example and a complete description of the required JSON structure and supported placeholders, see the article "How to configure Missing Roles for Buying center" at <https://support.aurea.com>.

- The items on the wall are configured by the `Y2.BuyingCenter` Expand configuration: This Expand configuration references the `Y2.BuyingCenter` table caption, field group and header group.

The Details control of field group `Y2.BuyingCenter` defines the look and contents of the items. The coloring of the items is based on the "Participation" catalog (implemented using a render hook for field **Text** (Y2:29)). The **Influence** field uses a `FieldValuePicker` widget.

- The static texts for the Buying Center Wall are defined in the `BuyingCenter` text group.

## BuyingCenter Action Parameters

Supported BuyingCenter Action Parameters.

The `BuyingCenter` action supports the following input arguments:

### **uid**

The record for which participants should be displayed (`Record` or `Linked`). Currently only opportunity records are supported.

This parameter is mandatory.

### **participantInfoAreald**

The info area of the participants. Currently only Y2 is supported.

This parameter is mandatory.

### **participantLinkId**

The link ID used to read participant records linked to the opportunity. This parameter is only relevant if multiple links between two info areas exist in the data model. for further details, see [Link IDs](#) on page 19.

**Default:** -1

### participantCatalogField

The catalog field (`<field name or number>`) whose values are used as categories (i.e. columns) of the buying center wall. `A_BuyingCenterY1` as defined in `UPDATE_DEFAULT` uses the **Attitude** field (Y2:7, XML name "Position").

This parameter is mandatory.

### participantFilterName

You can specify a fixed filter for Y2. Example - to hide participants that have been inactive for more than a year.

### participantSortFields

Allows you to specify a sort order for the items displayed on the wall.

Specify one or more fields (`<field name or number>`) from Y2. In `UPDATE_DEFAULT` the **Participation** field (Y2:6) is used.

Separate multiple fields by comma. Prefix a field with "-" (or "+") to define descending/ascending sorting.

**Default:** ascending.

### suggestionsPageSize

Defines the number of visible rows in the `RecordListBox` displaying suggestions.

**Default:** 10.

### createNewTab

See [createNewTab](#) in [Expand Action Parameters](#) on page 356.

## Calendar

Learn about the calendar action templates.

The `Calendar` action template (`calendar.aspx`) is used by the menu action `ID_NET_CALENDAR` in `UPDATE_DEFAULT`.

### Calendar Action Parameters

Supported Calendar Action Parameters.

The `Calendar` action supports the following input arguments:

#### uid

The record ID for a record to be displayed in the calendar, if the action is called from the MA context menu and `uid=Record`, the calendar opens to the date of the appointment record it was called from.

## DefaultFilterSet

Name of a filter set which is applied if the user has not yet selected a filter or filter set himself.

For information on the syntax, see [Calendar.FilterSets](#).

## FilterSet

Name of a filter set which is applied to filter items displayed in the calendar. It cannot be changed by the user.

For information on the syntax, see [Calendar.FilterSets](#).

## createNewTab

See [createNewTab](#) in [Expand Action Parameters](#) on page 356.

## menuName

Name of a menu to be used when clicking the **New Calendar Entry** button instead of `M_CalNew`.

## fixedFilterNames

Allows you to specify fixed filters which are applied to the calendar and cannot be changed by the user analogous to `fixedFilterName` in the Search action.

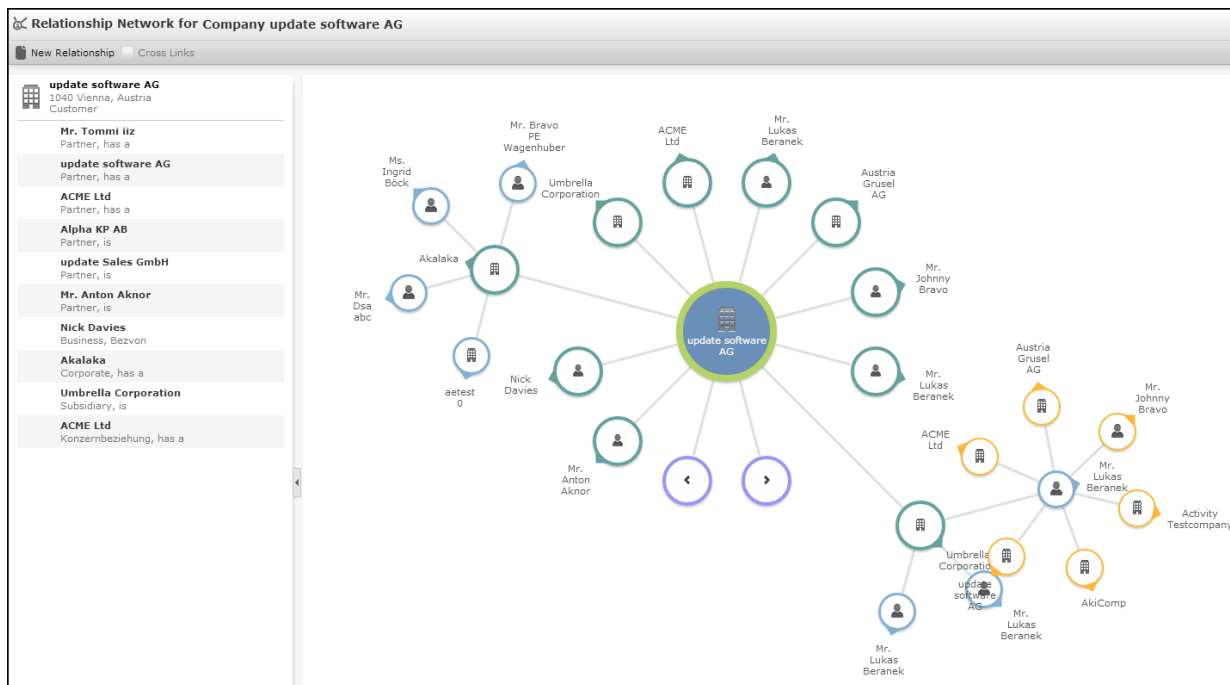
You can specify one filter for each supported info area (A1, A3, MA and TD). Enter a comma-separated list of filters. Example - `MA.NoEmail, A1.OnlyOpen`.

If the user selects or defines a filter set, the criteria are combined by logical AND.

## CircleOfInfluence

You can visualize the relationship between companies and persons using CircleOfInfluence action template.

The `CircleOfInfluence` action template (`circleofinfluence.aspx`) is used to visualize the relationships between companies and persons in an interactive **relationship graph** using the `CircleOfInfluence` widget.



In `UPDATE_DEFAULT` the `CircleOfInfluence` action is called by the `A_CircleOfInfluence` context menu action (**Relationship Network**) configured in `M_Company` and `M_Person`.

All companies/persons (FI, KP, PE, CP) with a relationship (PB record) to the current record are taken into account. The nodes represent company/person records, the lines represent the relationship records.

You can define which records are displayed by defining a fixed filter for the action. For further details, see [fixedFilterName](#).

If a node has more than 10 sub-nodes, two nodes with `<` and `>` are displayed allowing the user to "page" to the next 10 related companies/persons.

Users can expand/collapse further levels of the graph by clicking on a node. You can define the number of initially expanded levels via the `level` action parameter. For further details, see [level](#). The nodes are color-coded per level, i.e. depending on how remotely they are related to the center node.

The left part of the page (collapsible) lists all relationships for the currently selected node.

Use the **SDK** page to test the `CircleOfInfluence` widget.

To display the `CircleOfInfluence` widget in a sub-list panel (`SearchSubList` and `ExpandChildren` header), enter the following as **Form Name** (see [Related Data area \(Header Info Areas\)](#) on page 329):

```
{type: "CircleOfInfluence", options: { size: { height: 400 } } }
```

## Configuration Units

Learn about configuring Circle of Influence action template.

The Circle of Influence consists of the following configuration units:

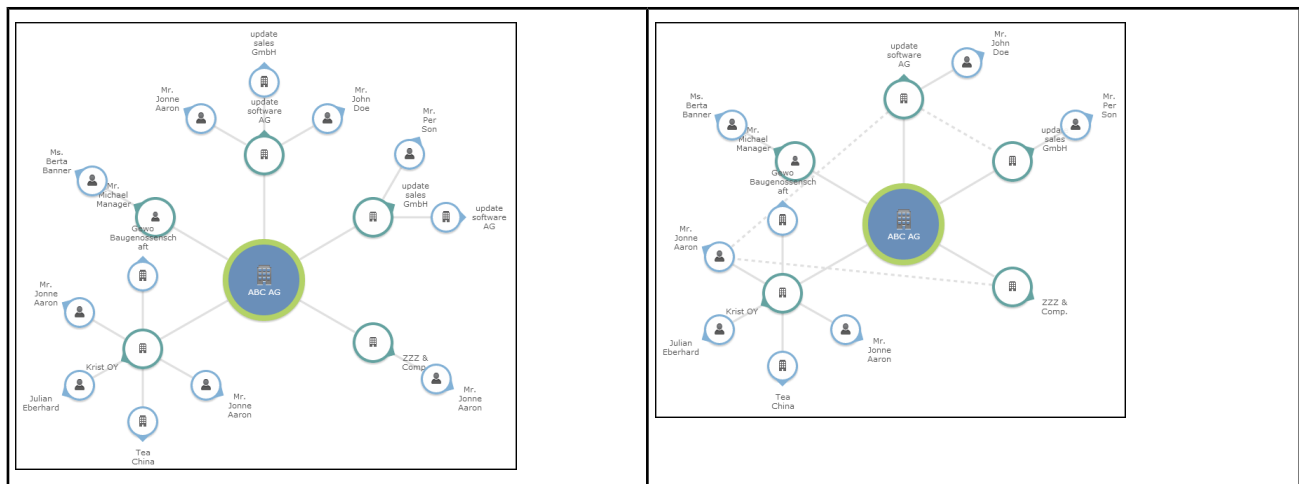
- The context menu action `A_CircleOfInfluence` with action `CircleOfInfluence`. For details on the available action parameters. For further details, see [CircleOfInfluence Action Parameters](#) on page 406.

`A_CircleOfInfluence` is assigned to CRM process 1.4.2.2.107. "Relationship hierarchy" (CRM process ID 50001410).

- The special header `CircleOfInfluence` containing the following:
  - "Circle of influence for {RECORD}" as header text. The info area default table captions (for FI, KP etc.) are used. Use `{RECORD:<Table Caption Name>}` to specify another than the default table caption.
  - A **New Relationship** button (`CoINewRelation`) calling the `CreateRelation` process in a popup for the currently selected node. If no record is selected, the button is disabled.
  - The **Cross Links** check box (`CoIToggleCrossLinks`) allowing users to turn on/off the displaying of multiple nodes for the same record

Example: If a person is related to a company and this company is related to another company to which the person is also related, the person is displayed as a child node of both companies.

If the **Cross Links** check box is enabled, the person is only displayed as related to the root company (continuous line) and the relationship to the other company is indicated by a dotted line to the same person node.



You can define an initial default via the `showCrossLinks` action parameter. For further details, see [showCrossLinks](#).

The header is displayed using page header style. It is not configurable.

- Special table captions `<InfoAreaId>.CoI` (fallback: `<InfoAreaId>`) configure the following:
  - The nodes use (the first line of) `<InfoAreaId>.CoI`, e.g. `FI.CoI`
  - The lines connecting the nodes display `PB.CoI` in a tooltip (on mouse-over)
  - The header of the `RecordListBox` (left part of the page) uses `<InfoAreaId>.CoI` for the currently selected node. Multi-line table captions are supported.
  - For the list of relationships `PB.CoI` is used.

## CircleOfInfluence Action Parameters

Supported CircleOfInfluence Action Parameters.

The `CircleOfInfluence` action supports the following input arguments:

### uid

The record to be displayed as the graph's central node (root node).

This parameter is mandatory.

`Record`: The current record is displayed as the central node.

`Linked`: The parent record of the current record is displayed as the central node.

### fixedFilterName

Allows you to specify a fixed filter, for `PB`, which is applied to the relationship graph. Example - filtering the nodes by **Relationship type**. The filter is applied to both the relationship graph and the list of relationship records.

In addition to defining filter criteria on `PB`, you can also filter `PB` records by criteria from `FI`, `PE`, etc. See the below examples for details.

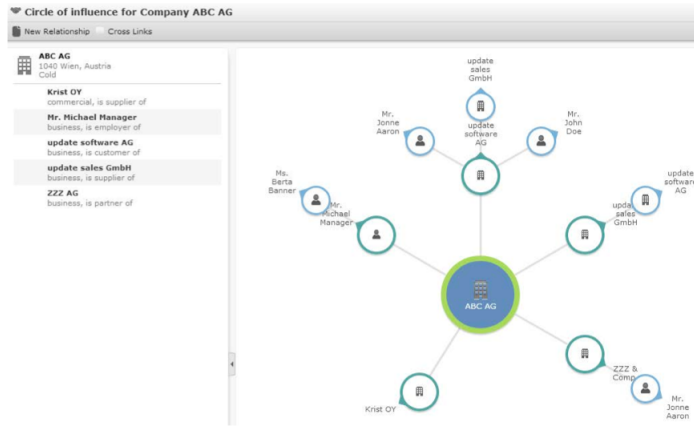
---

**Note:** The filter is not applied when creating/updating a relationship. Users need to refresh the page to see the changes.

---

## Examples

1. Relationship view for company "ABC AG" without a filter:



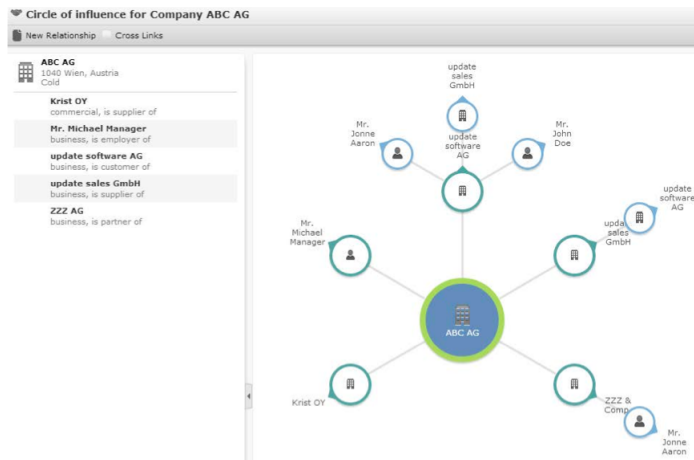
Relationship (PB) [Filter] [X]

Filter [Filter] [X] [OR] [v]

Relationship type (4) [v] family [0] CatCode = 4 [v] [X]

Relationship type (4) [v] personal [0] CatCode = 7 [v] [X]

2. Relationship view filtered by Relationship type: Private relationships are not displayed. In this case the person linked to Mr. Michael Manager:



3. Relationship view filtered by criteria from FI and PE:

Relationship (PB) [Filter] [X]

Filter [FI\_StaNo (0)] [v] [func] [v] [+FIELD]

Company (FI) [v] [X] LinkId 1: Link #1 [v] Relation WITH [v] Max Rows 0 [v] Optional [v]

Filter [Filter] [X] [OR] [v]

Company (2) [v] [X] = \*AG [v] [X]

Country (5) [v] [X] = Austria [0] CatCode = 1 [v] [X]

Person (PE) [v] [X] LinkId 1: Link #1 [v] Relation WITH [v] Max Rows 0 [v] Optional [v]

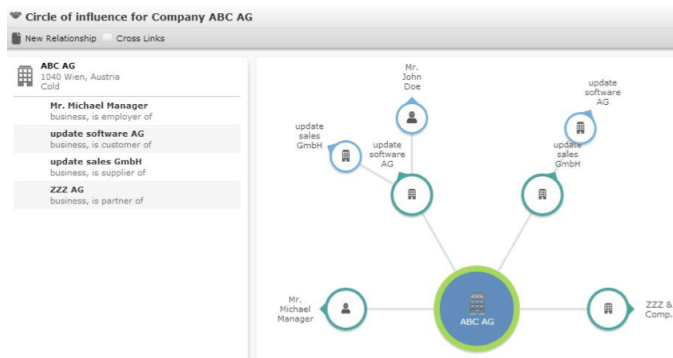
Filter [Filter] [X] [OR] [v]

Country (6) [v] [X] = Austria [0] CatCode = 1 [v] [X]

Add FI and PE as "child" info areas of PB and use the reverse link (link ID 1) to read the target records of the PB record. Otherwise the filter criteria would be applied to the selected node instead.

If you define conditions for more than one sub-info area, make sure you check Optional.

Only relationships to companies/persons which fulfill the defined criteria ("AG" at end of company name, country = Austria) are displayed:



### level

Number of levels that are initially visible.

**Default:** 1.

### pageHeaderGroup

The header group of `pageHeaderName`.

**Default:** Special.

### pageHeaderName

The header to be displayed.

**Default:** CircleOfInfluence.

### showCrossLinks

This parameter defines the initial default setting of the **Cross Links** check box.

**Default:** false.

## CreateRelation

Learn about CreateRelation action template.

The `CreateRelation` action template is used by step `CreateRelation->ViewRecord`.

`CreateRelation` process is called by `A_NewRelation` in `S_FINew` and `S_KPNew`.

For general information on [relationships](#), see *Web Business Logic User Guide*.

The `CreateRelation` action template supports the following input arguments:

### uid

The source record of the relationship. It is the company/person from which the `CreateRelation` process was started.



## uid2

The target record of the relationship.

## formName

The CreateRelation form allows the users to select or define the relationship. The `CreateRelation` action template supports the following output argument:

## relationUid

The newly created relationship record (PB).

**Add a new Relationship with Company Minions Ltd.**

Predefined | Customize

Do you want to create a reciprocal relationship?

**Search Relationship level**

Relationship type + ▾ Relationship + ▾  
 Business Area + ▾ Level  
 Reciprocal Rel. Type + ▾ Reciprocal Rel. + ▾  
 Reciprocal Rel. Bus. A... + ▾ Reciprocal Rel. Level  
 + Add more ... ▾

max. 5 ▾ Records per page Reset Start Search

**Search Results** << 1 - 5 of 5+ >>

Relationship type	Relationship	Bu...	Reciprocal Rel. Type	Recipr...	Reciprocal Rel. Bus. A...
business	supplier		business	customer	
business	supplier	Hardw...	business	customer	Hardware
corporate	property	Hardw...	corporate	owner	Hardware
business	partner		business	partner	

<< 1 - 5 of 5+ >>

Minions Ltd. is property of Mr. Gru  
 Mr. Gru is owner of Minions Ltd.

The `CreateRelation` form is configured as follows:

- The **Predefined** tab consists of a `SearchView` widget using the search and list controls from field group BL (B5 for OTC and FS). You can customize the search by defining `fixedFieldValues`, a `filter` etc. The `SearchView` widget allows users to select a relationship from the **Relationship Level** info area (BL/B5).

The **Relationship Level** info area is accessible via **Settings > Maintenance > Relationship Level** in `UPDATE_DEFAULT` (implemented in the Base vertical for info area 0BL). This allows users to create/manage relationship level records directly in Aurea CRM.Web. Both the menu item and the **New** button (in the page's `SearchResults` header) are assigned to CRM process 1.4.2.2.107 "Relationship hierarchy" (CRM process ID 50001410).

- The contents of the **Customize** tab are defined in the form, allowing users to define custom relationships by e.g. just selecting a **Relationship type**.

Selecting a record (or entering a value on the **Customize** tab) updates the preview text at the bottom of the page. (Double-clicking a record creates the relationship record.)

The preview text is created as follows:

**BTB:**

<uid1> + relationship\_text\_4 + BL.Relation table caption + relationship\_text\_5 + <uid2>

<uid2> + relationship\_text\_4 + BL.ReciprocalRelation table caption + relationship\_text\_5 + <uid1>

**OTC/FS:**

<uid1> + relationship\_text\_4 + B5.Relation + relationship\_text\_5 + <uid2>

<uid2> + relationship\_text\_4 + B5.ReciprocalRelation + relationship\_text\_5 + <uid1>

In UPDATE\_DEFAULT the table captions use field BL:1 and BL:5 (B5:1 and B5:5). Depending on which field(s) from BL/B5 you want to display and which catalog values are present in your environment, change the table captions and/or the texts (no. 4 and 5 of text group relationship\_text) in order to provide a helpful preview text.

## DayWeekPlanning

Learn about the DayWeek Planning action template.

The DayWeekPlanning action template (DayWeekPlanning.aspx) is used by the menu action ID\_NET\_DAYWEEKPLANNING in UPDATE\_DEFAULT.

The screenshot displays the 'Day & Week Planning' application interface. At the top, there is a search bar for 'Search Companies' with fields for 'Company', 'Country' (set to Austria), and 'From ZIP'. Below the search bar is a table of search results with columns: Company, Country, Postal Code, City, Street, Country Code, Area Code, Telephone, Modem, and FreeC3. One row is highlighted: 'update Software' in Austria, postal code 1040, Vienna, Opengasse 1... Below the search results is a calendar view for the week of 2.7.2012 to 6.7.2012. The calendar shows various activities scheduled across the days, such as 'call me (Eco Hotel ...)', '(update Software)', 'Gabelfrühstück', 'test', 'my visi (Krist Intl. OY)', and 'Täglich anrufen!'. On the left side of the calendar, there is a monthly overview for July 2012 and 'Activity Defaults' for 'Visit' with fields for Subject, Duration (30), Unit (Min), and Purpose (Support).

The **Daily & Weekly Planning** page consists of a Search page with the Calendar displayed below. It allows users to plan activities: The user searches for relevant records and then drags any record onto the calendar to create an activity. If multiple records are selected, the activity records are created in the order of selection.

Daily & Weekly Planning is possible for companies, persons, and any info area that has a company and/or person as a parent record. In UPDATE\_DEFAULT, Daily & Weekly Planning is configured for FI, KP, PE (FS only) and SB (OTC only).

To make especially visit planning easier, it is also possible to display the contact times of the selected company or person in the calendar. For other info areas, the contact times of the parent person or company can be displayed.

---

**Note:** By default, Aurea CRM.Web uses the **Contact Time** info area (CT) to display contact times on the **Daily & Weekly Planning** page. For further details, see [contactTimeTypes](#).

---

**Note:** If the contact times are stored differently in your project, you can implement a custom contact times provider, see the article “How to Implement a custom contact times provider” at <https://support.aurea.com>.

---

Daily & Weekly Planning consists of the following configuration units:

- The menu action `ID_NET_DAYWEEKPLANNING` with action `DayWeekPlanning`. For details, see [DayWeekPlanning Action Parameters](#) on page 412.
- The special header `DayWeekPlanning` contains the drop-down list (`InfoAreaSets` button) that allows users to choose the info area in which they want to search. The contents of the drop-down list is configured by the `searchAndListName` action parameter. For further details, see [searchAndListName](#).

The user's choice in which info area he wants to search is stored in the user's roaming settings. For further details, see [User Settings](#) on page 25.

- The area to filter the displayed calendar items is configured by the special header `CalendarAdjustments`. For further details, see [CalendarAdjustments](#).
- The **Activity Defaults** area is defined by the special header `DayWeekPlanningDefaultValues` containing text plus an optional image and the field group `MADayWeekPlanningDefaults`: The Details control of this field group contains the fields offered in the **Activity Defaults** area. The control needs to be set to one column via control attribute. For further details, see [columns](#) on page 201. Use the `defaultFieldValuesForMA` action parameter if you want to enter activity defaults. For further details, see [defaultFieldValuesForMA](#). Use `defaultFieldValuesForMA` to overrule the default activity duration of 1 hour.
- Above the calendar area, the special header `DayWeekPlanningCalendar` containing a subset of the elements from the `CalendarHeader` is displayed. For further details, see [CalendarHeader/DayWeekPlanningCalendar](#).

To allow users to switch from Calendar to Day & Week Planning and vice versa, the buttons `Calendar` and `DayWeekPlanning` are configured in these headers.

---

**Note:** The Web Configuration parameter `Search.AutoExpandOnSingleHit` is ignored for Day & Week Planning. The default action is not executed when only one search result is found on this page. For further details, see [Search.AutoExpandOnSingleHit](#).

---

## DayWeekPlanning Action Parameters

Supported dayweekplanning action parameters.

The `DayWeekPlanning` action supports the following input arguments:

### searchAndListName

Defines the info areas that are available for users to select and search in. Specify the info areas and (optional) the Search&List configurations separated by semicolon. This parameter is mandatory.

Defaults:

- **BTB:** `KP;FI`
- **FS:** `PE;KP;FI`
- **OTC:** `KP;FI;SB`

### Syntax:

```
infoAreaId:searchAndList;...
```

### Example:

```
FI:FIsearchAndList;KP:KPsearchAndList
```

The defined info areas are offered in a drop-down list in the top right corner of the page. The first entry is used as default when opening the page for the first time. The entries are sorted alphabetically.



If you defined info areas that are not children of FI and/or KP (e.g. AR), an error message is written to the client log.

### maxRows

Defines the maximum number of results returned by the search.

### defaultFieldValuesForMA

Allows you to define **Activity Defaults** displayed on the left below the calendar filters which are used when creating new activity records. The defaults can be changed by the user.

### Syntax:

---

```
<field name or number>:"<value>";...
```

**Example:**


`Contact:"#1";Duration:"30";Unit:"#2"` sets the **Activity Type** to "Visit" and the default duration to 30 minutes

---

**Note:** Values for the fields **Starts on**, **Starts at**, **Rep ID** and **Participants ID** are not retrieved from the defaults: If present, they are ignored. **Starts on** and **Starts at** are set according to where the user drops the record. The current user is always set as rep and participant.

---

**showSearch**

If set to `true`, the search mask is visible. If set to `false`, it is displayed collapsed and can be expanded by clicking .

**Default:** `true`.

**autoSearch**

If set to `true`, the search is immediately executed when the user opens the page.

Auto-search does not happen when the user updates a filter in the search mask. If you want auto-search to be activated when a filter is updated in the search mask, you have to set [autoApplyFilter](#) to `true`.

**Default:** `false`

**defaultFieldValues**

Defines fields that have a default value within the search, which the user can change, separated by semicolon. You can define default values for all info areas defined in `searchAndListName`.

**Syntax:**

```
<infoAreaId>.<field name or number>:"<value>";...
```

- To fill a field with the current rep, enter `fieldName:REP`
- To fill a field with the current date, enter `fieldName:TODAY`

Use operators (`<=`,`=>`) to define date ranges and functions

Syntax: `fieldName:Value:Operator`

- To set a value for a catalog field, enter `fieldName:"#catalog_ID"`

**Example:**

```
KP.RepID:REP;PR.Date:TODAY-14;MA.Contact:"#3";MA.Date:TODAY-7:>=;MA.Date:TODAY+14:<=
```

This sets the search masks as follows:

- Person: **Rep ID** = the current rep
- Offer: **Date** = the current date minus 14 days
- Activity: **Activity Type** = Event, **Date from** = the current date minus seven days, **Date to** = the current date plus 14 days

### fixedFieldValues

Defines fields that have a fixed value within the search which the user cannot change. You can define fixed values for all info areas defined in `searchAndListName`. Use the syntax as described for [defaultFieldValues](#).

### disableFilter

Disables the filter selection and definition functionality for the user. Both the predefined filters and the **More Filters** entry are removed from the **Add more** drop-down list of the Search page.

---

**Note:** To globally restrict working with filters to predefined filters only, create and check a Web Configuration parameter called `Search.NoFilterDefinition` (type: Checkbox). This removes the **More Filters** entry from the **Add more** drop-down list on all Search pages. For further details, see [Search.NoFilterDefinition](#). Users may only select predefined filters and can no longer create or edit any filters in Aurea CRM.Web.

---

### defaultFilterName

Allows you to specify a filter which is applied to the search by default. Users are still able to select another filter unless `disableFilter = true`.

---

**Note:** Filters have to be referenced as `<InfoAreaId>.<FilterName>` in the action (example - "KP.Filter1"). In Aurea CRM.Web they are then displayed as `<FilterName>` (example - "Filter1").

---

### fixedFilterName

Allows you to specify a fixed filter which is applied to the search and cannot be changed by the user. The filter name is not visible for the user.

---

**Note:** Filters have to be referenced as `<InfoAreaId>.<FilterName>` in the action definition (example - `KP.Filter1`).

---

### listParams

Defines the behavior of the search results list. For further details, see [ListParams](#) on page 374.

---

**Note:** `selectionMode` is not supported in the `DayWeekPlanning` action template.

---

## linkOfMAToSource

When creating activity records for info areas other than **Company** or **Person**, example - a **Ticket**, the link to the ticket record is not automatically added to the newly created activity. Records of these info areas are by default only used to determine the company and/or person record(s) for which the activity is created.

Use the `linkOfMAToSource` parameter to add links from the activity record to these info areas.

Syntax (JSON): `{infoAreaId:linkId,infoAreaId:linkId,...}`

---

**Note:** Each info area may only be used once in this structure.

---

Example: `{KM:-1;IB:-1}`

This creates links to ticket and installed base records using the default link. For further details, see [Link IDs](#) on page 19. The links are displayed under **Parent Data** in the activity record's Expand view.

## contactTimeTypes

Defines the **Type** of **Contact Time** (CT) that is taken into account when displaying contact times.

If `contactTimeTypes` is undefined, the contact times are only displayed if CT records exists for exactly one **Type**. If contact times for more than one type are configured, no contact times are displayed.

### Syntax

`<catalog value ID>`

`<info area ID>=<catalog value ID>`

Separate multiple entries by comma.

### Examples

2

The contact times of **Type** "Reachability" are displayed for companies and persons.

`FI=1,KP=2`

The contact times of **Type** "Visiting Times" are displayed for companies and of **Type** "Reachability" for persons.

## DefaultValue

Use `DefaultValue` action template to define default values in Aurea CRM.Web.

The `DefaultValue` action template is called by the `DefaultValuesDefine` button in `UPDATE_DEFAULT`.

Example: MA SearchResults header.

For further details, see [DefaultValuesDefine](#) on page 309. It is used for defining default values in Aurea CRM.Web. The default values are saved in a format of type "Info Area Default Values".

---

**Note:** Default values defined in the Aurea CRM win **Rights** module take precedence over the defaults defined via the `DefaultValuesDefine` button in Aurea CRM.Web.

---

## DefaultValue Action Parameters

Supported default value action parameters.

The `DefaultValue` action supports the following input arguments:

### InfoAreaId

The info area for which default values are defined. Enter `javascript:${['uid']}.infoAreaId` to determine the `InfoAreaId` based on the context, i.e. the current record.

### DefaultValuesFormatName

Name of a default values format to be loaded. The loaded default values can be changed by the user.

### FieldGroupName

Name of the field group used: The Details control of this field group is used for defining the default values format.

## DispatchingDashboard

Learn about the `DispatchingDashboard` action template in this topic.

The `DispatchingDashboard` action template (`DispatchingDashboard.aspx`) is used by the application menu action `ID_NET_DISPATCHING_DASHBOARD` and the context menu action `A_AssignTicket` in `UPDATE_DEFAULT`.



The screenshot displays the 'Search Tickets' interface. At the top, there is a search bar and a list of search results. The results table includes columns for No., Priority, Subject, Status, Created on, End Customer Company, Problem, and Processed... The selected ticket (No. 9978573) has a subject of 'Printer breakdown' and is in 'Open' status.

Below the search results is the 'Dispatching Dashboard' for the period '22.04 - 26.04.2013'. It features a calendar on the left and a grid of activity records on the right. The grid columns represent days from Monday (22.04.2013) to Friday (26.04.2013). The rows are categorized by 'Reps' (Austria, Finland) and 'Resources' (Mtg. Room, Beamer). Activity records are shown as icons (phone, person, etc.) or text labels (e.g., 'Startup Error') within the grid cells.

The **Dispatching Dashboard** allows the assignment of open tickets to - service technicians or support employees as well as to resources like meeting rooms or special tools required to fix the problem.

Working hours (global and individual) are displayed as described for the Resources view. For further details, see [ResourcesView](#) on page 448.

The **Dispatching Dashboard** page consists of a Search page with the Resource view displayed below. Dragging one or more ticket records (KM) onto the Resource view creates activity records (MA) linked to the respective ticket.

When called via `A_AssignTicket` the search results only contain the ticket from which the action was called (search context = new ticket).

Activity records that would be hidden when selecting a large time scale, can be displayed as small dots by checking the `ResourceView.ShowHiddenItems` Web Configuration parameter. For further details, see [ResourceView.ShowHiddenItems](#). Users can change this default by setting/removing the **Show hidden items** flag in the Resource View Settings.

Multiple activities belonging to the same ticket and activities belonging to hierarchical tickets (**Predecessor** field) are highlighted with an orange underline when a ticket or a related activity is selected.

You can update existing hierarchical tickets with a link to the respective root ticket (**KM\_2\_StaNo/SerNo/Link** fields) via a CRM.server to-do (type 18). For further details, see [To-Do Types](#) in the *CRM.Core Administrator Guide*.

The MA records are created as follows:

- The MA rep is the rep the ticket record was dropped on.
- Date and time is calculated from the position where the ticket was dropped.
- The MA record is linked to the KM record (via the respective link fields in MA).
- The MA is linked to the same company/person for which the ticket was created.
- More MA fields can be automatically entered based on the `defaultFieldValuesForMA` action parameter. For further details, see [defaultFieldValuesForMA](#).
- The MA record's duration is calculated based on the **Service Duration** (KM:117) of the ticket. Users can overwrite this default by changing the **Activity Defaults**.

To update ticket records once they are dropped onto the Resources view you need to create an appropriate trigger. Example - to set the status to "in progress".

The user's settings (rep set, filter set, grouping criterion and activity defaults) are stored client-independently using the settings service.

## Configuration Units

Learn about Dispatching Dashboard configuration units.

The **Dispatching Dashboard** configuration consists of the following configuration units:

- The menu actions `ID_NET_CALENDAR` and `A_AssignTicket` with action `DispatchingDashboard`. For details on the available action parameters. For further details, see [DispatchingDashboard Action Parameters](#) on page 421.
- The Search&List configuration, specified via the `searchAndListName` action parameter, defining the upper part of the page. For further details, see [searchAndListName](#).
- The special header `DispatchingDashboard` containing a `DuplicateRecord` and a `ResourcePlanSettingsEdit` button by default. The `DuplicateRecord` button allows users to duplicate an activity record in order to split the processing of a ticket into multiple activities. The `DuplicateRecord` button is only active if one record is selected.
- The items displayed in the Resource view use the table captions `MA.ResourceView` (and `A3.ResourceView`). For small items and multi-line table captions the complete table caption is displayed in the tooltip.
- The sidebar contains the date picker, drop-down lists for filter and rep sets and the **Group reps by** drop-down list. The additional options **Ticket-Related Reps** and **Combine Skills & Rep Set** allow further filtering for the appropriate reps and resources. For further details, see [Filtering Reps for Dispatching](#) on page 419

You can create a Web Configuration parameter `ResourceView.FilterSets` to offer predefined filter sets. For further details, see [ResourceView.FilterSets](#).

You can customize the **Group reps by** drop-down list via the `repGroupingOptions` action parameter. For further details, see [repGroupingOptions](#).

- The **Activity Defaults** area allows users to enter default values for the activity records to be created (example -. the **Activity Type**). It is defined by the special header `DayWeekPlanningDefaultValues` containing text plus an optional image and the field group `MADispatchingDashboardDefaults`: The Details control of this field group contains the fields offered in the **Activity Defaults** area. The control needs to be set to one column via control attribute. For further details, see [columns](#) on page 201.

Use the `defaultFieldValuesForMA` action parameter to predefine activity defaults. For further details, see [defaultFieldValuesForMA](#).

---

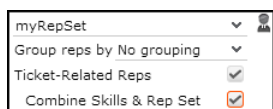
### Note:

- Settings made by the user in the **Activity Defaults** area overrule the automatic MA duration calculation (based on the **Service Duration** of the ticket) and values entered via `defaultFieldValuesForMA`.
  - Values for the MA fields **Starts on**, **Starts at**, **Rep** and **Participants** are **not** retrieved from the defaults (if entered, they are ignored). **Starts on** and **Starts at** are set according to where the user drops the ticket record.
- 


- Context menus: `M_ResourcesRepMenu`, `M_ResourcesRepMenu.MultiSelect`

## Filtering Reps for Dispatching

Learn about Filtering Reps for Dispatching.



Users have the following possibilities to filter the reps displayed on the Dispatching Dashboard:

- **Reps and Rep Sets:** Users can select reps and/or define rep sets via the  (**Reps and Rep Sets**) button (analogous to the Calendar).

---

**Note:** Max. 50 reps can be displayed. If the user's selection contains more than 50 reps, a message is displayed.

---

- **Ticket-Related Reps:** If checked, the rep list is filtered for reps with the appropriate skills. The selection from **Reps and Rep Sets** is disregarded in this case.

**Note:** To filter for ticket-related reps you need to define **Dispatching** options (**Resources** category) in the **Configuration** info area (MC). For further details, see [Configuration Info Area](#) in the *CRM.Core Administrator Guide*.

- **Combine Skills&Rep Set:** This option is only displayed, if **Ticket-Related Reps** is checked. It allows users to combine rep filter and skill: This results in a subset of reps from the selected rep set. Only reps from the selected rep set with the skill required for the any ticket selected in the list are displayed.

You can define the initial setting for **Ticket-Related Reps** and **Combine Skills&Rep Set** via the corresponding action parameters. For further details, see [ticketRelatedReps](#) and [combineSkillsAndGroup](#).

## Displaying Record Lists/Details for the Dispatching Dashboard

Learn to Display Record Lists/Details for the Dispatching Dashboard.

The screenshot displays the CRM Dispatching Dashboard interface. At the top, there is a 'Search Tickets' section with a search bar and a settings icon. Below this is the 'Search Results' section, which shows a table of tickets with columns for No., Priority, Subject, Status, Created..., End Customer Company, Problem, and Processed by ID. The table contains five rows of ticket data.

Below the search results is a calendar view for August 2014. The calendar shows a grid of days from August 1st to August 31st. To the right of the calendar is a list of representatives (Reps) including Administrator, Chris Aki, John Doe, Kazuko Kanzaki, Lisa Simpson, Miranda Meyer, Peter Parker, Simon Eder, Simon Svendsen, and Stephen Smith. The calendar cells contain small icons representing activities for each rep on specific days.

At the bottom of the dashboard, there is a list of activity details for August 2014, including telephone calls and visits for various customers like Ms. Leena Krist and Mr. Jonne Aaron.

UPDATE\_DEFAULT contains an `Example.DispatchingDashboard` form that allows you to display a list of records for the selected rep / an Expand view of the selected activity record below a `DispatchingDashboard` widget.

For details, see [Displaying Record Lists/Details for a Resource View](#) on page 452.

## DispatchingDashboard Action Parameters

Supported DispatchingDashboard Action Parameters.

The `DispatchingDashboard` action supports the following input arguments:

### combineSkillsAndGroup

If set to `true`, the **Combine Skills&Rep Set** option is checked when first calling the Dispatching Dashboard.

**Default:** `false`.

### ticketRelatedReps

If set to `true`, the **Ticket-Related Reps** option is checked when first calling the Dispatching Dashboard.

**Default:**`false`.

### refreshListAfterDrop

If set to `true`, the ticket list is refreshed each time a ticket is dropped onto a resource block.

This allows you to automatically remove already dispatched tickets from the search results list. This requires additional project customizing: Define a trigger that writes to some field in KM as soon as the MA is created, and define a filter for the KM list filtering for that field's value.

**Default:** False

### repGroupingOptions

Defines the content and initial value of the **Group reps by** drop-down list using JSON.

Syntax. For further details, see [RepGroupingOptions](#).

### copyFields

Field values to be copied from the ticket to the activity.

#### Syntax:

```
<field name or number in KM>:<field name or number in MA>;
```

Separate multiple entries by semicolon.

#### Example:

```
Title:Subject;ProblemDescription:Text
```

### defaultFieldValuesForMA

Allows you to define **Activity Defaults** displayed on the left below the filter options which are used when creating activity records. The defaults can be changed by the user.

#### Syntax:

```
<field name or number>:"<value>";...
```

#### Example:

Contact:"#1";Purpose:"#2" sets the **Activity Type** to "Visit" and the **Purpose** to "Support".

---

**Note:** Values for the MA fields **Starts on**, **Starts at**, **Rep** and **Participants** are not retrieved from the defaults (if entered, they are ignored). **Starts on** and **Starts at** are set according to where the user drops the ticket record.

---

### defaultFilterName

Allows you to specify a filter which is applied to the ticket search by default. Users are still able to select another filter unless `disableFilter = true`.

### fixedFilterName

Allows you to specify a fixed filter which is applied to the ticket search and cannot be changed by the user. The filter name is not visible for the user.

### link

Parent record for the ticket search: `Record` or `Linked`. This parameter is used when calling the Dispatching Dashboard from a ticket record via `A_AssignTicket` in `UPDATE_DEFAULT`.

### listParams

Additional list parameters separated by semicolon: `key,value;key,value; etc`. For further details, see [ListParams](#) on page 374

### searchAndListName


The name of the Search&List configuration to be used.

If undefined or not found, the `KM` Search&List configuration is used

### maxRows

Defines the maximum number of results returned by the search.

### showSearch

If set to `true`, the search mask is visible. If set to `false`, it is displayed collapsed and can be expanded by clicking .

**Default:** True

### autoSearch

If set to `true`, the search is immediately executed when the user opens the page.

Auto-search does not happen when the user updates a filter in the search mask. If you want auto-search to be activated when a filter is updated in the search mask, you have to set [autoApplyFilter](#) to `true`.

**Default:** false

### defaultFieldValues

Default field values for the ticket search can be changed by the user.

**Syntax:** `<infoAreaId>.<field name or number>:"<value>"`;

### fixedFieldValues

Fixed field values for the search can not be changed by the user.

**Syntax:** `<infoAreaId>.<field name or number>:"<value>"`;

### disableFilter

Disables the filter selection and definition functionality for the user.

### showNewButton

When using the `DispatchingDashboard` action within a process add the `showNewButton` parameter and set it to `false` to hide the **New Activity** button, since launching a process inside a process is not supported.

## ExecuteReport

Use execute report action template to display report.

The `ExecuteReport` action template is used for displaying reports defined in Aurea CRM win. For further details, see [Creating Aurea CRM win Reports in Aurea CRM.Web](#) on page 863.

## ExecuteReport Action Parameters

Supported ExecuteReport Action Parameters

The `ExecuteReport` action supports the following input arguments:

### name

The name of the report format to be used. For information on Aurea CRM win reports. For further details, see [Defining Report Formats](#) in the *Aurea CRM win User Manual*.

This parameter is mandatory.

## uid

Optional input record for the report: `Record` or `Linked`.

If `uid` is set and no record is selected, the report is created for all existing records, i.e. no `uid` is passed.

---

**Note:** The **Context** setting in the Aurea CRM win report format needs to be consistent with the action definition, i.e. if **Context** = "Current Record" and no record is selected, creating the report results in an error.

---

## parameters

Additional parameters passed to the report as JSON. Example - `{target: 'Create-NewTab', contentDisposition: 'inline'}`

For information on defaults and available parameters. For further details, see [Creating Aurea CRM win Reports in Aurea CRM.Web](#) on page 863.

## scope

Defines which kind of report format is to be used if two formats with the same name exists: `Public` or `Private`.

## HierarchicalCatalogTreeView

This action template is used to display hierarchical catalogs in a tree view.

Use the `HierarchicalCatalogTreeViewactiontemplate(HierarchicalCatalog.aspx)`, to display hierarchical catalogs in a tree view.

Example in `UPDATE_DEFAULT`: The menu action `A_AllInterests` in the `S_KPAll` menu displays all **Interests** (IT) for a person record.



Users can check/uncheck the desired catalog values. The respective **Interest** records are created in the background.

The `HierarchicalCatalogTreeView` action template can be used for all info areas where records consist of the value of a hierarchical catalog.



---

## HierarchicalCatalogTreeView Action Parameters

Supported HierarchicalCatalogTreeView Action Parameters.

The `HierarchicalCatalogTreeView` action supports the following input arguments:

### ChildCatalog

The field number of the child catalog. It is mandatory.

### ColumnCount

Determines the number of columns, per row, displaying the child catalog values.

**Default:** 5.

### CompanyRelated

If set to `true` for a company-related tree view, only records linked directly to the company, but not to a person, are displayed. Example - for all interests of a company, `true` means that only interests exclusively linked to the company are displayed, while `false` lists interests linked to persons as well.

**Default:** false.

### ExpandAll

If set to `true`, all parent catalog values (example - interest groups) are expanded including those where no child value (example - interest) is currently selected.

**Default:** false.

### HeaderGroupName

The header group of `HeaderName`.

### HeaderName

An optional header for the tree. You can specify any special header from `Header-GroupName`.

### InfoArea/RecID

Record to which the new record is linked (mandatory): `Record` | `Linked`.

### LinkInfoArea

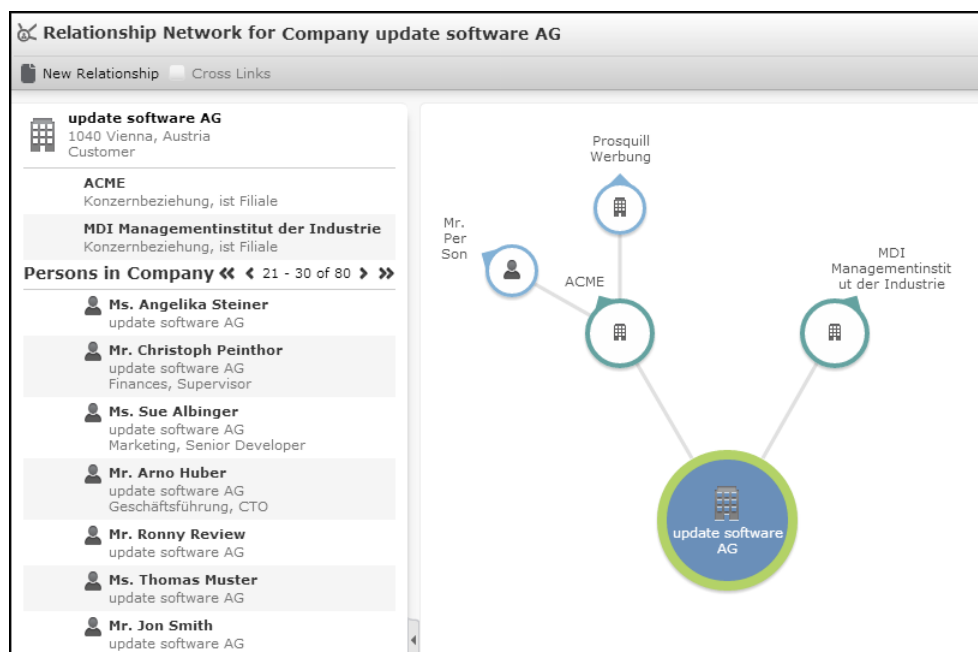
The info area of the new record (mandatory). Example - `IT`.

## HierarchicalCircleOfInfluence

Use HierarchicalCircleOfInfluence action template to visualize corporate structures in an interactive relationship graph.

The HierarchicalCircleOfInfluence action template can be used to visualize corporate structures - a network of branch offices, the structure of hospital departments, etc. - in an interactive relationship graph. The CircleOfInfluence widget (`circleofinfluence.aspx`) is used (see [CircleOfInfluence](#)), but only PB records that define the corporate structure are taken into account.

Relationships are defined as belonging to a corporate structure via the **Relationship hierarchy** flag in the **Relationship Level** info area (BL/B5), see [Relationships](#) and [Corporate Structure](#) in the *Web Business Logic User Guide*.



The root of the corporate structure, i.e. the topmost record in the hierarchy is always displayed in the center of the graph, no matter which record is currently selected. If more than one root record is found, i.e. if the corporate structure is defined erroneously, an error message is displayed.

The sidebar contains the following items:

- All companies/persons (FI, KP, PE, CP) that are related to the selected node as children in the corporate hierarchy

The special table caption `PB.CoI.Hierarchy` is used (fallbacks: `PB.CoI`, `PB`).

- If the selected node is a company, the persons (KP) in this company are listed as well.

The special table caption `KP.CoI` is used (fallback: `KP`).

---

For information on the functionality of the `CircleOfInfluence` widget, on the configuration units and the action parameters involved, see [CircleOfInfluence](#).

Since corporate structures are defined in the database, `UPDATE_DEFAULT` does not contain a menu action using the `HierarchicalCircleOfInfluence` action template.

The `level` argument is not used by the `HierarchicalCircleOfInfluence` action template. The `fixedFilterName` argument only accepts filters on PB.

`HierarchicalCircleOfInfluence` supports the following additional argument:

## hierarchyIndex

Learn to define `hierarchyIndex`.

You can define up to 10 different corporate structures via Z5 variables (`CORE::INDIRECT-PB-LINK-TREE-0` to 9). Use the `hierarchyIndex` parameter to specify which corporate structure should be read and displayed. If more than one structure is found and `hierarchyIndex` is undefined, a message is displayed and the corporate structure with index 0 is used.

## HierarchicalSerialEntry

Use `HierarchicalSerialEntry` to implement 'quick add pages' that contain a hierarchical item master tree.

The `HierarchicalSerialEntryactiontemplate(hierarchicalserialentry.aspx)` allows the implementation of quick add pages that contain a hierarchical item master tree visualizing product hierarchies (item structures). This allows users to select items from the tree instead of searching for item names or numbers.

---

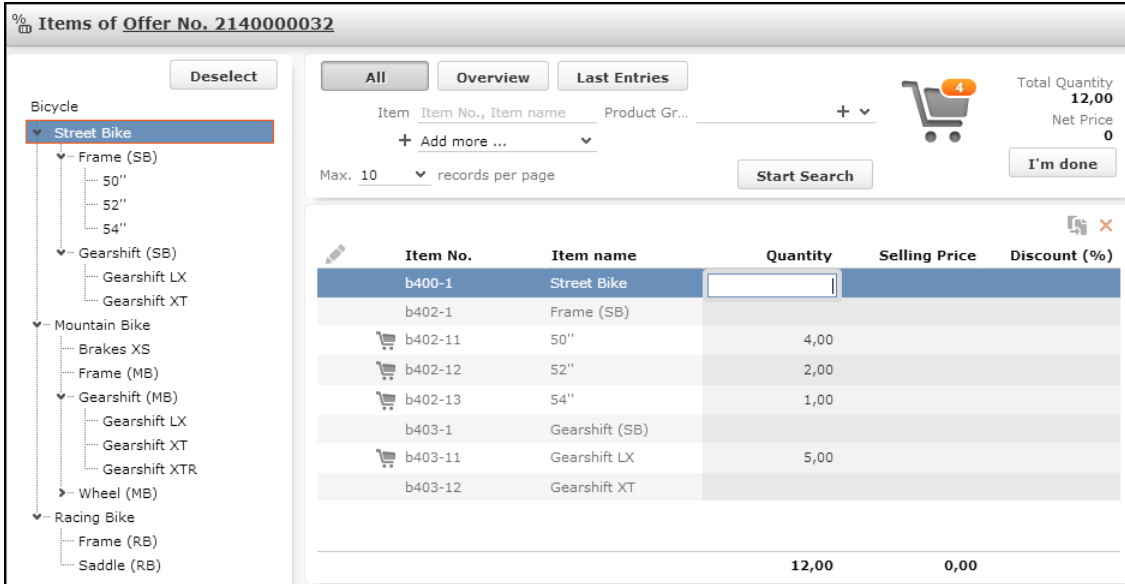
**Note:** The `HierarchicalSerialEntry` action template can only be used for **generic** quick add configurations.

---

Product hierarchies are defined in the **Item Structure** (IS) info area in Aurea CRM. Web accessible as a sub-list of the **Item Master** (AR) info area. For detailed information on defining product hierarchies, see [Item Structure](#) in the *Web Business Logic User Guide*.

For general information on how to configure quick add pages, see [SerialEntry](#) on page 454.

For additional background information, see the article “Hierarchical Serial Entry” at <https://support.aurea.com>.



Items of Offer No. 214000032

Deselect

All Overview Last Entries

Item Item No., Item name Product Gr... + v

+ Add more ... v

Max. 10 records per page Start Search I'm done

Total Quantity 12,00  
Net Price 0

Item No.	Item name	Quantity	Selling Price	Discount (%)
b400-1	Street Bike			
b402-1	Frame (SB)			
b402-11	50"	4,00		
b402-12	52"	2,00		
b402-13	54"	1,00		
b403-1	Gearshift (SB)			
b403-11	Gearshift LX	5,00		
b403-12	Gearshift XT			
		12,00	0,00	

The item structure to be read for displaying the item master tree is determined by the action definition by specifying an item number or uid. For further details, see [HierarchicalSerialEntry Action Parameters](#) on page 428. The item names displayed as tree nodes are read from the **Item Master** field in IS (IS:35).

Depending on the node selected in the tree, the items list is filtered for the respective sub-items. Additional criteria from the item search are combined by logical AND.

You can apply filters to both the tree and the list, e.g. to only display parent nodes in the tree (`hierarchyNodeFilter`) and actual items in the list (`fixedFilter`). For further details, see [hierarchyNodeFilter](#).

The selection in the tree is taken into account if **All** is selected. Selecting any other filter option (example - **Overview**) clears the selection in the tree.

To reset the items list while another filter option is selected, users can click on **Deselect** above the tree. This clears the selection and displays the items according to the selected filter option and search criteria regardless of whether the items are part of the item structure or not.

Since item structures are defined in the database, UPDATE\_DEFAULT does not contain a menu action using the `HierarchicalSerialEntry` action template.

Use the **SDK** page to test your hierarchical quick add configurations: **SDK > SerialEntry > HierarchicalSerialEntry**.

## HierarchicalSerialEntry Action Parameters

Supported HierarchicalSerialEntry Action Parameters.

The `HierarchicalSerialEntry` action template supports the same input arguments as `SerialEntry`. For further details, see [SerialEntry Action Parameters](#) on page 462. In addition, you need to at least specify the root record for the item structure to be read.

## Example

Argument	Value
name	Offer(AR=>AP)
uid	Record
hierarchyRootId	b1-1
optionsFormName	SerialEntry.Default.Options
copySourceFields	ItemNo,ItemName,Department,ProdGrp,Product
copyTargetFields	ItemNo,ItemName,Department,ProdGrp,Product
recentFilterOptions	FI,OfferDate,3

`HierarchicalSerialEntry` supports the following additional input arguments:

### hierarchyRootId

The root record of the hierarchy. Specify the **Item No.** (AR:0) of the root item of the product hierarchy you want to display.

Alternatively you can specify a generic root record. For further details, see [hierarchyRootUid](#).

### hierarchyInfoAreaId

The info area of the hierarchy. Currently only AR is supported.

You must specify `hierarchyInfoAreaId` if you use `hierarchyRootId`.

### hierarchyRootUid

You can specify the uid of the root of the hierarchy instead of `hierarchyRootId` plus `hierarchyInfoAreaId`: `Record` OR `Linked`.

Use `hierarchyRootUid` to implement one `HierarchicalSerialEntry` for multiple item structures e.g. in a process.

### hierarchyNodeFilter

Allows you to filter the tree view. The filter must be defined for the same info area as determined by `hierarchyInfoAreaId` OR `hierarchyRootUid`.

## Example

Filtering the item tree and list for the above example using the **Item Structure** flag (AR:52). Precondition: The **Item Structure** flag is set for all AR records that are parent nodes and disabled for AR records representing actual items.

The tree displays only parent nodes, the list displays the actual items for the currently selected node.

Items of Offer No. 214000032

Deselect

All Overview Last Entries

Item Item No., Item name Product Group + v

+ Add more ...

Max. 5 records per page

Start Search

I'm done

Total Quantity 12,00

Net Price 0

Item No.	Item name	Quantity	Selling Price	Discount (%)
b402-11	50"	4,00		
b402-12	52"	2,00		
b402-13	54"	1,00		
b403-11	Gearshift LX	5,00		
b403-12	Gearshift XT			
		12,00	0,00	

## hierarchyNodeSortFields

Allows you to specify the sort order for the item master tree. By default, the items in the tree are sorted by serial number (**AR-SerNo**).

Specify one or more fields (`<field name or number>`) from AR. Separate multiple fields by comma. Prefix a field with "-" (or "+") to define descending/ascending sorting.

**Default:** ascending.

**Example:** `hierarchyNodeSortFields = ItemName,-ItemNo`

If a field specified here does not exist in the data model, an error message is written to the client log.

## hierarchyRelationFilter

Allows you to filter the tree view by criteria defined on IS.

## hierarchyIndex

You can define up to 10 different item structures via Z5 variables (`CORE::INDIRECT-IS-LINK-TREE-0` to 9). Use the `hierarchyIndex` parameter to specify which item structure should be read and displayed. If more than one structure is found and `hierarchyIndex` is undefined, a message is displayed and the item structure with index 0 is used.

## MailMerge

Use MailMerge for creating letters from Aurea CRM.

The `MailMerge` action template is used by the menu action `A_MailMerge` in `UPDATE_DEFAULT`. It is used for creating letters from Aurea CRM records using the `u8.services.documents.mailmerge` function. For further details, see [Mail Merge Configuration](#) on page 844.

## MailMerge Action Parameters

The `MailMerge` action supports the following input arguments:

**uid**

Defines from which record the data for the merge fields in the document template is taken and to which record the resulting document(s) is/are linked.

Available values:

- `Record`: The uid of the current record
- `Linked`: The uid of the record linked to the current record

This parameter is mandatory.

**linkId**

Defines the link that is used to access the document generated and uploaded by MailMerge to the database. This ensures that rights restriction on the database does not prevent the user from accessing the document generated using MailMerge.

**saveMode**

Defines whether and how the resulting document(s) are uploaded to the Aurea CRM database.

Available values:

- `None`: Same as if left blank. The user is prompted to choose an upload action.
- `Link`: The resulting file(s) are created as a D1 record and linked to the record defined in parameter `uid` via D3 .
- `DoNotSave`: The resulting file(s) are not uploaded at all, just displayed to the user.

**Default:** Link

**formatName**

Defines the word processor format to be used for creating the letter(s). If left empty, a selection dialog is displayed allowing the user to select any available word processor format.

You can specify the name of any defined word processor format.

**formatFilter**

If `formatName` is empty, use this parameter to filter the word processor formats displayed in the format selection dialog.

You can specify any filter string and use wildcards (example - `up*`).

Specifying a filter string enables you to display only word processor formats that are suitable for the current info area, provided you use an adequate naming scheme for the format names.

**background**

If set, the whole letter creation process is done in the background and the user just receives a message of the result.

Possible values: `<empty>` | `true` | `false`

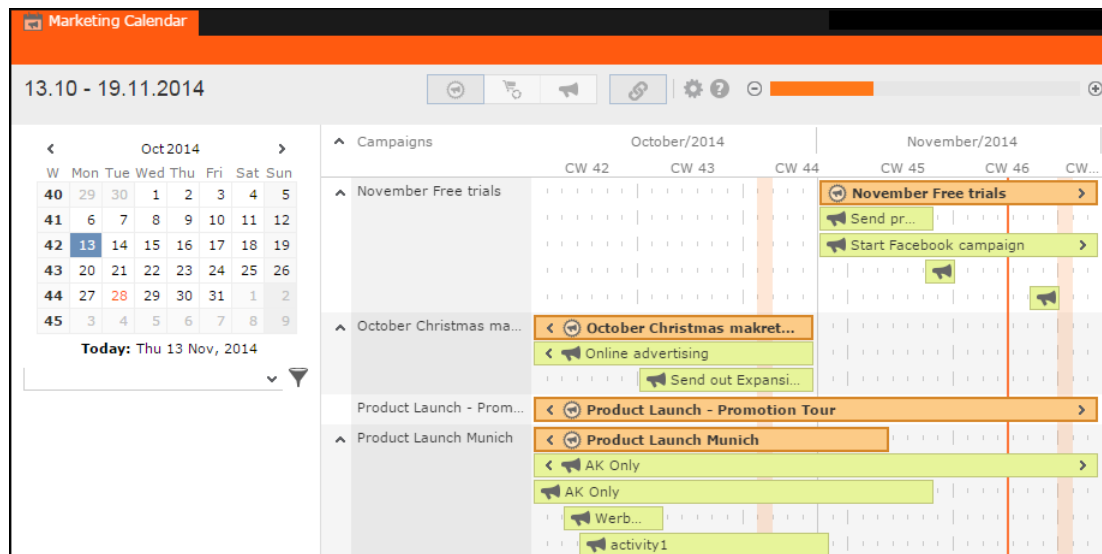
This setting overrides the **Create in Background** option in the word processor format, see Defining Word Processor Formats.

**Default:** empty.

## MarketingCalendar

Learn about MarketingCalendar action template.

The `MarketingCalendar` action template (`MarketingCalendar.aspx`) is used by the `ID_NET_MARKETINGCALENDAR` menu action (`ID_NET_CAMPAIGN_MGMT` menu) in `UPDATE_DEFAULT`.



The **Marketing Calendar** provides a calendar-like overview of campaigns (CM), marketing activities (AK) and sales cycles (VR) for the selected time frame allowing users to move and resize items.

**Note:** To implement restrictions on moving marketing activities when moving campaigns (e.g. only move marketing activities with **Status** = "Scheduled"), you need to configure the appropriate update rights.

By default, items can be edited via double-click. Marketing activities without an end date are displayed with duration = 1 day.

Campaigns and sales cycles are sorted alphabetically. The sorting of marketing activities depends on where they are displayed:

- If displayed alone, they are sorted alphabetically.
- If displayed linked to sales cycles or campaigns, they are sorted by start date. For this case custom sorting can be defined in the action. For further details, see [sortFieldAK](#).



The marketing calendar consists of the following configuration units:

- The `ID_NET_MARKETINGCALENDAR` menu action. For further details, see [Marketing-Calendar Action Parameters](#) on page 433.

`ID_NET_MARKETINGCALENDAR` is assigned to CRM process 1.3.1.2.1. "Marketing Calendar" (CRM process ID 700?).

- The special header `MarketingCalendarHeader` containing the following:
  - Toggle buttons to display campaigns or sales cycles with their linked marketing activities, or marketing activities only. It is configured by `Toggles(M_Marketing-CalendarViews)`.
  - A toggle button that determines whether to move linked records (i.e. marketing activities) when moving a campaign.
  - A settings button allowing the user to select the color coding scheme of the displayed items by info area or status.
  - A legend button explaining the currently active color coding. For further details, see [Legend](#) on page 315.

In addition, campaign and sales cycle items use bold text in order to distinguish them from marketing activities.

- A slider to adjust the zoom level of the calendar view.
- The items displayed in the marketing calendar use the table captions `<infoAreaId>.MarketingCalendar (fallback <infoAreaId>)`. For small items and multi-line table captions the complete table caption is displayed in the tooltip.
- The sidebar contains the date picker and the filter set button, analogous to the calendar. Default filters can be configured in the action. For further details, see [defaultFilterSet](#).

The user's settings toggle button states, color coding and filters are stored in the user's roaming settings. For further details, see [User Settings](#) on page 25.

## MarketingCalendar Action Parameters

Supported MarketingCalendar Action Parameters.

The `MarketingCalendar` action supports the following input arguments:

### createNewTab

See [createNewTab](#).

### defaultFilterSet

Name of a filter set which is applied if the user has not yet defined/selected a filter or filter set himself.

For information on the syntax, see [MarketingCalendar.FilterSets](#).

## fixedFilterNames

Allows you to specify fixed filters which are applied to the marketing calendar and cannot be changed by the user.

You can specify one filter for each supported info area (CM, AK and VR). Enter a comma-separated list of filters. Example -CM.NoTemplates,AK.NoTemplates,VR.NoTemplates. These fixed filters are defined in UPDATE\_DEFAULT to hide campaigns etc. marked as "Template".

If the user selects or defines a filter set, the criteria are combined by logical AND.

## sortFieldAK

Allows you to specify sort fields for sorting the marketing activities when displayed as children of campaigns or sales cycles.

Enter the desired field name or number. Separate multiple fields by comma. Use the prefix "+" for ascending and "-" for descending sort order. No prefix means ascending.

### Example

```
sortFieldAK = +Level,-Date
```

## Message

Learn about message action template.

Use the `Message` action template to display a message on a page.

The `Message` action template supports the following input arguments:

- **Message:** A comma-separated list of parameters with arbitrary names that is passed to the URL.

```
<param1>,<param2>,...
```

Create each of these parameters as new input arguments and enter the desired text. You can enter the text directly or reference texts from the CRM.Designer database (`TXT_<textgroup>_<number>`).

- **Style:** Use this parameter to control the layout of the message. If undefined (or set to `Normal`), the defined texts are appended to each other.

Available values: `Normal`, `BulletList`, `NumberedList`, `Table`.

## NewSiblingRecord

Learn about the `NewSiblingRecord` action template.

The `NewSiblingRecord` action template is used to create a new child record linked to the same parent as the current record without having to navigate to the parent record first. Example - create a new MA record while viewing an existing MA record (Expand and List view).

The `NewSiblingRecord` action automatically passes the required parent link information from the existing record to the new sibling record, e.g. creating a new MA starting from an MA record that is linked to a KP automatically transfers the company and person keys to the new record. To pass additional links or any other field values to the new record, define "copy fields" in the action specified in `actionName`. See example below.

In `UPDATE_DEFAULT` the `NewSiblingRecord` action is used by the `A_NewSiblingRecord` and `A_NewSiblingTicketRecord` menu actions. Two buttons use these menu actions: `NewSiblingRecord` and `NewSiblingTicketRecord`.

Example - To implement this for MA (and without passing on any additional links), you only need to add the `NewSiblingRecord` button to the MA `Expand` header.

The `NewSiblingRecord` action template supports the following input arguments:

### uid

The record for which a sibling record is created. It is the current record.

### actionName

The action that actually creates the new sibling record, e.g. `A_NewSiblingRecordExecute` which calls an `ExpandNew` action. For further details, see [Expand](#) on page 578.

### fieldGroupName

An optional field group to be used instead of the info area's default field group `<infoAreaId>`.

### linkNames

A comma-separated list of info area IDs specifying which additional link(s) should be used for the new record.

Example: `linkNames = KM,KP|PE|FI` uses the link to KM and the first link to KP or PE or FI.

---

**Note:** The specified links must be present in the **Parent Data** section of the source record's `Expand` view. For further details, see [Expand Configuration](#) on page 352 (**View Parent Relations**).

---

**Note:** Make sure the action specified in `actionName` can handle the result of `linkNames.Otherwiseuse"copyfields"(copyUid/copySourceFields/copyTargetFields)` instead to pass the desired link information to the new record (see example below).

---

### Example: New Ticket with multiple FI/KP links

Creating a sibling ticket using the out-of-the-box `A_NewSiblingTicketRecord` menu actions uses the source ticket's default link, which is the ticket's direct parent for the other company/person links as well (**End Customer**, **Contact** and **Billing**).

To use the additional links from the source ticket in the new sibling record, define copy fields in the action entered as `actionName`, e. g. `A_NewSiblingTicketRecordExecute`:

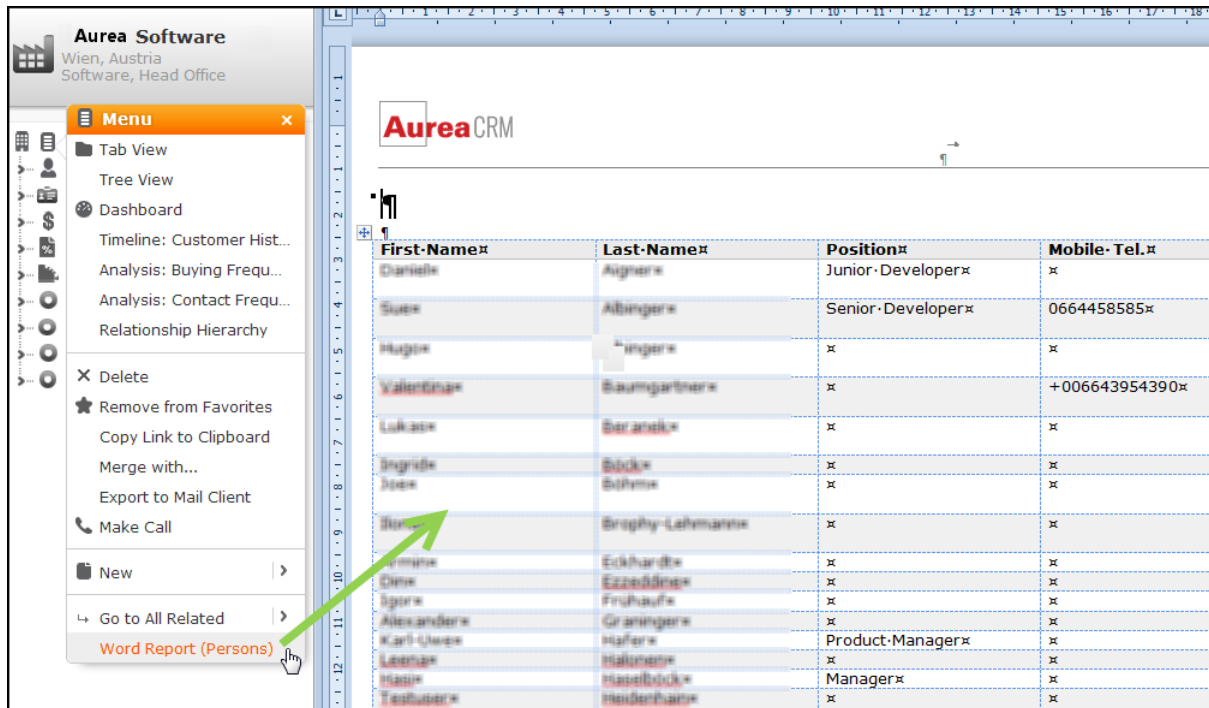
Make sure you add all required key fields (example - 81-84 for the **End Customer Company** and **Person** links) as well as the reference fields (example - 85 and 86). Adding the latter ensures that the user sees which company/person is used while creating the record. Without those fields the correct links are displayed only after saving the new record.

If you want the new record to be automatically saved, set the `autoSave` parameter to `true`. For further details, see [autoSave](#).

## QueryReport

The `QueryReport` action template can be used for actions defined for menu entries, buttons and within processes.

The `QueryReport` action template calls a JavaScript function (`u8.services.queries.executeExportFromAction`) that allows users to create a report based on a query executed in the background.



The `QueryReport` action template can be used for actions defined for menu entries, buttons and within processes.

The `QueryReport` action automatically detects parameterized queries: In this case a popup is displayed allowing the user to enter the parameter(s). For further details, see [Parameterized Queries](#) on page 500. Clicking **Start Query** in the popup executes the query using the specified parameters and creates the report as defined in the `QueryReport` action.

For general information on reports in Aurea CRM.Web, see [XML Exports & Reporting](#) on page 856.

## QueryReport Action Parameters

Supported `QueryReport` Action Parameters.

The `QueryReport` action supports the following input arguments:

### link

Defines the context for the query's execution: If set to `Record`, the current record is passed to the query as a parameter. If undefined, the query is executed without context.

### query

The name of the query to be executed in the background.

This parameter is mandatory.

## scope

Defines which kind of query is to be used if two queries with the same name exist: `Public` OR `Private`.

This parameter is supported by both `QueryRun` and `Query Designer` actions.

## reportName

The name of the transformation (\*.XSLT file) to be used for generating the report from the exported XML, e.g. `ExportListHtml`.

The `reportName` argument can contain additional export options, e.g. `ExportExpandPdf[Export.Xml.FatClientCompatible=true]`, `Format.application/pdf` to use an Aurea CRM win-compatible schema and a PDF plug-in.

For details on the syntax and all available options, see and [Export Options](#) on page 858.

If undefined, an XML export without transformation is generated.

## contentDisposition

Defines the way a report is presented:

- `inline`: The browser tries to display the report results in the current browser window.
- `attachment`: The browser asks the user to save the report results or open them in a separate window.

With `contentDisposition=inline` a `targetFrame` has to be specified for the report to be displayed in. For further details, see [targetFrame](#).

**Default:** attachment

---

**Note:** The browser's default behavior may overrule your `contentDisposition` / `targetFrame` settings. example - opening a report in a new tab instead of inline.

---

## targetFrame

The frame in which the report is displayed if `contentDisposition=inline`.

You can define the following:

- If undefined, the report is opened in the current worker window.
- If set to the name of a window, the report is opened in the respective window.
- If set to `_blank`, the report is opened in a new window or browser tab, depending on the browser settings.
- If set to `CreateNewTab`, the report is opened in a new tab in Aurea CRM.Web.
- If set to `ReUse("MyTabName")`, the report is opened in the specified tab of Aurea CRM.Web.

To display a report in a form, use the `FormFrame` widget. Use the widget's `name` parameter to specify a name for the `iframe` where you want to display the report, e.g. `"myIframe"`. Example - In the action definition, for a button on the form, define `targetFrame = myIframe`.

## caption

Defines a name for the report. The semantics of this caption depends on the transformation itself, it can be written to the meta-data of the document, used as a file name etc.

## exportFatClientCompatible

Determines the XML schema used by the export:

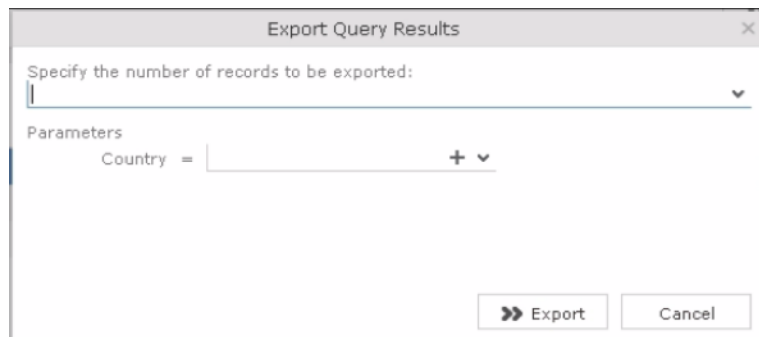
- `true`: the XML export uses a Aurea CRM win-compatible schema.
- `false`: the XML export uses the Aurea CRM.Web schema. This is the default.

## Show/Hide the Export Query Results dialog

Learn to show and hide the Export Query Results dialog.

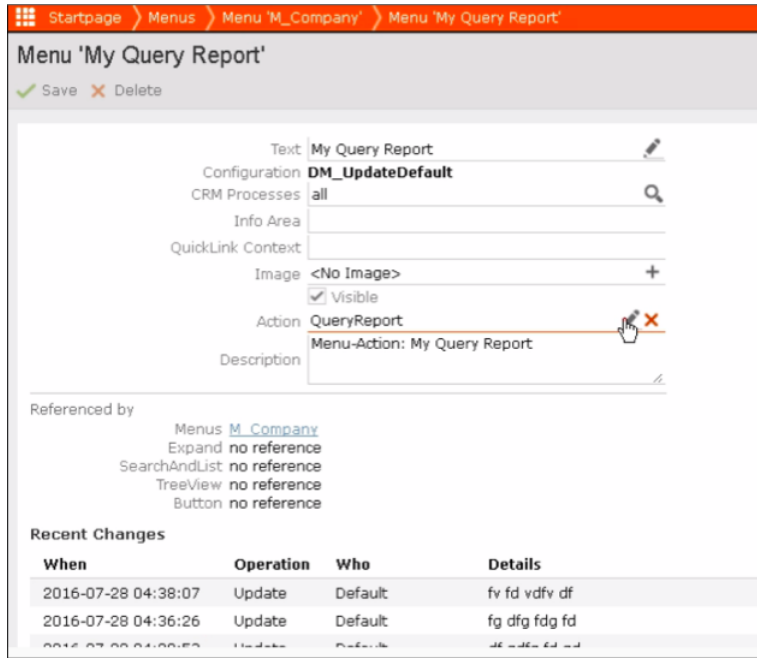
When a user exports a query result item, CRM.Web pops the **Export Query Results** dialog by default. The user can **specify the number of records to be exported** and provide other parameter values. You can use the `noUi` parameter in the `QueryReport` action template to show or hide the **Export Query Results** dialog.

The following screenshot shows the **Export Query Results** dialog.

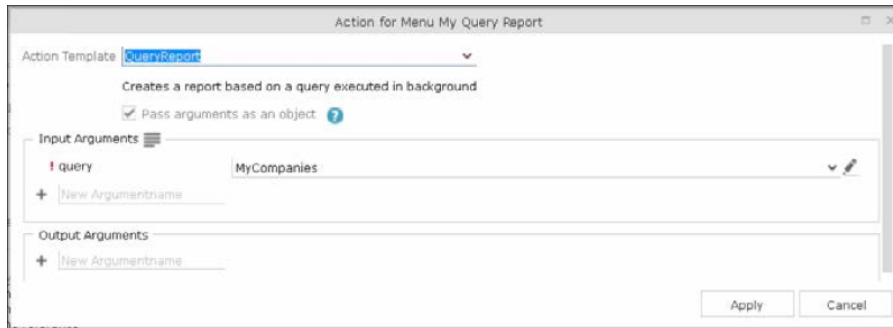


To prevent the **Export Query Results** dialog from popping up, perform the following steps in the CRM.Designer:

1. Create or open the query report menu item in the required info area. The following screenshot shows the **My Query Report** menu for the **M\_Company** infoarea.

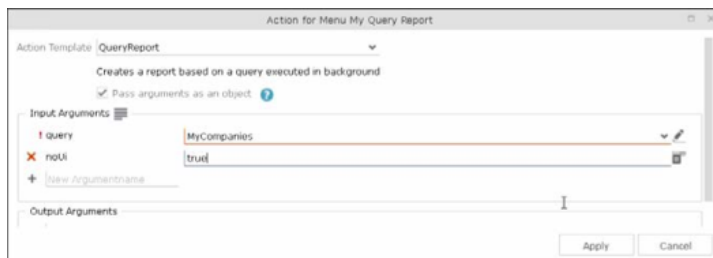


2. Select **QueryReport** action template in the **Action** field and click . The **Action for Menu My Query Report** dialog displays.



3. In the **Input Arguments** section, enter **noUi** in the text field and click **+**.

The **noUI** parameter is added to the input arguments list. Enter `true` as the parameter value.



**Note:** Entering `false` displays the **Export Query Results** dialog. This is also the default behaviour if this parameter is not defined.



4. Click **Apply** and then click **Save**.
5. Enter a description in the **Reason for Change** dialog. Click **Save**.
6. Login to CRM.Web application and click **Settings > Invalidate Cache** .
7. Run a search on the Company infoarea and from the context menu of a search result list item, select **My Query Report**.

The **Export Query Results** dialog does not show up.

## QueryRun

To create predefined queries use QueryRun action template.

The `QueryRun` action template (`queryrun.aspx`) allows users to execute a predefined query in Aurea CRM.Web.

The `QueryRun` action template can be used for context menu actions with a linked record or for application menu actions.

The `QueryRun` action automatically detects parameterized queries. In this case a form is displayed allowing the user to enter the parameter(s). For further details, see [Parameterized Queries](#) on page 500. Clicking **Start Query** executes the query using the specified parameters.

Company	Country	Postal Code	Revenue	Currency
JoeSoft AG	Austria	1160	352.148.214	
update Software AG	Austria	1040		
update Software AG	Austria	1040		

The `QueryRun` action can therefore be used as a search page for multiple info areas.

For parameterized queries containing a sub-info area without output fields, the entire sub-info area is eliminated from the query if the user does not enter at least one parameter for that info area.

The `ID_NET_QUERIES` application menu action uses the `Query Designer` action allowing users to create/edit queries in Aurea CRM.Web. The `Query Designer` action supports a subset of the `QueryRun` action's parameters as indicated below.

For both `QueryRun` and `Query Designer` actions you can define `listParams` as additional input arguments. For further details, see [ListParams](#) on page 374.

If no `listParams` are defined in the action, the global `listParams` specified in the `List.ListParams` Web Configuration parameter are applied. For further details, see [List.ListParams/List.SubList.ListParams](#).

## QueryRun Action Parameters

### Supported QueryRun Action Parameters

The `QueryRun` action supports the following input arguments:

#### **scope**

Defines which kind of query is to be used if two queries with the same name exist:  
`Public` or `Private`.

This parameter is supported by both `QueryRun` and `Query Designer` actions.

#### **query**

The name of the query.

This parameter is supported by both `QueryRun` and `Query Designer` actions.

For `QueryRun` actions this parameter is mandatory.

#### **maxRows**

Defines the maximum number of result rows per page.

**Default:** 50.

#### **form**

Defines the form displayed for parameterized queries.

**Default:** `QueryParametersTemplate`.

#### **headerGroupName**

The type of the header specified in `headerName`.

Enter the name of a header group or "Special" for an info area-independent special header.

#### **headerName**

Defines the optional header.

#### **hideContextMenu**

If set to `true`, no context menu is displayed for the result records.

**Default:** = `false`.

#### **link/linkId**

`link`: Optional parent record of the query results: `Record` or `Linked`.

---

`linkId`: ID of the link used for `link`. This parameter is only relevant if multiple links between two info areas exist in the data model (example - multiple FI-links in KM). For further details, see [Link IDs](#) on page 19.

These parameters are supported by both `QueryRun` and `Query Designer` actions.

**Default=-1**

### menu

Defines the menu displayed for the result rows. By default, this menu contains the menu of the default expand configuration of each info area contained in the query result.

### resultHeaderGroupName

The type of the header specified in `resultHeaderName`.

Enter the name of a header group or "Special" for an info area-independent special header.

### resultHeaderName

Defines the optional header displayed above the result list.

### disableDefaultAction

If set to `true`, double-clicking on a result row does not execute the default action. For further details, see [Default Action](#) on page 289.

**Default:** false.

### selectionMode

Determines if users can select multiple rows from the list and thus perform action on multiple list items at once. The following values are available:

- `Single`: Selecting multiple list items is disabled.
- `Multi`: Selecting multiple list items is enabled.

**Default:** Multi

### virtualInfoAreaReadMode

Determines whether virtual info areas are used for context menus of query result rows (`static` | `dynamic`). For further details, [virtualInfoAreaReadMode](#) on page 285.

This parameter is supported by both `QueryRun` and `Query Designer` actions.

### catalogParametersAsText

Enter a comma-separated list of catalogs (example `K2, K7`) to be using text input instead of catalog input allowing users to enter catalog values as text strings in parameterized queries to allow sub-string searches.

---

**Note: Restriction:** Only variable catalogs are supported.

---

This parameter is supported by both `QueryRun` and `Query Designer` actions. In addition, you can use this parameter for `QueryParameterForm` and `QueryResultPresenter` widgets.

## RedirectPage

Learn about the `RedirectPage` action template.

Use the `RedirectPage` action template to open a custom page with custom record data (not just a uid). Use this action when data or pages from another system are integrated into Aurea CRM.Web and, therefore, the external record key is required. Example - for displaying external web pages that require additional parameters.

The `RedirectPage` action template can be used for for actions defined for menu entries, buttons and within processes.

To prevent phishing attacks when opening an external page, you can define a black/white list in the `settings.xml` thus ensuring that a `RedirectPage` action cannot be redirected to a harmful site. For further details, see [Security Settings](#) in the *Aurea CRM.Web Installation Guide*.

## RedirectPage Action Parameters

Supported `RedirectPage` Action Parameters

The `RedirectPage` action template supports the following input arguments:

### URLPage

The address of the called page either absolute or relative to Aurea CRM.Web's application directory.

### URLParameters

The parameters passed to the page specified in `URLPage`.

The address of page called by the `RedirectPage` action is built from `URLPage` + `URLParameters` after replacing the placeholders of `URLParameters`.

You can use the following placeholders:

- up to six **records**:
  - Record #1 (`InfoArea/RecID`) with the "1st record" parameter
  - Record #2 (`InfoArea2/RecID2`) with the "2nd record" parameter
  - Record #3 (`InfoArea3/RecID3`) with the "3rd record" parameter
  - Record #4 (`ParentInfoArea4`) is a parent record of the 1st record with a given Parent info area

- Record #5 (`ParentInfoArea5`) is a parent record of the 2nd record with a given Parent info area
- Record #6 (`ParentInfoArea6`) is a parent record of the 3rd record with a given Parent info area
- **field values:**  
Use the syntax `{RecordIndex:FieldID};`  
example `{1:2};` for field number 2 of Record #1.  
Use `{RecordIndex:EXT:ExtSystemName}` to get the external key of a record for the specified external system.
- information on the **currently logged-in user:**
  - `$curRepName;` - rep name
  - `$curUser;` - login name
  - `$curTenantName;` - tenant name
  - `$curRepEmail1;` and `$curRepEmail2;` - email address of the user (ID:Email1 and ID:Email2 respectively)
  - in addition, you can use all available `$cur` expressions. For further details, see [Using Variable Expressions](#) on page 505.

Join multiple parameters with "&".

## Examples

```
{1:2};&{1:5};
```

Passes fields 2 and 5 of record #1 in the URL

```
{1:2};&$curRepName;
```

Passes fields 2 of record #1 and the name of the currently logged-in rep.

Action Template: RedirectPage	
Redirect-Page for passing any record data to pages	
<input checked="" type="checkbox"/> Pass arguments as an object ?	
Input Arguments	
URLPage	http://www.mysystem.com/
URLParameters	<b>{1:2};&amp;\$curRepName;</b>
Target	_new
InfoArea / RecID	Record

```
www.mysystem.com/?update%20software%20AG%3B&Lisa%20Simpson
```

For additional information, see the article “How to use Redirect Page” at <https://support.aurea.com>.

## Target

Determines where the page specified in `URLPage` is opened, e.g. `_new`.

## CatalogValueEncoding

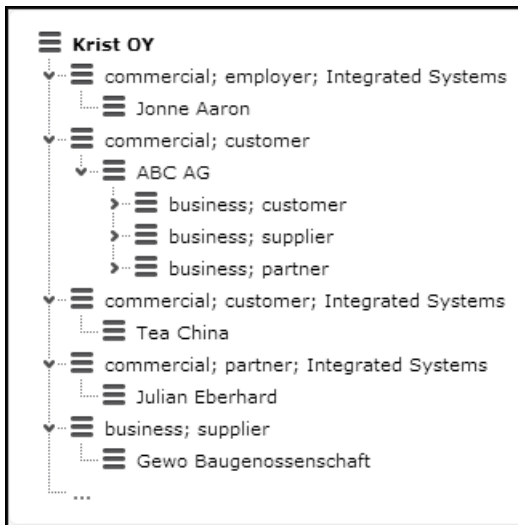
Defines how the contents of catalog fields are encoded. Available values: `Text`, `Numeric`, `TextInBaseLanguage`

**Default:** Numeric

## RelationTreeView

Use `RelationTreeView` action template to visualize the relationships between companies and persons as a hierarchical tree.

The `RelationTreeView` action template (`relationtree.aspx`) is used to visualize the relationships between companies and persons as a hierarchical tree.



In `UPDATE_DEFAULT` it is called by the `A_RelationHierarchy` (**Relationship Hierarchy**) context menu action configured in `M_Company` and by `A_RelationHierarchyKP` in `M_Person`.

The individual nodes of the relationship tree use the table caption `<InfoAreaId>.Rel`, e.g. `KP.Rel` in `UPDATE_DEFAULT`. Fallback: `<InfoAreaId>`.

By default, the recursion depth for the relationship hierarchy tree is 20. To reduce the number of levels displayed in the relationship tree (example - for performance reasons), create and set the `RelationshipHierarchy.MaxRecursionDepth` Web Configuration parameter. For further details, see [RelationshipHierarchy.MaxRecursionDepth](#). To specify the number of loaded and expanded levels for individual relationship trees, use the `ExpandLevel` action parameter. For further details, see [ExpandLevel](#).

---

**Note:** The relationship hierarchy is currently not supported for PE.

---

---

## RelationTreeView Action Parameters

Supported RelationTreeView Action Parameters.

The `RelationTreeView` action supports the following input arguments:

### CheckRights

If set to `true`, the user's access rights are taken into account when displaying the relationship hierarchy.

Nodes that are locked due to the user's access rights can not be expanded and no context menu is displayed. For such nodes text no.7 from the `relationship_text` text group is displayed.

**Default:** `false`.

### ExpandLevel

Determines how many levels of the relationship tree are initially loaded and expanded. Deeper levels are retrieved from the server and expanded, when the user clicks on a branch.

`ExpandLevel = -1` loads and expands the entire tree upon opening the page.

`ExpandLevel = 0` displays the tree completely collapsed.

**Default:** `-1`

### ExpandFI

Setting this to the custom expand results in the Company (FI) infoarea's related entity to be displayed only when the relationship entity node is selected in the relationship tree heirarchy view.

---

**Note:** If this is not set, then on opening the relationship view all the related entities are displayed.

---

### ExpandKP

Setting this to a custom expand, results in the PersonInCompany (KP) infoarea's related entity to be displayed only when the relationship entity node is selected in the relationship tree hierarchy view.

---

**Note:** If this is not set, then on opening the relationship view all the related entities are displayed.

---

### ExpandPB

Setting this to a custom expand, results in the Relationship (PB) infoarea's related entity to be displayed only when the relationship entity node is selected in the relationship tree heirarchy view.

**Note:** If this is not set, then on opening the relationship view all the related entities are displayed.

---

### InfoArea / RecID

The root record of the hierarchy tree: `Record` or `Linked`.

### MenuFI

Determines the context menu used for company records.

**Default:** `M_Company`.

### MenuKP

Determines the context menu used for person records.

**Default:** `M_Person`.

### MenuPB

Determines the context menu used for relationship records.

**Default:** `M_Relationship`.

### Reciprocal

If set to `true`, reciprocal relations are displayed as well.

**Default:** `false`.

### RelationTreeType

Determines the type of relationship displayed in the tree. Available values: `CompanyOverview`, `CompanyRelated` and `PersonRelated`.

This parameter is mandatory.

**Default:** `CompanyOverview`

---

**Note:** `PersonRelated` is only supported if the root record is a person record (KP or PE).

---

## ResourcesView

Use `resourceview` action template to display activities for selected resources.

The `ResourcesView` action template (`resourcesView.aspx`) is used by the menu action `ID_NET_CALENDAR` in `UPDATE_DEFAULT (Collaboration > Resources)`.



The **Resources** page displays activities (MA) and absences (A3) for the selected reps. Reps of type "Employee" and "Resource" are listed according to the selected filter/rep set and group. Max. 50 reps can be displayed. If the user's selection contains more than 50 reps, a message is displayed.

Items can be rescheduled and reassigned using drag&drop.

Working hours are displayed as follows:

- Working hours are defined globally via the `Calendar.DayStartsAt` and `Calendar.DayEndsAt` Web Configuration parameters.
- Weekends are grayed out by default.
- To define individual working hours for reps: In the rep's ID record check **Individual work hours** (ID:105) and enter the rep's working hours in the fields ID:29 to ID:56.

Entering no individual working hours for a specific weekday creates a day off, i.e. this day is grayed out for this rep.

If you enter individual working hours on a Saturday or Sunday, weekends are not grayed out for this rep.

- Individual working hours overrule the global working hours defined via `Calendar.DayStartsAt` and `Calendar.DayEndsAt`.

The user's settings, rep set, filter set, grouping criterion and date range, are stored client-independently using the settings service.

Items that would be hidden when selecting a large time scale, can be displayed as small dots by checking the `ResourceView.ShowHiddenItems` Web Configuration parameter. For further details, see [ResourceView.ShowHiddenItems](#). Users can change this default by setting/removing the **Show hidden items** flag in the Resource View Settings.

## Configuration Units

Learn about the configuration unit of ResourcesView action templates.

The **Resources** view consists of the following configuration units:

- The menu action `ID_NET_CALENDAR` with action `ResourcesView`. For further details, see [ResourcesView Action Parameters](#) on page 450.
- The sidebar contains the date picker, drop-down lists for filter and rep sets analogous to the calendar and the **Group reps by** drop-down list.

You can create a Web Configuration parameter `ResourceView.FilterSets` to offer predefined filter sets.

You can customize the **Group reps by** drop-down list via the `RepGroupingOptions` action parameter. For further details, see [RepGroupingOptions](#).

- The items displayed in the Resources view use the table captions `MA.ResourceView` and `A3.ResourceView`. For small items and multi-line table captions the complete table caption is displayed in the tooltip.
- Context menus: `M_ResourcesRepMenu`, `M_ResourcesRepMenu.MultiSelect`

## ResourcesView Action Parameters

Supported ResourcesView Action Parameters.

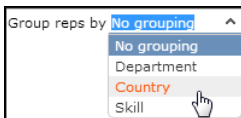
The `ResourcesView` action supports the following input arguments:

### createNewTab

See [createNewTab](#) in [Expand Action Parameters](#) on page 356.

### RepGroupingOptions

Defines the content and initial value of the **Group reps by** drop-down list using JSON.



Default configuration:

```
{
  "items":
  [
    {"textId": "TXT_net_resourceplan_text_3"},
    {"text": "Department", "key": 1, "infoAreaId": "ID", "fieldIds": [2],
"tableCaption": "IDGroup"},
    {"text": "Country", "key": 2, "infoAreaId": "ID", "fieldIds": [13]},
    {"text": "Skill", "key": 3, "infoAreaId": "SI", "fieldIds": ["ItemName",
"ProblemGroup"]}
  ],
  "selectedIndex": 0
}
```

## Syntax

"items" defines the entries of the drop-down list.

"selectedIndex" defines the initial value.

The following optional properties are supported:

Property	Description
text	Label of the list entry.
textId	Text group and ID of the language-dependent text, see <a href="#">Texts</a> on page 50.
key	The unique key used by the drop-down list to restore the selected value when returning to the <b>Resources</b> page. If used, <code>infoAreaId</code> and <code>fieldIds</code> must be defined.
infoAreaId	The info area of the fields used for grouping. If used, <code>key</code> and <code>fieldIds</code> must be defined.
fieldIds	Field IDs or XML names used for grouping. If used <code>key</code> and <code>infoAreaId</code> must be defined.
tableCaption	The table caption used to display the rep groups. If undefined, the caption is assembled from the defined <code>fieldIds</code> in the defined order.

**Note:** If you want to reset `RepGroupingOptions` to undefined, you cannot do this in the action definition: You need to delete the definition from the action template itself.

## ForceCurrentRep

If set to `true`, the **Resource** page is opened showing only the current rep instead of the previously selected rep set.

Use this parameter when displaying a Resource view on a form, e.g. on a start page.

**Default:** false

## showNewButton

See [showNewButton](#).

## Displaying Record Lists/Details for a Resource View

Use forms to display a record list/record details for items selected in a Resource View or a Dispatching Dashboard.

The screenshot shows a CRM interface with a 'Reps' resource view for the year 2013. The view displays a calendar grid with activity records for various reps. A specific activity record is expanded, showing details for '01.10.2013 12:06: Visit for Gewo Baugenossenschaft' assigned to 'Simon Eder'.

Activity Record Details:

- Subject: Gewo Baugenossenschaft, Simon Eder
- Type: Visit
- Status: Scheduled
- Starts on: 01.10.2013
- Ends on: 02.10.2013
- Starts at: 12:06
- Ends at: 12:06
- Priority: ★★★
- Create To-Do:
- Result:
- Private:

Activity Record List:

Rep ID	Required	Optional	For Information
Simon Eder	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

You can display a record list/record details for items selected in a Resource View or a Dispatching Dashboard by using forms.

UPDATE\_DEFAULT contains the following sample forms:

- `Example.DispatchingDashboard`
- `Example.ResourceView`

Create a `ShowForm` action calling one of these forms. For implementation details concerning the sample forms, see the article “How to use the resource view in a Form” at <https://support.aurea.com>.

Clicking on an item (i.e. an activity record) displays an Expand view of that record.

Clicking in a row (i.e. on a rep) displays a list of activity records for that rep. For further details, see [Displaying Record Lists/Details for the Dispatching Dashboard](#) on page 420 for a screenshot. If the number of rows exceeds the page's size, define `showPaging = true` for the `RecordListBox` widget.

# SearchPhoneEmail

Learn about the SearchPhoneEmail action template.

The `SearchPhoneEmail` action template (`SearchPhoneEmail.aspx`) is used by the `ID_NET_PHONEEMAILSEARCH` menu action in `UPDATE_DEFAULT` (**Accounts > Phone & email Search**).

The results list of the Phone & email Search is configured by the List control of field group `<InfoAreaId>.EmailPhone` (example - `FI.EmailPhone`).

## SearchPhoneEmail Action Parameters

Supported SearchPhoneEmail action parameters.

The `SearchPhoneEmail` action supports the following input arguments:

Startpage > Action Templates > Action Template 'SearchPhoneEmail'

Action Template 'SearchPhoneEmail' \*

Save Delete

Callable from: Application menu

Description: Searches for a Person / Company from a phone number or an e-mail address.

URL/Function: `crm/SearchPhoneEmail.aspx`

Options:
 

- Pass arguments as object
- This action template is deprecated

Input Arguments		
Name	<code>cachedMode</code>	Type: Value - Must
Description	Sets it the List CachedMode should be used	
Name	<code>listParams</code>	Type: Value - Optional
Description	Additional List-Parameters	
Name	<code>originalNumber</code>	Type: Value - Optional
Description	Sets the search parameter 'Original Number' initially to true	
Name	<code>EmailSearch</code>	Type: Value - Optional
Description	Passes the email address when creating a custom URL	

### cachedMode

If set to `true`, lists are cached on the client. For further details, see [cachedMode](#).

**Default:** `true`

### listParams

Additional list parameters separated by semicolon: `key,value;key,value;` etc. For further details, see [ListParams](#) on page 374.

### originalNumber

If set to `true`, the **Original Number** check box is initially checked.

Default: true.

### EmailSearch

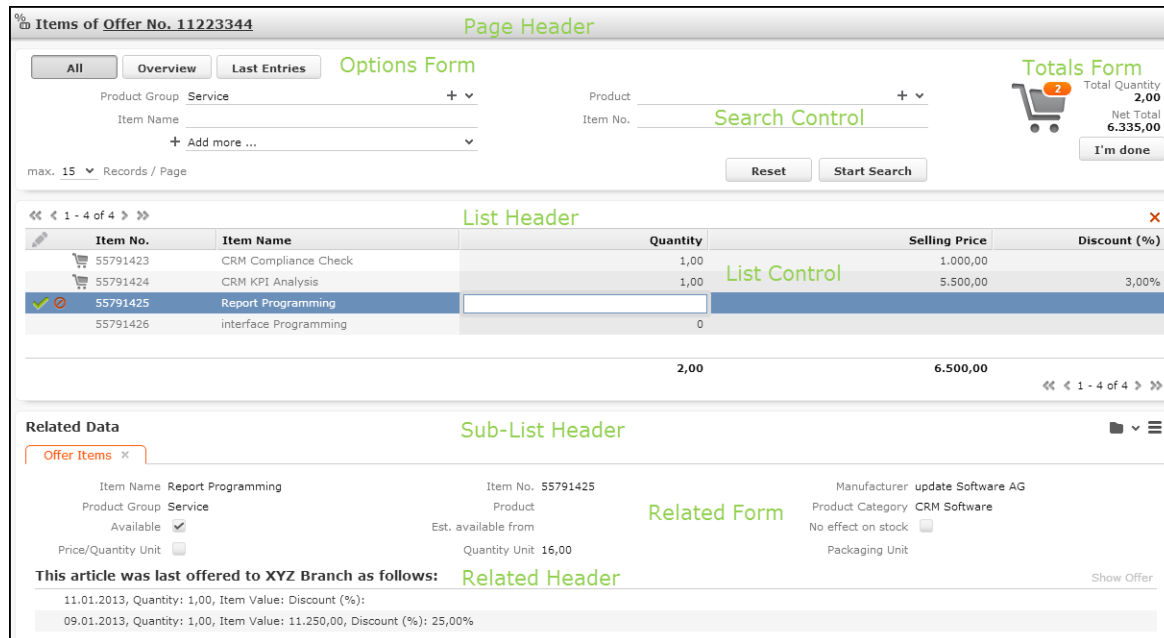
When creating a custom URL use the custom parameter `EmailSearch` to pass an email address.

The email address field is dynamically added to the form when this parameter is used.

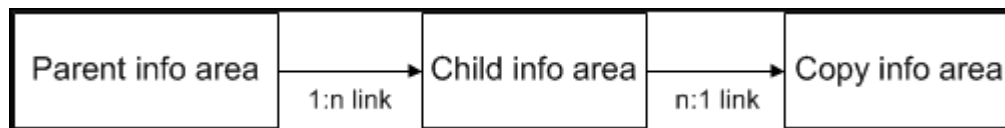
## SerialEntry

To implement quick add pages use serialEntry action template.

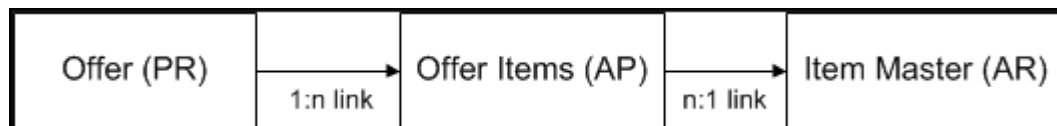
The `SerialEntry` action template (`serialentry.aspx`) allows the implementation of quick add pages. The quick add functionality provides users with the possibility to quickly create a number of child records, e.g. offer items (AP) for an offer (PR).



Quick add pages can be configured for any info area with a parent and a "copy info area" linked as follows:



Example:



In this example, the quick add functionality creates offer item records for an offer getting item information (name, number, product group etc.) from the item master (AR).

In UPDATE\_DEFAULT the `SerialEntry` action template is used for the following quick add pages:

- **BTB version:** Generic quick add configurations for **Offers** (PR) and **Orders** (AU) (called by **Quick Add (Items)** in the respective context menu):
  - `A_SerialEntryAP` in `M_Offer`
  - `A_SerialEntryUP` in `M_Order`
- **OTC version.** For further details, see [Quick Add Configurations \(OTC\)](#) on page 467.


Use the **SDK** page to test you quick add configurations: **SDK > SerialEntry**.

## Configuration Units

Learn about the configuration unit of `serialEntry` action templates.

A quick add page consists of the following configuration units:

- The `SerialEntry` action called from the parent info area's context menu. For details on the action parameters, see [SerialEntry Action Parameters](#) on page 462.
- The following headers:


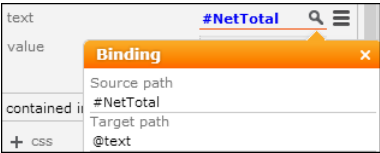
Header	Description
Page header	The quick add page's header displaying data from the parent record, example - special header <code>SerialEntry</code> from header group <code>PR</code> . For further details, see <a href="#">headerGroupName</a> .  You can configure an extended page header for this. For further details, see <a href="#">Page Headers</a> on page 334.
List header	The header of the item list containing a <b>X (Delete)</b> and (if configured) a  <b>(Duplicate Row)</b> button (see <a href="#">DuplicateRow</a> on page 310), example - special header <code>SerialEntry.List</code> from header group <code>PR</code> .

---

Header	Description
Sub-list header	<p>For the selected row in the item list a sub-list is displayed. The sub-list is defined by the parent info area's sub-list header, example - <code>SerialEntry.SubList</code> from header group PR.</p> <p>The sub-list can display a form, example - <code>SerialEntry.Offer.RelatedForm</code> (see below).</p>
Related header	<p>The header of the recent items list in the Related Data area, example - <code>SerialEntry.RelatedHeader</code> from header group PR.</p> <p>It contains a link to display the recent items's parent record, example - <b>Show offer</b>.</p>

- The following forms:



Form	Description
Options form	<p>Contains the filter buttons displayed above the search (default: <b>All, Overview, Last Entries</b>). The available filters and default selection are defined by the <code>SerialEntry.&lt;Name&gt;Filter</code> Web Configuration parameter specified in the <code>dataSource</code> component of the form. For further details, see <a href="#">SerialEntry.&lt;Name&gt;Filter</a>.</p> <p>If defined, the action parameter <code>filterType</code> overrides the initial value specified in the Web Configuration parameter.</p> <p>To completely hide the options form, set the action parameter <code>showOptions</code> to <code>false</code>.</p> <p>The <b>Last Entries</b> filter is configured by action parameter <code>recentFilterOptions</code>. For further details, see <a href="#">recentFilterOptions</a>.</p>
Totals form	<p>Defines the contents of the "Shopping Cart", example - <code>SerialEntry.Offer.Totals</code>:</p>  <p>The values are updated with every user input based on a query.</p> <p>To bind the output columns of the query to the widgets in the totals form, the XML name of the fields has to be used in a binding expression prefixed with "#".</p>  <p>A maximum number of three label-value pairs can be configured.</p> <p>The <b>I'm done</b> buttons raises a <code>ProceedOrClose</code> command. This allows a quick add page to be used inside a process (<code>Proceed</code>) as well as stand-alone (<code>Close</code>).</p>
Related form	

Form	Description																					
	<p>Defines the Related Data area of the items list, example - <code>SerialEntry.Offer.RelatedForm</code>.</p> <p>You need to specify the form in the quick add page's sub-list header (see above).</p> <div data-bbox="431 457 1187 726" style="border: 1px solid black; padding: 5px;"> <p><b>Related Data</b> <span style="float: right;">☰</span></p> <p><span style="border: 1px solid red; padding: 2px;">Offer Items</span> ×</p> <table border="0" style="width: 100%;"> <tr> <td>Item Name CRM Compliance Check</td> <td>Item No. 55791423</td> <td>Manufacturer: update Software AG</td> </tr> <tr> <td>Product Gr... Service</td> <td>Product</td> <td>Product Ca... CRM Software</td> </tr> <tr> <td>Available <input checked="" type="checkbox"/></td> <td>Est. availab...</td> <td>No effect o... <input type="checkbox"/></td> </tr> <tr> <td>Price/Quant... <input type="checkbox"/></td> <td>Quantity Unit 8,00</td> <td>Packaging ...</td> </tr> </table> <p><b>This article was last offered to XYZ Branch as follows:</b> <span style="float: right;">Show Offer</span></p> <table border="0" style="width: 100%;"> <tr> <td colspan="3">11.01.2013, Quantity: 5,00, Item Value: Discount (%):</td> </tr> <tr> <td colspan="3" style="background-color: #f0f0f0;">09.01.2013, Quantity: 1,00, Item Value: 4.050,00, Discount (%): 10,00%</td> </tr> <tr> <td colspan="3">09.01.2013, Quantity: 1,00, Item Value: 1.000,00, Discount (%):</td> </tr> </table> </div> <p>The form can display read-only details from the parent, child and copy records configured via the respective Expand configurations, example - <code>AP.SerialEntry.RelatedForm</code>.</p> <p>If an item has already been offered to the current company, these recent items are listed below the Details control.</p> <p>For implementation details, see the article “Serial Entry Related Area” at <a href="https://support.aurea.com">https://support.aurea.com</a></p>	Item Name CRM Compliance Check	Item No. 55791423	Manufacturer: update Software AG	Product Gr... Service	Product	Product Ca... CRM Software	Available <input checked="" type="checkbox"/>	Est. availab...	No effect o... <input type="checkbox"/>	Price/Quant... <input type="checkbox"/>	Quantity Unit 8,00	Packaging ...	11.01.2013, Quantity: 5,00, Item Value: Discount (%):			09.01.2013, Quantity: 1,00, Item Value: 4.050,00, Discount (%): 10,00%			09.01.2013, Quantity: 1,00, Item Value: 1.000,00, Discount (%):		
Item Name CRM Compliance Check	Item No. 55791423	Manufacturer: update Software AG																				
Product Gr... Service	Product	Product Ca... CRM Software																				
Available <input checked="" type="checkbox"/>	Est. availab...	No effect o... <input type="checkbox"/>																				
Price/Quant... <input type="checkbox"/>	Quantity Unit 8,00	Packaging ...																				
11.01.2013, Quantity: 5,00, Item Value: Discount (%):																						
09.01.2013, Quantity: 1,00, Item Value: 4.050,00, Discount (%): 10,00%																						
09.01.2013, Quantity: 1,00, Item Value: 1.000,00, Discount (%):																						

- The `SerialEntry.<Name>Filter` Web Configuration parameter defining which filters (buttons) are displayed in the options form and the initially selected filter. For further details, see [SerialEntry.<Name>Filter](#).
- The query used to calculate the content of the totals form, example - `SerialEntry.Offer.Totals`.
- The query used to calculate the recent items displayed in the Related Data area, example - `SerialEntry.Offer.RelatedForm`.
- The following field controls:

Control	Description																				
Search	<p>The Search control containing search fields from the copy info area, example - the Search control from field group <code>AR</code>. For further details, see <a href="#">searchFieldGroupName</a>.</p> <p>The search cannot be collapsed.</p>																				
List	<p>The List control defining the items list, example - the List control from field group <code>AP.SerialEntry</code>. For further details, see <a href="#">listField-GroupName</a>.</p> <p>The List control contains fields from the copy info area (example - <b>Item No.</b> and <b>Item Name</b> from AR as linked fields) and from the child info area (example - <b>Quantity</b>, <b>Selling Price</b> and <b>Discount (%)</b> from AP).</p> <p>Entering values creates the child records, example - the order items.</p> <table border="1" data-bbox="451 888 972 1018"> <thead> <tr> <th data-bbox="451 888 511 919"></th> <th data-bbox="511 888 662 919">Item No.</th> <th data-bbox="662 888 776 919">Item N...</th> <th data-bbox="776 888 972 919">Quantity</th> </tr> </thead> <tbody> <tr> <td data-bbox="451 919 511 945">🛒</td> <td data-bbox="511 919 662 945">55791423</td> <td data-bbox="662 919 776 945">CRM Compli...</td> <td data-bbox="776 919 972 945">1,00</td> </tr> <tr> <td data-bbox="451 945 511 970">🛒</td> <td data-bbox="511 945 662 970">55791424</td> <td data-bbox="662 945 776 970">CRM KPI An...</td> <td data-bbox="776 945 972 970">1,00</td> </tr> <tr> <td data-bbox="451 970 511 995">✔️</td> <td data-bbox="511 970 662 995">55791425</td> <td data-bbox="662 970 776 995">Report Prog...</td> <td data-bbox="776 970 972 995"><input type="text"/></td> </tr> <tr> <td data-bbox="451 995 511 1018"></td> <td data-bbox="511 995 662 1018">55791426</td> <td data-bbox="662 995 776 1018">interface Pr...</td> <td data-bbox="776 995 972 1018"></td> </tr> </tbody> </table> <p>These order item rows display a shopping cart image. You can change the image by adding a field control attribute <code>"ShoppingCartImage" = "te-img-myshoppingcart"</code>. The value of this attribute must be the name of a CSS sprite image.</p> <p>Clicking on the shopping cart displays a Quick View of the child record.</p> <p>The items list contains a sum totals row that automatically calculates the sum totals of the (numeric) values entered.</p> <p>Fields from the child info area are editable according to the user's access rights.</p>		Item No.	Item N...	Quantity	🛒	55791423	CRM Compli...	1,00	🛒	55791424	CRM KPI An...	1,00	✔️	55791425	Report Prog...	<input type="text"/>		55791426	interface Pr...	
	Item No.	Item N...	Quantity																		
🛒	55791423	CRM Compli...	1,00																		
🛒	55791424	CRM KPI An...	1,00																		
✔️	55791425	Report Prog...	<input type="text"/>																		
	55791426	interface Pr...																			

Control	Description
MiniDetails	<p>The MiniDetails control defining the MiniDetails for the items list, example - the MiniDetails control from field group <code>MU.SerialEntry</code> (OTC version).</p> <p>The action parameter <code>showMiniDetails</code> determines whether MiniDetails are displayed. For further details, see <a href="#">showMiniDetails</a>.</p>
Popup list	The List control defining the popup displayed when the user clicks the Shopping Cart button, e.g the List control from field group <code>AP.SerialEntry.Totals</code> .

**Note:** Some configuration units need to comply with naming conventions in order to be applied. For further details, see [Naming Conventions](#) on page 539.

For additional information, see the article “Serial Entry” at <https://support.aurea.com>.

If your item master contains product hierarchies (see [Item Structure](#) in the *Web Business Logic User Guide*), you can display these items in a hierarchical tree view to the left of the quick add page. To implement this, use the `HierarchicalSerialEntry` action template. For further details, see [HierarchicalSerialEntry](#) on page 427.

## Naming Conventions

Quick add configuration units need to follow special naming conventions.

For some configuration units these naming conventions can be overridden by specifying a name via an action parameter.

The following table lists default names and action parameters, if available, for all configuration units:

Configuration unit	Default Name	Action Parameter
Page header	<p>Header group: <code>&lt;ParentInfoAreaId&gt;</code></p> <p>Special header: <code>SerialEntry</code></p> <p>Example: <code>SerialEntry from PR</code></p>	<code>headerGroupName</code>
List header	<p>Header group: <code>&lt;ParentInfoAreaId&gt;</code></p> <p>Special header: <code>SerialEntry.List</code></p> <p>Example: <code>SerialEntry.List from PR</code></p>	<code>none</code>

Configuration unit	Default Name	Action Parameter
Sub-list header	Header group: <ParentInfoAreaId> Special header: <code>SerialEntry.SubList</code> Example: <code>SerialEntry.SubList</code> from PR	none
Related header	Header group: <ParentInfoAreaId> Special header: <code>SerialEntry.RelatedHeader</code> Example: <code>SerialEntry.RelatedHeader</code> from PR	none
Options form	<code>SerialEntry.&lt;Type&gt;.Options</code> See <a href="#">name</a> for possible values of <Type>. Example: <code>SerialEntry.Default.Options</code>	<code>optionsFormName</code>
Totals form	<code>SerialEntry.&lt;Type&gt;.Totals</code> Example: <code>SerialEntry.Offer.Totals</code>	<code>totalsFormName</code>
Totals Query	<code>SerialEntry.&lt;Type&gt;.Totals</code> Example: <code>SerialEntry.Offer.Totals</code>	none
Related form	<code>SerialEntry.&lt;Type&gt;.RelatedForm</code> Example: <code>SerialEntry.Offer.RelatedForm</code>	none
Search control	Field group: <CopyInfoAreaId> Example: Search control from AR	<code>searchFieldGroupName</code>
List control	Fieldgroup:<ChildInfoAreaId>.SerialEntry Example: List control from AP.SerialEntry	<code>listFieldGroupName</code>
Popup List control	Field group: <ChildInfoAreaId>.SerialEntry.Totals fallbacks: <ChildInfoAreaId>.SerialEntry and <ChildInfoAreaId> Example: List control from AP.SerialEntry.Totals	none

## SerialEntry Action Parameters

Supported serial entry action parameters.

The `SerialEntry` action supports the following input arguments:

### name

Defines the name or type of the quick add configuration.

### Syntax:

For generic quick add configurations enter `<name>(<copyInfoAreaId>=><childInfoAreaID>)`.

For special quick add implementations as defined in the OTC vertical specify the type by entering `<name>`.

If `name` does not contain an expression in parentheses, the definition is interpreted as a quick add `type`. The following quick add types are supported:

Default (used for generic quick add configurations)	POS	StockControl
Offer	ProductDiscussion	
Order	Sample	

### Examples:

BTB:

A quick add configuration named Offer of type Default, that creates offer items (AP) linked to items in the item master (AR):

```
name = Offer(AR=>AP)
```

OTC:

Special quick add implementations:

```
name = Offer
```

```
name = Sample
```

### uid

The quick add page's parent record. Example - the offer for which items are created.

### listFieldGroupName

The name of the field group whose List control is used for the item list.

Item No.	Item Name	Quantity	Selling Price	Discount (%)
55791423	CRM Compliance Check	1,00	1.000,00	
55791424	CRM KPI Analysis	1,00	5.500,00	3,00%
55791425	Report Programming			
55791426	interface Programming			
		2,00	6.500,00	

If undefined, the default is used. For further details, see [Naming Conventions](#) on page 539.

## filterType

Defines the initially selected filter of the options form.



The following filters are supported depending on the quick add configuration's type:

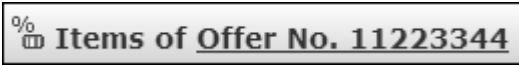
filterType	Description	Supported for types
(undefined)	no filter is applied	all
Overview	The grid displays all currently existing child records (example - order items), i.e. all items for which a quantity has been entered.  <b>Note:</b> Items without <b>Item No.</b> are not listed in this view.	Default, Offer, Order, POS, Product Discussion, Sample
Last	The grid displays items linked to the last n parent records. n is defined by <a href="#">recentFilterOptions</a> .	Default, POS
LastOrder	Items from the last n orders. n is defined by <code>recentFilterOptions</code> .	Order
Sample	Items defined as samples (info area MK).	Sample
Action	Items from a special offer (info area AT).	Order, POS, Sample
Listing	Items linked to a valid listing record (LS).	Order, POS
SalesOrganization	Items belonging to the same sales organization (via VA and VB) as the parent record.	Order

For information on the business logic involved. For further details, see [Web Business Logic User Guide](#).

`filterType` overrides the initial value specified in the `SerialEntry.<Name>Filter` Web Configuration parameter. For further details, see [SerialEntry.<Name>Filter](#).

### headerGroupName

Defines the header group for the quick add page displaying information from the parent record specified by uid.



If undefined, the default is used. For further details, see [Naming Conventions](#) on page 539.

### optionsFormName

The name of the form containing the filter options for the quick add page, example - `SerialEntry.Default.Options`.



If undefined, the default is used. For further details, see [Naming Conventions](#) on page 539.

### copySourceFields

A comma-separated list of fields (`<field name or number>`) to be copied from the source record (example - from AR) to the target (child) record (example - to AP).

Example: `defaultFields = ItemNo,ItemName,Department,ProdGrp,Product`.

### copyTargetFields

The target fields for the fields specified in [copySourceFields](#).

### searchFieldGroupName

The name of the field group whose Search control is used for item search. The info area to be search is always the copy info area, example - AR.

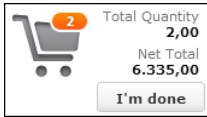


If undefined, the default is used. For further details, see [Naming Conventions](#) on page 539.

### totalsFormName

The name of the form containing the overview of currently added items (Shopping Cart), example - `SerialEntry.Offer.Totals`.





If undefined, the default is used. For further details, see [Naming Conventions](#) on page 539 .

### showOptions

If set to `false`, the options form is hidden.

**Default:** true.

### autoSearch

If set to `true`, the search is immediately executed when the user opens the quick add page.

Auto-search does not happen when the user updates a filter in the search mask. If you want auto-search to be activated when a filter is updated in the search mask, you have to set [autoApplyFilter](#) to `true`.

**Default:** True

### fixedFilter

Specifies a fixed filter which is applied to the search and cannot be changed by the user.

The filter is AND-combined with the selected filter option (Overview, Last Entries etc.).

### recentFilterOptions

Defines which items are listed when **Last Entries** is selected.

#### Syntax:

```
ParentInfoAreaId, DateField, HistoryLength
```

`ParentInfoAreaId` is the parent of the serial entries parent info area, example - FI for Offers.

If `ParentInfoAreaId` is set to "", FI is used as default.

If `ParentInfoAreaId` is set to "-", no parent link is used.

If `DateField` is set to "", a field named "Date" is used as default. If no **Date** field exists, the first date field of the info area is used.

#### Example:

```
recentFilterOptions = FI, OfferDate, 3
```

The **Last Entries** filter displays all offer items from the three most recent offers linked to the same company as the current offer. The most recent offers are determined using the **Offer Date** field.

### showMiniDetails

If set to `false`, no MiniDetails can be displayed in the item list.

**Default:** true.

### ignoredFieldsOnDuplicate

A comma-separated list of fields (<field name or number>) to be ignored when a row is duplicated. For further details, see [DuplicateRow](#) on page 310.

#### Syntax:

```
[ "<field name or number>", "<field name or number>", ... ]
```

#### Example:

```
[ "Send", "Issued", ... ]
```

Example in UPDATE\_DEFAULT (OTC): `ignoredFieldsOnDuplicate = ["Quantity"]` is defined in `A_OrderPositions` to allow for duplicating rows of orders with several delivery dates (**Order Delivery Items** info area (LP)).

### ignoredLinksOnDuplicate

A list of links to be ignored when a row is duplicated. For further details, see [DuplicateRow](#) on page 310.

#### Syntax:

```
[{ "infoAreaId": "<infoAreaId>", "linkId": <linkId> }, { "infoAreaId":  
"<infoAreaId>", "linkId": <linkId> }, ... ]
```

#### Example:

```
ignoredLinksOnDuplicate = [{ "infoAreaId": "LP", "linkId": 0 }, { "in-  
foAreaId": "SP", "linkId": 0 }]
```

Example in UPDATE\_DEFAULT (OTC): `ignoredLinksOnDuplicate = [{ "infoAreaId": "LP", "linkId": 0 }]` is defined in `A_OrderPositions` to allow for duplicating rows of orders with several delivery dates (**Order Delivery Items** info area (LP)).

### searchDefaultFieldValues

Default field values for the item search. It can be changed by the user. Use the same syntax as described for the Search action template. For further details, see [fixedFieldValues](#).

The info area to be searched (example - AR) must be explicitly specified in `searchInfoAreaId`, otherwise `searchDefaultFieldValues` is ignored.

### searchFixedFieldValues

Fixed field values for the item search. It can not be changed by the user. Use the same syntax as described for the Search action template. For further details, see [fixedFieldValues](#).

The info area to be searched (example - AR) must be explicitly specified in `searchInfoAreaId`, otherwise `searchDefaultFieldValues` is ignored.

### searchInfoAreaId

The info area of the item search (example - AR).

You must specify `searchInfoAreaId` if you want to define default/fixed search values in `searchDefaultFieldValues` / `searchFixedFieldValues`.

### sourceCopyFieldGroupName

The name of the field group used to define a dynamic filter for the item search. The filter values are defined as functions in the List control. For further details, [Dynamic Filters](#) on page 509.

The record from which the values are read is defined by `uid`.

---

**Note:** Dynamic filters are only supported by generic quick add configurations.

---

### disableFilter

Add this argument and set it to `true` to disable the filter selection and definition functionality for the item search.

### showRelatedView

Add this argument and set it to `false` to hide the Related Data area.

## Quick Add Configurations (OTC)

This topic has information about quick add pages.

The OTC vertical contains quick add pages for the following info areas:

Parent	Child	Menu Action	Menu
Offer (PR)	Offer Item (AP)	<code>A_OfferSerialEntry</code>	<code>M_Offer</code>
Order (AU)	Order Item (UP)	<code>A_OrderPositions</code>	<code>M_Order</code>
Refunding (VG)	Refunding Item (VP)	<code>A_RefundingSerialEntry</code>	<code>M_Refunding</code>
Activity (MA)	ProductDiscussion Item (BG)	<code>A_MeetingSPP</code>	<code>M_Appointments</code>

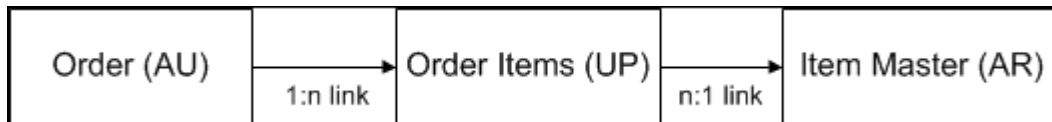
Parent	Child	Menu Action	Menu
POS Monitoring (PS)	POS Monitoring Item (PP)	A_POSSerialEntry	M_POS
Company (FI)	Promotional Material (WM)	A_PromotionalMaterialSerialEntry	M_Company, M_PromotionalMaterial
Person (PE)	Sample (MU)	D_Sample	M_Appointments
Stock Control (BV)	Stock Control Item (BP)	A_StockSerialEntry	M_StockControl
Installed Base (IB)	Installed Base Item (IP)	A_InstBaseSerialEntry	M_InstBase

For information on the involved business logic (**Order, Offer, POS, Samples**, etc.). For further details, see [Web Business Logic OTC User Guide](#).

The quick add pages for **Refunding, Promotional Material** and **Installed Base** use the generic implementation as described above. The following topics describe some of the special quick add configuration types implemented in the OTC vertical.

### Order

Learn about the serial entry action template - Order.



The `SerialEntry` action having `name = Order` is called via `A_OrderPositions` (**Quick Add (Order Items)**) in `M_Order`.

---

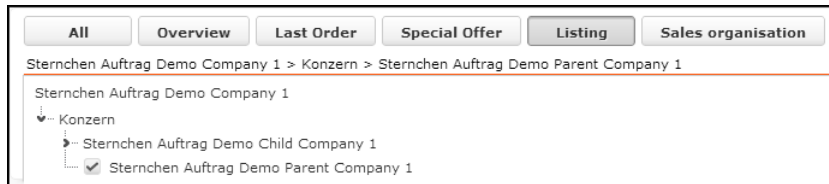
**Note:** The Offer quick add page is configured analogously with configuration units for PR and AP.

---

### Configuration Units

The Order quick add page consists of the following configuration units:

- Page header `SerialEntry` of header group `AU`
- Options form `SerialEntry.Order.Options`



The available filter buttons and default selection are defined via the `SerialEntry.OrderFilter` Web Configuration parameter. For further details, see [SerialEntry.<Name>Filter](#).

If **Listing** is selected, the company relationship hierarchy is displayed in a drop-down list below the filter buttons allowing users to select the desired listing path. The **Inheritance - Quick Add** configuration parameter (**Listing** category) in the **Configuration** info area (MC) determines which relationship type is used. For further details, see [Configuration Info Area](#) in the *CRM.Core Administrator Guide*.

For more information on listings (**inheritance**, **external system**, etc.). For further details, see [Web Business Logic OTC User Guide](#).

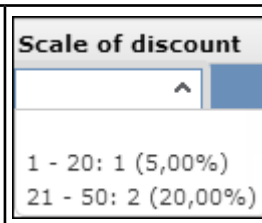
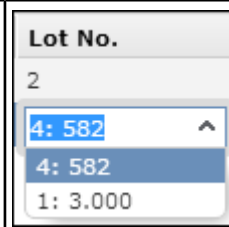
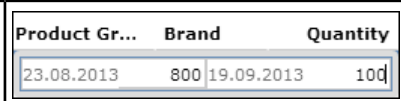
For orders that are directly linked to a special offer (AT), i.e. without a marketing activity, the options form (and Web Configuration parameter) `SerialEntry.Order.Options.SpecialOffer` is used. Only the **Overview** and **Special Offer** filter buttons are available in this case.

- Totals form `SerialEntry.Order.Totals` (shopping cart)
- Search control from field group `AR.SerialEntry`

**Note:** The business logic determines which fields are copied from AR to UP. Therefore the `copySourceFields` and `copyTargetFields` action parameters are not used for the Order quick add page.

- List header `SerialEntry.List` from header group `AU` containing a **X (Delete)** and **🔄 (Duplicate Row)** button. For further details, see [DuplicateRow](#) on page 310. When duplicating UP rows all linked delivery items (LP) are copied as well.
- List control from field group `UP.SerialEntry` defining the items list and the popup list:

Item No.	Item Name	Product...	Br...	Quantity	End Price	Discount in...	Discount	Disc.Value2	Scale of discount	Lot No.
000111001	MeinDiscloseAR			25						
0010	MeinPrioAuftrag									
110	Packpapier									

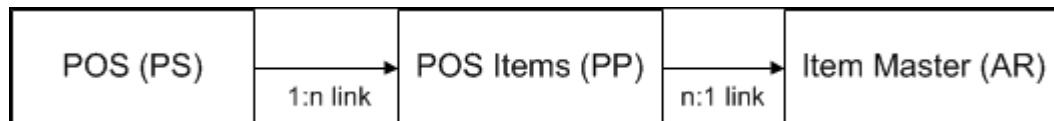
<p>For the <b>Scale of discount</b> field a special widget is used, displaying the available volumes and discounts in a drop-down list as <b>&lt;Volume From&gt; - &lt;Volume Up To&gt;: &lt;Discount vol. in kind&gt; (&lt;Discount&gt;%)</b>. The values are taken from the <b>Special Offer Quantity Scale</b> info area (AM). The selected value (<b>Volume From</b>) is written to the <b>Quantity</b> field. To disable this functionality set the <b>Do not apply scale of discount</b> configuration parameter (<b>General Settings</b> category) in the <b>Configuration</b> info area (MC).</p>	
<p>For the <b>Lot No.</b> field a special widget is used, displaying the available lots and their quantities in a drop-down list as <b>&lt;LotNo.&gt;: &lt;Quantity&gt;</b>.</p>	
<p>For the <b>Quantity</b> field (UP:20) the widget <code>SerialEntryDeliveriesValues</code> is used. It is configured via the <code>Widget</code> field attribute. If multiple <b>Delivery</b> records (LD) are linked to an order, the <b>Quantity</b> field displays a field for each delivery date.</p>	

To use the same widget for the **Discount vol. in kind** field (UP:139) as well, add it to the `UP.SerialEntry` List control and define `Widget = SerialEntryDeliveriesValues` for that field.

- MiniDetails control from field group `UP.SerialEntry`
- The Related Data area is defined by `SerialEntry.Order.RelatedForm`, `SerialEntry.RelatedHeader` of header group `AU` and the `SerialEntry.ShowOrder` button.

## POS

Learn about the serial entry action template - POS.

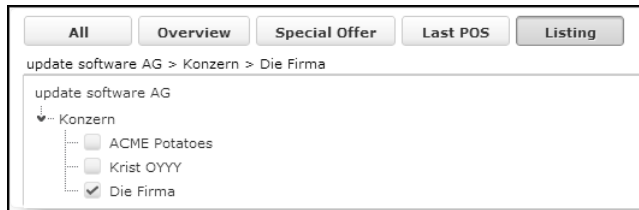


The `SerialEntry` action (with `name = POS`) is called via `A_POSSerialEntry` (**Quick Add (Items)**) in `M_POS`.

## Configuration Units

The POS quick add page consists of the following configuration units:

- Page header `SerialEntry` of header group `PS`
- Options form `SerialEntry.POS.Options`



The available filter buttons and default selection are defined via the `SerialEntry.POSFilter` Web Configuration parameter. For further details, see [SerialEntry.<Name>Filter](#).

If **Listing** is selected, the company relationship hierarchy is displayed in a drop-down list below the filter buttons allowing users to select the desired listing path. The **Inheritance - Quick Add** configuration parameter (**Listing** category) in the **Configuration** info area (MC) determines which relationship type is used, see [Configuration Info Area](#) in the *CRM.Core Administrator Guide*.

For more information on listings (**inheritance**, **external system** etc.), see [Web Business Logic OTC User Guide](#).

- Totals form `SerialEntry.POS.Totals`
- Search control from field group `AR.SerialEntryPOS`.

**Note:** The business logic determines which fields are copied from LS and AT to PP. From AR additional fields (`ItemNo`, `ItemName`, `Segment`, `ProdCat`, `Brand`, `Grade`) are copied via the `copySourceFields` and `copyTargetFields` action parameters.

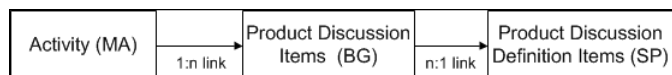
- List header `SerialEntry.List` from header group `PS` containing a **X (Delete)** and **📄 (Duplicate Row)** button. For further details, see [DuplicateRow](#) on page 310.
- List control from field group `PP.SerialEntry` defining the items list and the popup list:

Item No.	Item Name	Stock	Facings	Facing %	Placing 1	Price
2	Artikel 2	500	5	13%	Rack	5,00
12341234	Vitamin C	1.000	15	100%		3,00
455677	Multi-Vitamin	1.000	10	75%		5,00
LSI 100202	Anti Pain Capsules 20 mg	500	3	50%		10,00

- MiniDetails control from field group `PP.SerialEntry`
- The Related Data area is defined by `SerialEntry.POS.RelatedForm`, `SerialEntry.RelatedHeader` of header group `PS` and the `SerialEntry.ShowPOS` button.

## Product Discussion

Learn about the serial entry action template - Product Discussion.

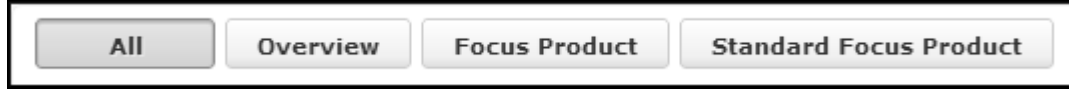


The `SerialEntry` action (with `name = ProductDiscussion`) is called via `A_MeetingSPP (Quick Add (Product Discussions))` in `M_Appointments`.

## Configuration Units

The Product Discussion quick add page consists of the following configuration units:

- Page header `SerialEntry` of header group `BG`
- Options form `SerialEntry.ProductDiscussion.Options`



The **All** and **Overview** buttons and default selection are defined via the `SerialEntry.ProductDiscussionFilter` Web Configuration parameter. For further details, see [SerialEntry.<Name>Filter](#).

Additional filter buttons are automatically created based on the values of the fixed catalog **Type** used in the **Product Discussion Definition** info area (SR). Per default, this catalog contains the values **Focus Product** and **Standard Focus Product**. If you create additional custom values for the **Type** catalog, the values are displayed in the options form as additional filter buttons.

Each of these filter buttons loads the SP records linked to an SR record of the selected **Type**.

- Totals form `SerialEntry.ProductDiscussion.Totals`
- Search control from field group `SP.SerialEntry`

---

**Note:** The copy fields are configured via the `copySourceFields` and `copyTargetFields` action parameters.

---

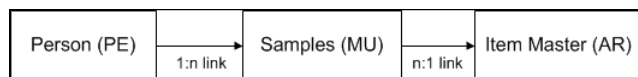
- List header `SerialEntry.List` from header group `BG` containing a **X (Delete)** button.
- List control from field group `BG.SerialEntry` defining the items list and the popup list:

Item No.	Item Name	Bra...	Product Group	Number / Quantity	OP prescr.practice	OP reason	OP image	OP agreement
12	Nape12	Mars	schoki		+ v			
111	Kopierpapier	Office	Papier					
110	Packpapier	Office	Papier					
4444	Dinosaurierpa...							
455677	Multi-Vitamin							

- No MiniDetails are configured for the Product Discussion quick add page.
- The RelatedData area is defined by `SerialEntry.ProductDiscussion.RelatedForm`, `SerialEntry.RelatedHeader` of header group `BG`, and the `SerialEntry.ShowActivity` button.

## Sample

Learn about the serial entry action template - POS.



Sample (MU) records are linked to persons (PE). The item master (AR) is the copy info area.



The `SerialEntry` action (with `name = Sample`) is called via `D_Sample` (**Quick Add (Samples)**) in `M_Appointments` from person-related activity records.

## Configuration Units

The Sample quick add page consists of the following configuration units:

- Page header `SerialEntry` of header group `MU`
- Options form `SerialEntry.Sample.Options`

The screenshot shows a header area with three buttons: 'Overview', 'Samples' (which is selected), and 'Special Offer'. Below the buttons, it displays 'Business Area BA1' and 'Date 29.01.2013'.

The available filter buttons and default selection are defined via the `SerialEntry.SampleFilter` Web Configuration parameter. For further details, see [SerialEntry.<Name>Filter](#).

In addition the options form contains the fields **Business Area** (AZ:1) and **Starts at** (MA:0).

Totals form `SerialEntry.Sample.Totals` (shopping cart)

- Search control from field group `AR.SerialEntry`

---

**Note:** The business logic determines which fields are copied from AR to MU. Therefore the `copySourceFields` and `copyTargetFields` action parameters are not used for the Sample quick add page.

---

- List header `SerialEntry.List` from header group `MU` containing a **X (Delete)** and **(Duplicate Row)** button. For further details, see [DuplicateRow](#) on page 310.

When duplicating MU rows the values for **Issued** and **Send** are not copied.

- List control from field group `MU.SerialEntry` defining the items list and the popup list:

Item No.	Item Name	Issued	Send	Available Quota	Price	Curre...	Stock	Lot No.
999888	Seidenpapier	10	10	484			Bestand 1	2
110	Packpapier	18		*			Bestand 1	4

The **Available Quota** field is recalculated based on the definition in the **Sample Code** info area (MK) every time the user enters quantities in **Issued** or **Send**.

Use the **Quota exceeded** configuration parameter (**Sample** category) in the **Configuration** info area (MC) to determine whether users can exceed the available quota, see [Configuration Info Area](#) in the *CRM.Core Administrator Guide*. If **Quota**

**exceeded** is unchecked, values that exceed the available quota cannot be saved and a message is displayed.

For the **Lot No.** field a special widget is used, displaying the available lots and their quantities in a drop-down list as <LotNo.>: <Quantity>.

If the quantities should not be displayed, set the **Lots without quantities** configuration parameter (**Inventory** category) in the **Configuration** info area (MC).

**Lot No.**

2

4: 582

4: 582

1: 3.000

- MiniDetails control from field group `MU.SerialEntry`:

Item No.	Item Name	Issued	Send	Available Quota	Price	Curre...	Stock	Lot No.
999888	Seidenpapier	10	10	484			Bestand 1	2
110	Packpapier	18		*			Bestand 1	4

Item No. 110	Item Name Packpapier	Date 17.01.2013
Rep Christian Juff	Item No. 110	Item Name Packpapier
Sector SalesGroup A	Product Group Papier	Brand Office
Grade 120 Gramm	Department Metallwaren	Issued 18
Status	Currency	Costs
Lot No. 4	Activity Telephone 17.01.2013	Special Offer ...

- No Related Data area is configured for the Samples quick add page.

## Action Parameters

Learn about serial entry action parameters.

Use/create the following additional action parameters to control which items are listed on the quick add page:

### ShowZeroQuota

If set to `false`, items with **Available Quota**  $\leq 0$  are hidden from the list. If set to `true` and MC configuration parameter **Quota exceeded** is checked, items with **Available Quota**  $\leq 0$  are listed.

### ShowZeroStock

If set to `false`, items for which the current rep has 0 or negative stock for the respective lot numbers are hidden from the list. Thus lots with 0 available can not be selected for creating new MU records.

### ShowNoStock

If set to `false`, items for which the current rep has no stock record for the respective lot numbers are hidden from the list. If set to `true`, the list displays all available items regardless of their lot numbers and the current rep's actual stocks.

**Default:** `true`

---

**Note:** These settings apply to the **Sample** and **Special Offer** views only. The **Overview** and the totals form (shopping cart) always list all currently existing MU records.

---

For details on sample quotas and stock control, see [Web Business Logic OTC User Guide](#).

## Timeline

Use the timeline action template to visualize customer history.

The `Timeline` action template (`timeline.aspx`) allows users to visualize a customer history, i.e. to display a specific set of child records (activities, offers etc.) e.g. for a selected company along a time line.

Examples in `UPDATE_DEFAULT`: `A_AccountTimeline` (in `M_Company` and `M_Person`), `A_MarketingTimeline` (in `M_Person`).

The type of child records and their graphic representation (icons, borders) are determined via the Timeline configuration referenced in the action. For further details, see [Timelines](#) on page 538.

## Timeline Action Parameters

Supported timeline action parameters.

The `Timeline` action supports the following input arguments:

### uid

The uid of the timeline's parent record: `Record` or `Linked`.

This record uid is used as a link record when querying for the records to be shown on the timeline.

### name

Name of the Timeline configuration. For further details, see [Configuring a Timeline](#) on page 540.

### zoom

The initial zoom level of the timeline. Available options: `month`, `quarter`, `year`, `all`.

Users can zoom in/out via arrow buttons and scroll wheel. The maximum zoom level is 10 years, minimum 7 days.

**Default:** `quarter`

## Trigger

The Trigger action template allows you to call a trigger via menu entries, buttons and within processes.

The `Trigger` action template calling the `javascript:u8.services.businessObjects.executeTrigger` function allows a non-interactive trigger to be executed in Aurea CRM.Web. For further details, see [Executing Non-Interactive Triggers in Aurea CRM.Web](#) on page 476.

The `Trigger` action template allows you to call a trigger via menu entries, buttons and within processes. To execute a trigger for multiple records selected in lists or sub-lists, either add the trigger as a button or as a menu action in the corresponding `M_MultiSelect` context menu. For further details, see [M\\_MultiSelect](#).

For detailed information on triggers and how to define them. For further details, see [Triggers](#) in the *CRM.Core Administrator Guide*.

For more information on executing triggers in Aurea CRM.Web, see the article “How to execute Triggers” at <https://support.aurea.com>.

## Executing Non-Interactive Triggers in Aurea CRM.Web

Learn to execute Non-Interactive Triggers in Aurea CRM.Web.

For the trigger actions "Add New", "Edit/Update" and "Delete", all affected records are returned. If for example a trigger deletes a company, the result contains the deleted company plus all dependent records that were deleted as well. The result also contains records that were modified by follow-up triggers. The result can be processed by a JavaScript function. All other trigger actions do not return a result since they do not modify records.

The following triggers/trigger actions are supported in Aurea CRM.Web:

Trigger	Supported in Aurea CRM.Web?	Comment
Show	No	
Add New	Yes	Affected records are returned as result.
Edit/Update	Yes	Affected records are returned as result.
Delete	Yes	Affected records are returned as result.
Program Call	Yes	Attention: The program call happens on the server, the program which is called runs directly in the process of IIS (without user interface and interaction possibilities). Thus, if you use this kind of trigger you have to make sure that the program you call can run fully automated without a user interface.  No result is returned.

Trigger	Supported in Aurea CRM.Web?	Comment
Single/Serial Letter	No	For creating letters in Aurea CRM.Web. For further details, see <a href="#">Letters &amp; Mail Merge</a> on page 843.
Default Values	Yes	No result is returned.
Web-Level	No	
email	Yes	No result is returned.
SMS	Yes	No result is returned.
Plausibility Check	Yes	<p>Since plausibility check triggers run automatically when the user leaves a field during editing, it is not necessary to use the <code>Trigger</code> action template for plausibility check triggers.</p> <p>You can call such triggers via the <b>Rights</b> definition instead. For further details, see <a href="#">Triggers</a> in the <i>CRM.Core Administrator Guide</i>. You need to define plausibility check triggers for specific fields (i.e. in the <b>Upd Fields</b> column) for them to be executed automatically when a user leaves a field.</p> <p>No result is returned.</p>
Interactive	No	

**Note:** Any other triggers not listed here cannot be used in Aurea CRM.Web.

## Trigger Action Parameters

Supported trigger Action Parameters.

The `Trigger` action supports the following input arguments:

### name

The name of the trigger format to be executed.

### **uid**

The `uid` of an optional record passed to the trigger.

### **recordSet**

Defines for which records from a list or sub-list the trigger is executed: `selected`, `all` or `visible`.

Optionally, the selection can be further filtered by adding either an info area ID or "Root" in []. Example: `selected[Root]`

If you specify "Root" only the root record of each row is included in the record set.

If you specify an info area ID, only records belonging to that info area are included in the record set.

**Default:** `selected`

### **infoAreaId**

Determines the trigger's "context" which is the info area on which the trigger is executed.

Use this parameter if you have specified neither `uid` nor `recordSet`. Example it can be used for a trigger that creates a new to-do record, specify `infoAreaId = TD`.

### **refreshAfterExecution**

If set to `true`, any visible view (list, sub-list, expand) containing records that have been changed by an executed trigger or workflow is refreshed immediately.

**Default:** `true`

---

**Note:** If a trigger calls another trigger (defined in the currently active rights definition), changes performed by that follow-up trigger may not be immediately reflected in the current view.

---

### **callbackFunction**

This argument defines an optional callback function that is called after the execution of a trigger or workflow.

You can specify:

- a custom JavaScript function
- a framework function
- an inline function

### **Syntax**

To call a custom JavaScript or framework function enter the function name:

`javascript:triggerCallbackFunction`

To call an inline function enter the JavaScript code:

```
javascript:function(sender, args) {alert("this is a trigger callback function");}
```

### callback

A marker parameter for the framework entered automatically if a callback is required for the execution.

## URL

Learn about the URL action templates.

Use the `URL` action template to call an URL from Aurea CRM.Web Example - to display an external website.

The `URL` action template can be used for actions defined for application menu entries.

The `URL` action template supports the following input arguments:

### Target

See [Target](#).

### URL

See [URLPage](#).

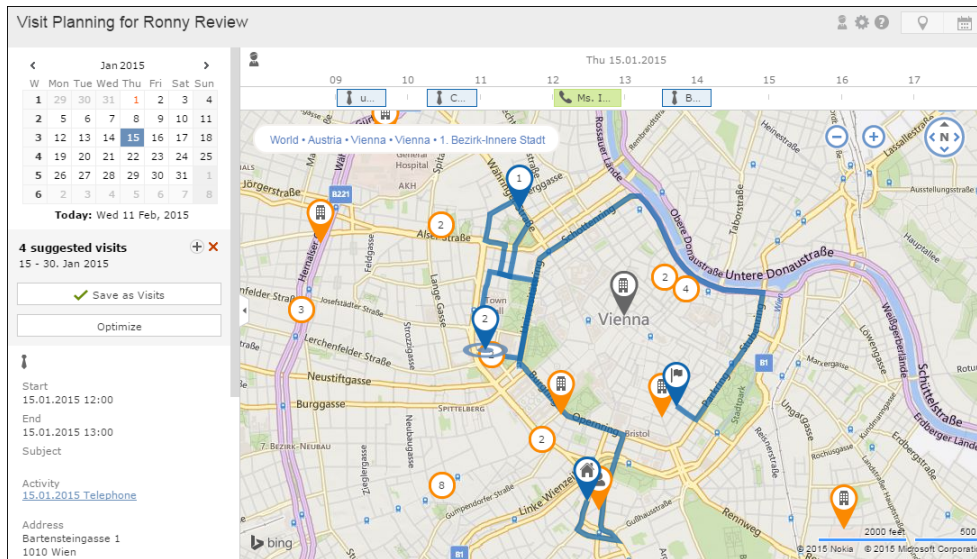
If you want to open an external page that requires additional parameters like a record ID, use the `RedirectPage` action template instead. For further details, see [RedirectPage](#) on page 444.

To prevent phishing attacks when opening an external page, you can define a black/white list in the `settings.xml` thus ensuring that an `URL` action cannot be redirected to a harmful site. For further details, see **Security Settings** in the *Aurea CRM.Web Installation Guide* ([https://help.aurea.com/crm/crm-web-install-guide/crm-web-install-guide/dita/topics/c\\_web\\_install-1\\_1-Introduction\\_crm.html](https://help.aurea.com/crm/crm-web-install-guide/crm-web-install-guide/dita/topics/c_web_install-1_1-Introduction_crm.html)).

## VisitPlanning

Learn to plan routes on an interactive map.

The `VisitPlanning` action template is used to create a location-based visit planning page allowing users to plan routes based on their own location and the locations of their customers on an interactive map.



In UPDATE\_DEFAULT the VisitPlanning action is called by the VisitPlanning application menu action configured in ID\_NET\_APPOINTMENT\_MGMT (**Collaboration > Visit Planning**). In addition, the A\_VisitPlanningFromCustomer context menu action is available in the context menus for companies and persons (M\_Company, M\_Person).

Based on the information saved in the Aurea CRM database (location, working/opening hours, ABC ratings, open work orders etc. For further details, see [Default Values](#) on page 481) Visit Planning provides the following functionality:

- **Creation of suggestions** for customer visits for the selected rep: Visit suggestions are created as **Suggestion** (SV) records that are displayed as pins on the map and items in the time line above the map. Existing activity (MA) records for the selected rep and day are displayed as well.

For any issues on the route warning about idle times, a visit being outside opening hours etc.) **Annotation** (SX) records are created and are displayed within the route using different icons based on the value in the **Annotation quality** field (SX:5).

- Users can manually **adjust the route** as follows:
- edit suggestions directly in the map
- rearrange items using drag&drop on the time line
- add/remove persons/companies by selecting **Add to/Remove from Route** from the context menu (M\_VisitPlanning).

Driving times and annotations are updated accordingly. Other pins in the route are not changed.

- **Optimization** of manually created routes by clicking the **Optimize** button, see the article “Visit planning optimization” at <https://support.aurea.com>.



Existing MA records are not changed by the optimizer.

- **Creation of activities:** Clicking on the **Save as visits** button creates MA records of type "Visit" from the suggestions on the map. The SV records are deleted automatically.

The starting point of the route is not turned into an MA record. A route's starting point is defined by the **Starting location** flag (SV:12). Make sure this is never manually changed (example - via the `Readonly` field attribute).

For a detailed description of the use interface and functionality, see **Visit Planning** in the *Web Business Logic User Guide* ([https://help.aurea.com/crm/crm-web-bl-btb/crm-web-bl-otc/dita/topics/c\\_web\\_bl\\_btb-1\\_1-Aurea-CRM-web\\_crm.html](https://help.aurea.com/crm/crm-web-bl-btb/crm-web-bl-otc/dita/topics/c_web_bl_btb-1_1-Aurea-CRM-web_crm.html)).

For a complete description of the Visit Planning algorithm, see the article "Visit planning optimization" at <https://support.aurea.com>

Use the **SDK** page to test all elements of the Visit Planning page.

## Default Values

Learn about the default values in Visitplanning action template.

Visit planning is not functional out-of-the-box. The following minimum customizations must be in place for the feature to work.

Manual visit planning:

- A valid Bing license key must be specified in the **License key** configuration parameter (**Geo Coordinates** category) in the **Configuration** info area (MC), see [Configuration Info Area](#) in the *CRM.Core Administrator Guide*.
- Location data (i.e. geographic coordinates) for your reps and customers must be available. They should be entered in the respective **Latitude** and **Longitude** fields. For further details, see [Location](#). For further details, see [Geo-Coordinates](#) in the *CRM.Core Administrator Guide*.

Generation of suggestions:

- You need to define queries for visit candidates and planned visits that need to be referenced via Z5 variables. For further details, see [Variables and Queries](#).
- The info areas to be searched and the default time frame for creating suggestions are defined in the action. For further details, see [VisitPlanning Action Parameters](#) on page 484.

## Rep

The  (**Select Rep**) button in the page header allows selecting another rep.

The selected rep's name is displayed in the header and in the info text in the **New Suggestions** dialog.

**Note:** Since important parts of the Visit Planning configuration depend on Z5 variables, you need to take into account that **variables are always evaluated for the currently logged-in rep**. E.g. if a variable value depends on the rep's country, a rep from one country cannot correct planning for reps from another country.

---

**Default:** currently logged-in rep

### Location

For customer records (companies and persons) to be considered by the Visit Planning, their location must be present in the database, entered in the respective **Latitude** and **Longitude** fields.

The rep's starting point is specified in the **New Suggestions** dialog (**Start from** and **Start next day from** drop-down lists). Available values:

- "Home" : The starting point is read from the **Latitude** and **Longitude** fields in the selected rep's ID record.
- "Office": The coordinates of the selected rep's linked FI record are used.
- Last Visit: The coordinates of the location of the last MA record (last 3 days, maximum) are used. The location to be used is determined in the following order: FI > KP > PE.

When optimizing the route using the **Optimize** button, the first visit (MA or SV) of the day is always used as starting point.

**Default:** Home

### Working Hours / Contact Times

#### Rep

Suggestions are created within the rep's working hours as defined in the rep's **Rep** (ID) record by checking Individual work hours (ID:105) and entering the working hours in the fields ID:29 to ID:56. If no individual working hours are defined for a rep, the settings defined in the `Calendar.DayStartsAt` / `Calendar.DayEndsAt` Web Configuration parameters are used as a fallback. For further details, see [Calendar Parameters](#) on page 97.

#### Customer

Contact times (CT) defined for companies and persons are taken into account. If no contact times have been specified for a company or person, Aurea CRM assumes the customer is available any time during the rep's working hours. No annotation is created in that case.

As for reps the `Calendar.DayStartsAt` / `Calendar.DayEndsAt` Web Configuration parameters are used as a fallback.

In addition to working hours/contact times, public holidays as specified in [Calendar.HolidayCalendarID](#)) and company holidays as defined in CT:27/28 are taken into account.

**Default working hour:** 8 AM to 6 PM

**Default contact time:** during working hour.

## Variables and Queries

Suggestions for the selected rep and time frame are calculated based on the following:

- a query determining already planned activities that are part of the route,
- a query determining the "candidates" for visits based on values like ABC, target vs. actual visits etc.
- additional parameters like default visit length, weighting, break time between visits.

These settings are referenced via Z5 variables. For further details, see [Variables](#) on page 53.

The variables referencing the two queries must be defined:



- `VisitPlanning.Query.Candidates`
- `VisitPlanning.Query.Planned`

For the optional variables appropriate defaults are in place if undefined.

## Configuration Units

Learn about configuring units for visit planning page.

The Visit Planning page consists of the following configuration units:

- The application menu action `VisitPlanning` and context menu action `A_VisitPlanningFromCustomer` with action `VisitPlanning`. For details on the available action parameters. For further details, see [VisitPlanning Action Parameters](#) on page 484.  
Both menu actions are assigned to CRM process 1.2.6.2. "Visit Preparation" (CRM process ID 70000320).
- The special header `VisitPlanning` containing the following buttons:
- A  (**Select Rep**) button allowing the user to select another rep.
- A  (**Settings**) button allowing the user to filter the items (companies and persons) displayed on the map. Analogous to Calendar filter sets, you can provide default filters sets for your users via the `VisitPlanning.FilterSets` Web Configuration parameter. For further details, see [VisitPlanning.FilterSets](#).

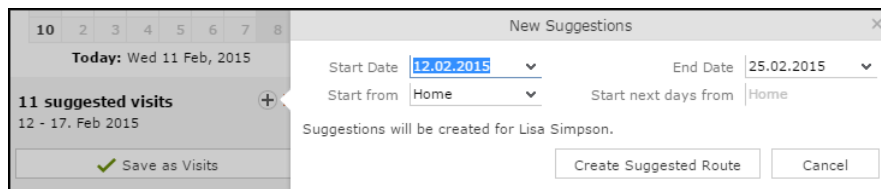
In addition, you can specify fixed filters in the action. For further details, see [customerInfoAreasFilter](#).

The filter criteria are also taken into account when Aurea CRM calculates suggestions.

Information about the selected rep and active filter sets is displayed in the **New Suggestions** dialog.

- A legend button explaining the color coding and icons on the map. For further details, see [Legend](#) on page 315.

- Toggle buttons allowing the user to toggle between the **Map** and **Calendar** view.
- The (collapsible) side bar contains the following elements:
- a date picker



- the suggestions area:
  - **+** (**New suggestions**): Calls the **New Suggestions** dialog. Start and end date is configurable in the action. Starting location is "Home" by default. For further details, see [Location](#).
  - **×** (**Discard all suggestions**): Deletes all SV (and SX) records for the currently selected rep.
  - **Save as visits**: Creates MA records from the SV records.
  - **Optimize**: Optimizes the currently planned route, see the article Visit Planning Optimization at <https://support.aurea.com>.
- information on the currently selected item, pin or time line item, configured by the MiniDetails controls of field groups `FI/KP/MA/PE/SV.VisitPlanning`.
- The table captions that are used in the side bar, map area and time line must follow the same naming convention (`AF/KP/FI/PE/SV/SX.VisitPlanning` are defined in `UPDATE_DEFAULT`).
- The static texts for the Visit Planning page are defined in the `VisitPlanning` text group.

## VisitPlanning Action Parameters

Supported VisitPlanning Action Parameters.

The `VisitPlanning` action supports the following input arguments (all optional):

### uid

An optional record to be included in the map (`Record` or `Linked`). This argument is used if the action is called from a company or person.

### customerInfoAreas

A comma-separated list of info area IDs specifying from which info areas visit suggestions are created.

Default: `FI, KP, PE`.

### customerInfoAreasFilter

Allows you to specify fixed filters per info area (comma-separated).

### defaultSuggestionStart

Determines the start date used in the **New Suggestions** dialog. You can specify a fixed date or a date expression. For further details, see [Using Variable Expressions](#) on page 505

**Default:** \$curDay+1

This setting defines the initial default when first accessing the Visit Planning page. Afterwards the date selected by the user is saved in his roaming settings. For further details, see [User Settings](#) on page 25.

### defaultSuggestionDuration

Determines the end date used in the **New Suggestions** dialog in days (1 to 31).

**Default:** 1.

## Workflow

The Workflow action template allows you to call a workflow via menu entries, buttons and within processes.

The `Workflow` action template, by calling the `javascript:u8.services.businessObjects.executeWorkflow` function, allows a non-interactive (or background) workflow to be executed in Aurea CRM.Web. For further details, see [Executing Non-Interactive Workflows in Aurea CRM.Web](#) on page 485.

The `Workflow` action template allows you to call a workflow via menu entries, buttons and within processes.

For detailed information on workflows and how to define them, see Aurea CRM automator documentation.

---

**Note:** This feature only works for workflows that read records. It is not intended for workflows that modify records.

---

## Executing Non-Interactive Workflows in Aurea CRM.Web

Learn about Non-Interactive Workflows in Aurea CRM.Web.

A workflow you want to call via the `Workflow` action template must meet the following conditions:

- the workflow contains no user interface elements
- no user interaction is needed during the workflow's execution
- the workflow is executed in the background, i.e. Hidden is checked in the workflow's settings.

When calling a workflow via a button in a `SearchResults` header, use the **Query State Action** setting in the button's definition to control when the button is active (one/many/no records selected in the list). For further details, see [Configuring a Button](#) on page 305.

Workflow calls and results are written to the server log in debug mode. If an error occurs, a message is written to the client log, see [Core Error Codes for Other Aurea CRM Products](#) in the *CRM.Core Administrator Guide*.

## Workflow Action Parameters

Supported Workflow Action parameters.

The `Workflow` action supports the following input arguments:

### name

The name of the workflow format to be executed.

### uid/uid2

The `uid` of one or two optional records passed to the workflow.

### parameters

Parameters passed to the workflow.

### Syntax:

```
javascript:[{name: "sParam", values: ["string parameter"]},  
{name: "fParam", values: ["32.55"]},  
{name: "iParam", values: ["3542"]}]
```

### callback

A marker parameter for the framework entered automatically if a callback is required for the execution.

### refreshAfterExecution

If set to `true`, any visible view (list, sub-list, expand) containing records that have been changed by an executed trigger or workflow is refreshed immediately.

**Default:** `true`

---

**Note:** If a trigger calls another trigger (defined in the currently active rights definition), changes performed by that follow-up trigger may not be immediately reflected in the current view.

---

### recordSet

Defines for which records from a list or sub-list the workflow is executed. Same as for triggers. For further details, see [recordSet](#).

---

**Note:** Regardless of the number of selected records, only the records actually required by the workflow are taken into account (max. 2). The workflow is NOT called individually for each selected record.

---

### callbackFunction

This argument defines an optional callback function that is called after the execution of a trigger or workflow.

You can specify

- a custom JavaScript function
- a framework function
- an inline function

### Syntax

To call a custom JavaScript or framework function enter the function name:

```
javascript:triggerCallbackFunction
```

To call an inline function enter the JavaScript code:

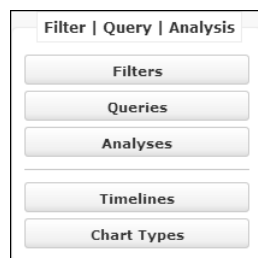
```
javascript:function(sender, args) {alert("this is a trigger callback function");}
```

## Filter | Query | Analysis

Learn how to define and manage filters to generate records for analyses.

You can use the **Filter | Query | Analysis** area of CRM.Designer to do the following:


- define/customize filters, queries, and analyses for accessing data in your Aurea CRM database.
- configure timelines graphically displaying records along a timeline.
- customize the chart types used for analyses.




## Filters

Filters are used to filter data by defining conditions.



































In Aurea CRM.Web, filters can be used and defined on:

- Search pages via the **Add more** feature
- **Calendar** and **Resources** pages by clicking  (**Define filters and filter sets**)

To define/customize filters in CRM.Designer:

1. Go to CRM.Designer main page.
2. Click **Filters** . **All Filter** page is displayed.

For further details, see [Designing Configuration Units](#) on page 64.

Infoarea	Filter Name	Configuration	Description
Task (A1)	A1.MyTasks	  UPDATE_DEFAULT	
	A1.ServiceStartpage	  UPDATE_DEFAULT	
	A1.Sidebar	  UPDATE_DEFAULT	
Absence (A3)	A3.MyAbsences	  my_config	
	A3.Sidebar	  UPDATE_DEFAULT	
Work Order (AF)	AF.WordOrders_with_Act...	  UPDATE_DEFAULT	
Mailing (BR)	BR.Mailings_with_Activity	  UPDATE_DEFAULT	
Reminder (ER)	ER.MyReminders	  UPDATE_DEFAULT	
Questionnaire/Portfolio (...)	F1.Portfolio	  BASE/UPDATE_DEFAULT	
	F1.PortfolioFI	  BASE/UPDATE_DEFAULT	
	F1.PortfolioOJ	  UPDATE_DEFAULT	
	F1.PortfolioPJ	  UPDATE_DEFAULT	
	F1.PortfolioY1	  BASE/UPDATE_DEFAULT	
	F1.Questionnaire	  BASE/UPDATE_DEFAULT	
Company (F1)	F1.Company_from_DACH	  my_config	
	F1.filter_in_parent_config	  customized_parent_config	
Rep (ID)	ID.curOrgGroup	  my_config	

A filter definition consists of an info area and optional related info areas and filter criteria. For further details, see [Defining Filters](#) on page 489. Unlike queries, filters do not have output and sort fields. For further details, see [Queries](#) on page 493.

Filters are saved as configuration units in the current user's configuration. They are always stored in and loaded from the CRM.Designer database. Filters defined in parent configurations are visible in both Aurea CRM.Web and CRM.Designer. Users can simply use and change them, and save them to their configuration.

Use the **Readonly** and/or **Invisible in lists** flags in the filter definition in CRM.Designer, to hide a filter and/or prevent users from changing it.



In Aurea CRM.Web users can define filters for all info areas to which they have access rights. To restrict the list of available info areas without generally denying access to an info area:

1. In the **Rights** module, click in the **Additional settings per info area** cell of the desired info area.
2. Check the **Do not display the info area used to define filters and queries (web)** check box.

For more information, see **Rights** in the *CRM.Core Administrator Guide*.

Users can still use filters containing info areas that they cannot select themselves. However, if a user removes such an info area from the filter definition, he cannot add it again due to his rights restrictions.

In CRM.Designer you can define filters regardless of your access rights.

## Defining Filters

Defining filters in Aurea CRM.

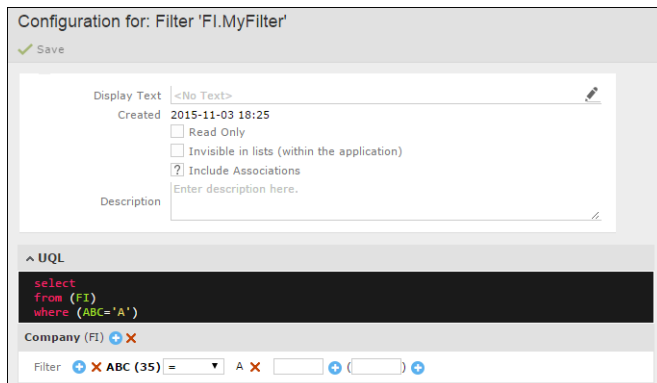
In Aurea CRM.Web and the CRM.Designer you can define and edit filters using the AQL syntax. For further details, see [AQL Syntax](#) on page 882 and Filters in the Aurea CRM.Web User Interface documentation. You can also edit the AQL syntax in the AQL editor.

To define a filter in CRM.Designer:

1. Enter a **Name**.
2. Select the root info area from the **Infoarea** drop-down list.
3. Click **New**.

The info area ID of the root info area is automatically added to the filter's name as a prefix uniquely identifying the filter in CRM.Designer. Example - FI.MyFilter  
The prefix is not displayed when selecting the filter in Aurea CRM.Web.

4. Click  (**Edit**).



Configuration for: Filter 'FI.MyFilter'

Save

Display Text <No Text>

Created 2015-11-03 18:25

Read Only

Invisible in lists (within the application)

Include Associations

Description Enter description here.

UQL

```
select
from (FI)
where (ABC='A')
```

Company (FI)

Filter ABC (35)

## 5. Specify the following:

- **Display-Text** (optional): You can enter a multilingual name for the filter. This name is used when selecting filters in Aurea CRM.Web. The text is saved in the `net_filternames` text group. For further details, see [Texts](#) on page 176.
- **Readonly**: If checked, users cannot edit the filter in Aurea CRM.Web.
- **Invisible in lists**: If checked, users cannot select the filter in Aurea CRM.Web. If unchecked, users can select and edit the filter. Visible filters from parent configurations are read-only.
- **Description**: Add an internal description visible in CRM.Designer only.

6. **Filter**: Define the filter criteria:

- Select a field and click + (**Add**).

---

**Note:** Filters on decoded fields (Z-fields) are not supported. For further details, see [Data Types in Aurea CRM](#) on page 877.

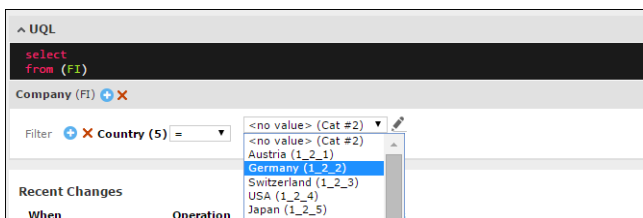
---

- Select an operator from the drop-down list.

- Enter a value, a variable expression or function and click + (**Add**). For further details, see [Advanced Filter Criteria](#) on page 504.

For catalog values you can either enter the text in the default language or the catalog code (as `#<code>`). In addition, you can search by external key (`#xk:`) or by (part of) the value's description (`#desc:`). For a complete description of the syntax plus examples, see [Filtering by Catalog Values](#) on page 885.

For synchronized catalogs (see [Updating Catalogs](#) on page 778 and [Catalog Value Synchronization](#)) you can select the value from the drop-down list:



To filter by values from hierarchical catalogs, define the filter for the child catalog and define the parent and child value as values:




---

**Note:** The AQL notation using "~" is not supported in CRM.Designer.

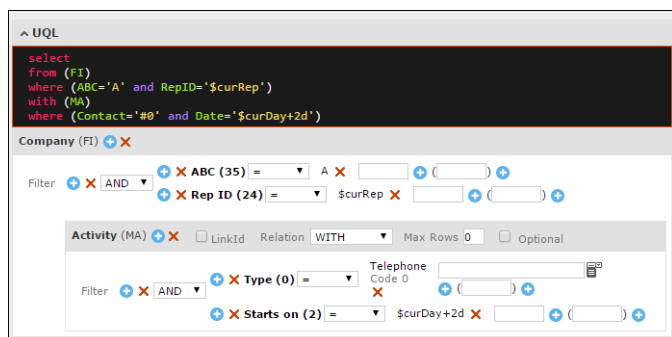
---

---

**Note:** For hierarchical catalog values, the parent catalog value must be the same for all values of one filter node. You need to enter different parent catalog values as multiple nodes.

---

7. To add a related info area, click + (**Add**) beside the root info area and select the desired info area.



- a. Specify the following:

- **LinkId:** If you want to use another than the default link, specify the **LinkId**. For further details, see [Link IDs](#) on page 19.
- **Max Rows:** Defines how many linked records are read for each parent.
- **Optional:** Check **Optional** to display the root record only if a related record exists in at least one of the sub-info areas.

- b. Select the desired **Relation** type:

- **HAVING:** Returns the root record only if at least one related record exists.
- **WITHOUT:** Returns the root record only if no related record exists.

- c. Define filter criteria for the related info area(s) if applicable, e.g. to filter for companies with at least one **Activity** record of type "Telephone" on the day after tomorrow.

8. Click **Save**.

AQL syntax for the above example is visible in the AQL editor, as shown below:

```
select from (FI)
where (ABC="A" AND RepID="$curRep")
with (MA as FIMA)
where (Contact="#0" AND Date="$curDay+2d")
```

---

**Note:** The maximum number of filters is 1000 for a single query.

---

## Working with filter for multiple info areas

Configuring to use multiple filters in a selectRecord function when two or more info areas exists.

To use multiple filters in a selectRecord function when two or more info areas exist the following configuration is required.

To enable multiple filters with more than one info area:

1. Use CRM.Designer to create filters for two different info areas. For example - `FI.CompanyCity` and `KP.ContactName`.
2. In CRM.Designer add text field with selected record in some third info area filled group control search. For example, select record for filter in FI and KP as shown below:

```
selectRecord({
  from: {showFilter:true,
  infoareas:[{infoAreaId: "FI", filter: "FI.CompanyCity"}, {infoAreaId:
"KP", filter: "KP.ContactName"}]},
  useAutoSearch:true }
});
```

3. Open CRM.web and locate the info area where the search field is added. Use the selected text field to search for Company and `FI.CompanyCity`. In case you search for Contact Person, the `KP.Company` name is also applied to the search.

## Format Types for Queries and Analyses

Format type used for queries and analyses.

The following format types are used for queries and analyses:

- **Private:** The definition is stored per user.
- **Public:** The definition is shared by all users. If the definitions are stored in the CRM.Designer database, they are stored in the highest writable parent configuration.
- **All:** The standard designer logic is applied. For an analysis name, all private, all public and all analyses defined above the configuration for public analyses are searched.

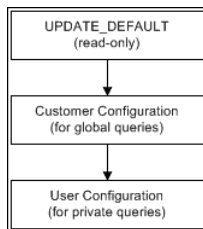
Queries can be saved in the CRM.Designer and/or Aurea CRM database. For further details, see [Storing Queries](#) on page 495.

Queries are divided into public queries available for all users and private queries that only belong to one user.

For queries stored in the Aurea CRM database, this difference is stored in the format for public queries the **User** field is empty. Queries that are stored in/loaded from the CRM.Designer database, the definition whether a query is private or public depends on the following:

- For loading queries, all queries in the user configuration are private and all queries in one of the parent configurations are public.
- For saving queries, private queries are stored in the current configuration and public queries are stored in the highest writable parent configuration.

A configuration structure allowing users to load and save private and public queries is shown in the following graphic:



Analyses are always saved in the CRM.Designer database, even if the underlying query comes from the Aurea CRM database. Public analyses (defined in Aurea CRM.Web) are stored in the highest writable parent configuration.

Use the `DesignerConfiguration.PublicSaveConfiguration` Web Configuration parameter to define the configuration where public queries and analyses are stored. For further details, see [DesignerConfiguration.PublicSaveConfiguration](#).

---

**Note:** If the parent configuration of the user configuration is read-only, the user gets an error message when trying to save a public query or analysis to the CRM.Designer database.

---

## Queries

Learn about the advanced way of accessing data by using queries.

Queries provide an advanced way of accessing data. The user can select the fields of an info area that can be seen, the filters that are applied to those info areas and how the result is sorted.

To define/customize queries in CRM.Designer, click **Queries** on the CRM.Designer main page. For further details, see [Designing Configuration Units](#) on page 64 and [Defining Queries](#) on page 496 .

All Queries	
Name <input type="text"/>	Copy from <input type="text" value="Empty Query"/> <input type="button" value="New"/>
Query Name	Configuration
Activities with Contacts	UPDATE_DEFAULT
AK.FindAsYouType	BASE/UPDATE_DEFAULT
AR.FindAsYouType	BASE/UPDATE_DEFAULT
CM.FindAsYouType	BASE/UPDATE_DEFAULT
Companies_Without_TerritoryAllocation	UPDATE_DEFAULT
Company_BuyingAnalysis	UPDATE_DEFAULT
Company_ContactAnalysis	UPDATE_DEFAULT
FI.FindAsYouType	UPDATE_DEFAULT
FI_Country_Revenue	my_config
IB.FindAsYouType	UPDATE_DEFAULT

In Aurea CRM.Web queries are used for/by:

- In the **Query Designer** users can execute predefined or saved queries, modify or create new queries according to the assigned query rights. For further details, see [Query.Rights](#).
- **Analyses** are based on queries. For further details, see [Defining Analyses](#) on page 515.
- **Forms** (e.g. dashboards for start pages) can contain e.g. a `QueryResultPresenter` widget executing a query and displaying the count of returned records. For further details, see [Forms](#) on page 621.
- The **QueryRun** action template executes a query and displays the query results. For further details, see [QueryRun](#) on page 441.
- The **QueryReport** action template executes a query in the background and displays a report using the query results. For further details, see [QueryReport](#) on page 436.
- **CRM.server** uses queries to create target groups for marketing activities. For further details, see [Mail Merge Letters Based on Marketing Activities](#) on page 849.

In addition, CRM.server can generate query snapshots or a CSV export of query results in the background. For further details, see [Background Queries](#) on page 502.

Use the **SDK** page to explore the functionality of all query-related widgets: **SDK > Query**.

## Queries and Access Rights

In Aurea CRM.Web users can define queries for all info areas to which they have access rights.

In CRM.Designer you can define queries containing any info area regardless of the access rights of your users. If a query contains locked info areas all elements except the ones that are locked are returned. This means a locked info area appears in Aurea CRM.Web as if no data is available.

Example - You may prepare a report containing data from **Company**, **Person**, **Activity** and **Offer** for all users regardless of their individual access rights and the **Offer** info area may be locked for some users. Hence, the report contains no data for **Offer** info area for the blocked users.

---

**Note:** If the root of the query is affected by the rights definition, e.g. because it contains a WITH condition for a locked child info area, no data is returned.

---

To restrict the list of available info areas without generally denying access to an info area:

1. In the **Rights** module, click in the **Additional settings per info area** cell of the desired info area
2. Check the **Do not display the info area used to define filters and queries (web)** check box.

For more information, see **Rights** in the *Aurea CRM win Administrator Guide*.

Users can still use queries containing info areas that they cannot select themselves. However, if a user removes such an info area from the filter definition, he cannot add it again due to his rights restrictions.

If a query contains a field that is locked in the current access rights definition (**Deny Access** on field level), the field is displayed with "Access denied" in the **Output Fields** list. The locked field can not be selected for defining a condition. However, when defining the condition as AQL any field can be used.

## Storing Queries

This topic has information about saving and loading queries.

The `Query.LoadFrom` and `Query.SaveTo` Web Configuration parameters determine from where queries are loaded and where they are saved. For further details, see [Query.LoadFrom](#) and [Query.SaveTo](#).

Depending on the `Query.SaveTo` Web Configuration parameter queries defined in Aurea CRM.Web are saved

- as formats of type "query" in the Aurea CRM database (default)
- as configuration units in the CRM.Designer database
- as both

Queries defined in Aurea CRM.Web can be saved as either "public" or "private". For further details, see [Format Types for Queries and Analyses](#) on page 492:

- Private queries are saved in the current user's configuration if `Query.SaveTo = "CRM.Designer"` or under the current user's name if `Query.SaveTo = "Aurea CRM"`.
- Public queries are saved in the current user's highest writable parent configuration or in the configuration specified in the `DesignerConfiguration.PublicSaveConfiguration Web Configuration` parameter if `Query.SaveTo = "CRM.Designer"` or without a user name (if `Query.SaveTo = "update.CRM"`). For further details, see [DesignerConfiguration.PublicSaveConfiguration](#).

Queries defined in CRM.Designer are always saved as configuration units in the CRM.Designer database.

The queries that are available for users in Aurea CRM.Web depends on the `Query.LoadFrom Web Configuration` parameter.

**Default:** "both, CRM.Designer first", i.e. the query is first searched in the CRM.Designer database, and if not found, it is loaded from the Aurea CRM database.

It is recommended that queries that are defined for customization purposes (to display a query result, an analysis or query results on generic form) are stored in the CRM.Designer database. Thus, the query is also transferred with the referencing objects during a bulk download and upload.

If you define all queries (and analyses) that are directly referenced in the configuration in CRM.Designer, you can set both `Query.LoadFrom` and `Query.SaveTo` to "update.CRM". Users does not see the queries and analyses used by the configuration when they search for queries and analyses in Aurea CRM.Web.

Queries that are executed in batch operations by CRM.server (e.g. Marketing Activity Execution) require the default setting `Query.SaveTo = update.CRM`.

## Defining Queries

Learn how to define queries in Aurea CRM.

In Aurea CRM.Web and CRM.Designer you can define and edit queries using the AQL syntax, see [AQL Syntax](#) on page 882 and [Queries in the Aurea CRM.Web User Interface](#) documentation.

To define a query in CRM.Designer:

1. Select a query or create a new one. For further details, see [Designing Configuration Units](#) on page 64.

When creating a new query you need to select the root info area first:





Configuration for: Query 'MyQuery'

✓ Save

Created 2015-11-03 20:19

Read Only

Invisible in lists (within the application)

Include Associations

Enable Hints

CRM Processes

Description

^ UQL

```
select (Company, Country, ZipCode, Employees, Revenue, MA.Contact, MA.Date, MA.Time, MA.Duration, MA.Unit, MA.Subject, MA.Person)
from (FI)
where (ABC='A')
with (MA)
where (RepID='$curRep' or ParticipantID='$curRep')
orderby (Company, MA.Contact, MA.Date)
```

Company (FI)

Fields  Company (2) , Country (5) , Postal Code (7) , Employees (37) , Revenue (36)  FI-StaNo (0)

Sort Company (2) (asc)  FI-StaNo (0)

Filter   ABC (35) =

Activity (MA)    LinkId Relation WITH  Max Rows 0  Optional

Fields  Type (0) , Starts on (2) , Starts at (3) , Duration (4) , Unit (5) , Subject (103) , Person (77)

Status (1)

Sort Type (0) (asc) , Starts on (2) (asc)  Status (1)

Filter   OR    Rep ID (8) =

Participant ID (12) =

## 2. Specify the following:

- **Readonly:** If checked, users cannot edit the query in Aurea CRM.Web.
- **Invisible in lists:** If checked, users cannot select the query in Aurea CRM.Web.

**Note:** Use these settings for queries called via menu actions or used by predefined analyses. Alternatively, you can use the `Query.Rights` Web Configuration parameter to determine whether users can execute and/or edit predefined queries. For further details, see [Query.Rights](#).

- **Include Associations:** Determines whether affiliations should be taken into account. For further details, see [Global Switch / Query-Specific Switch](#).
- **CRM Processes:** Assign one or more CRM processes to the query. For further details, see [Roles & CRM Processes](#) on page 727.
- **Description:** Add an internal description (only visible in CRM.designer).
- **Used by Mail Merge:** Select this option to make the query appear on the template document used for Mail merge.

**Note:**


- The flag `Use for Mail Merge` is automatically set by the system when a query is used by Office Add-in
- The flag `Use for Mail Merge` is explicitly set by users in the Query Definition screen.

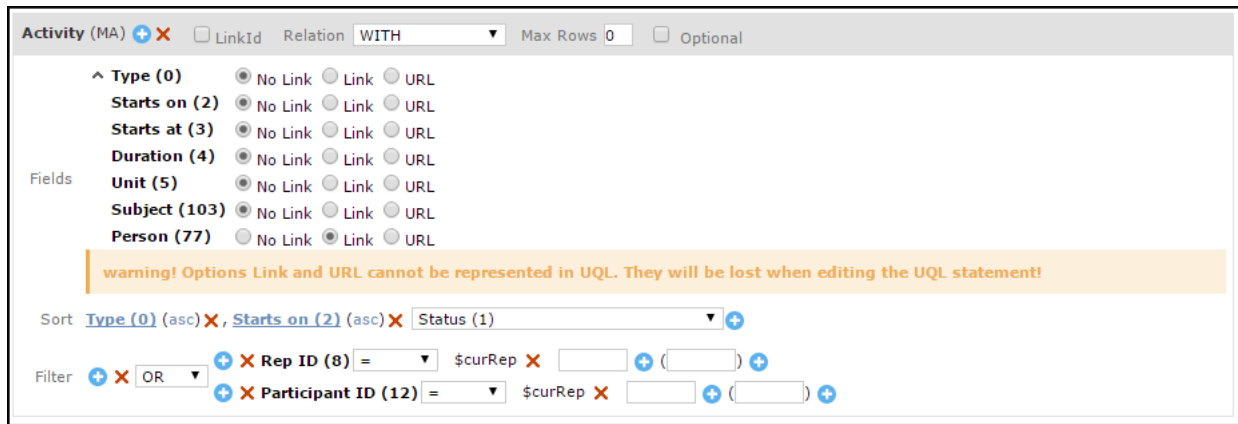
**Note:** On deleting a query with flag `Use for Mail Merge` set, the following additional pop-up message is shown:

**This query is selected to be used with Word Single- & Serial Letters. Deletion of the Query can cause different Word Templates stop working. Do you still want to delete the 'ExampleQuery' query?**

### 3. Fields: Define the query's output fields.

**Note:** By default, rep ID fields (field type "uid") are transformed to contain the rep name (string) when displaying and exporting a query result. To get the rep ID instead, use the `Export.EmitRepNames` Web Configuration parameter or export option. For further details, see [Export.EmitRepNames](#).

After adding your fields, you can expand them using the  button to define which fields is shown as links to records by checking desired the Link-Field boxes.



The screenshot shows the 'Activity (MA)' query definition screen. At the top, there are controls for 'LinkId', 'Relation' (set to 'WITH'), 'Max Rows' (0), and 'Optional'. Below this, a list of fields is shown with radio buttons for 'No Link', 'Link', and 'URL'. The fields are: Type (0), Starts on (2), Starts at (3), Duration (4), Unit (5), Subject (103), and Person (77). A warning message states: 'warning! Options Link and URL cannot be represented in UQL. They will be lost when editing the UQL statement!'. Below the fields, there are sort and filter options. The sort options are 'Type (0) (asc)', 'Starts on (2) (asc)', and 'Status (1)'. The filter options are 'Rep ID (8) = \$curRep' and 'Participant ID (12) = \$curRep'.

Link fields are displayed like hyperlinks when a user executes a query or analysis, on the `QueryRun` page for queries used in forms and in analysis results.

The link opens the info area the field is defined for. If a Z-field (i.e. a read-only field consisting of several fields from another info area) is defined as a link, this associated info area is opened when a user clicks on the link. In the example the link to the Z-field **Person** opens the **Person** (KP) record.

- 
- Sort:** Define the query's sort fields. Click **asc** to change the sort order to **desc** and vice versa.
  - Filter:** Define the filter criteria. For further details, see [Defining Filters](#) on page 489, [Parameterized Queries](#) on page 500 and [Advanced Filter Criteria](#) on page 504.

---

**Note:** Filters on decoded fields (Z-fields) are not supported. For further details, see [Data Types in Aurea CRM](#) on page 877.

---

- To add a (child) info area, click + (**Add**) beside the root info area and select the desired info area. Specify the following:
  - LinkId:** If you want to use another than the default link, specify the **LinkId**. For further details, see [Link IDs](#) on page 19.
  - Max Rows:** Defines how many linked child records are read for each parent.
  - Optional:** Check **Optional** to display the parent record only if a child record exists in at least one of the sub-info areas.
- Select the desired **Relation** type:
  - WITH:** Returns the parent record only if at least one child record exists.
  - WITHOUT:** Returns the parent record only if no child record exists; allows advanced conditions on the child records to be defined.
  - PLUS:** Returns the parent record even if no child record exists. (This allows to implement the `UNION` operator from SQL.)
  - WITH SUMMED:** Returns the parent record only if at least one child record exists; aggregates data from child records.
  - PLUS SUMMED:** Returns the parent record even if no child record exists; aggregates data from child records.
  - HAVING:** Returns the parent record only if at least one child record exists; allows advanced conditions on the child records to be defined.

The query definition in CRM.Designer also supports extended conditions, but it does not include Aurea CRM.Web's restrictions on extended conditions. Therefore, you must define queries that can actually be executed.

Extended conditions can only be used for tables that are related with the "HAVING" or "WITHOUT" types. For these types, you can use the SUM and COUNT fields at the end of the field list. Do not mix standard conditions and extended conditions and combine them with "AND".

- Define output fields, sort fields and filter criteria for the child info areas as applicable.
- Click **Save**.

Queries defined in CRM.Designer are always saved as configuration units in the CRM.Designer database.

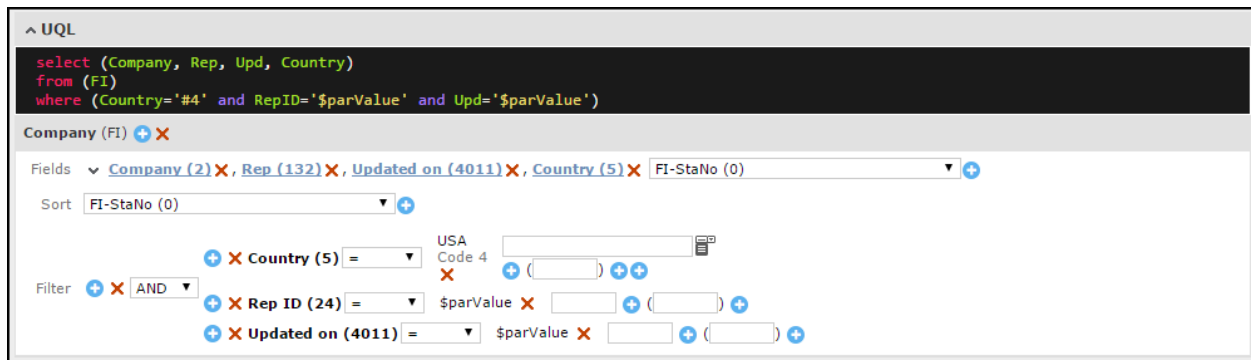
AQL syntax for the above example is visible in the AQL editor and is shown below:

```
select (Company, Country, ZipCode, Employees, Revenue, FIMA.Contact, FIMA.Date,
FIMA.Time, FIMA.Duration, FIMA.Unit, FIMA.Subject, FIMA.Person) from (FI)
where (ABC="A")
with (MA as FIMA)
where (RepID="$curRep" OR ParticipantID="$curRep")
orderby (Company, FIMA.Contact, FIMA.Date)
```

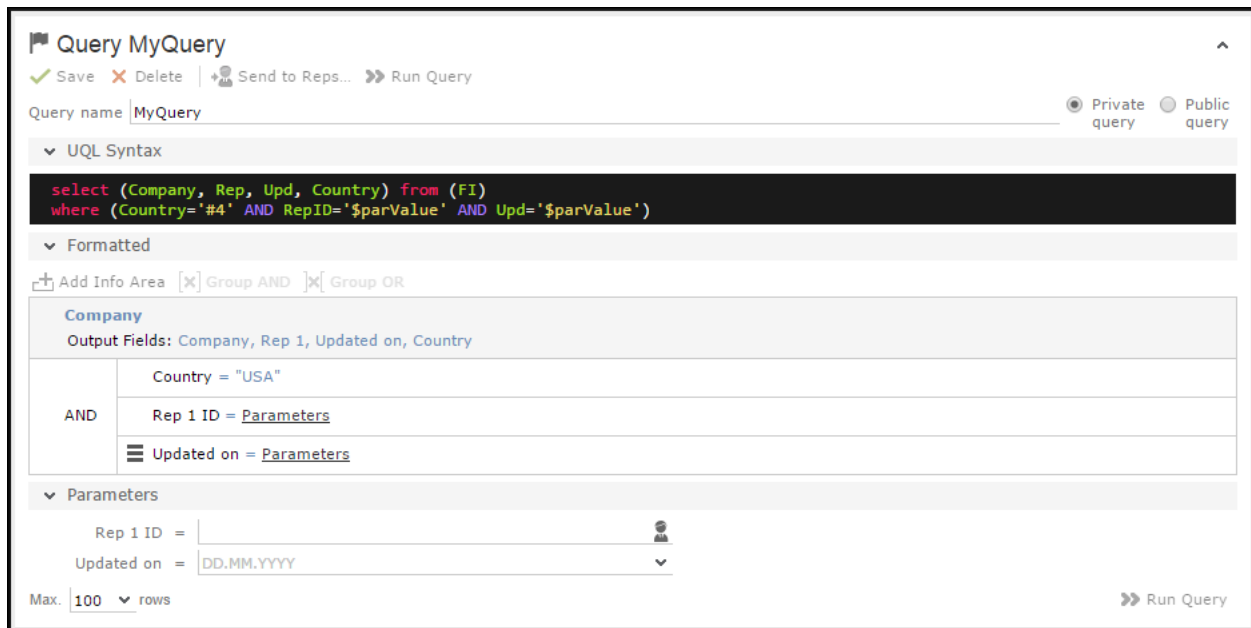
## Parameterized Queries

Parameterized queries require input at runtime.

Parameterized queries are queries that require user input. Use the placeholder `$parValue` for fields the user has to enter when executing the query. The screenshot below shows a new query “MyQuery” in the CRM.Designer.



In Aurea CRM.Web these fields are displayed as input fields in the respective query form Query Designer, QueryRun and QueryReport actions. See the screenshot below:



## Using One Value in Multiple Conditions

You can define parameterized queries where the same value is used by multiple conditions. The user only has to input the value once. Use placeholders e.g. called `$parValue<number>` for fields the user has to enter when executing the query. The value specified by the user is applied to all occurrences of `$parValue<number>`.

## Example

```

select (Company, Country, RepID, Rep2ID, Upd, KP.Sex, KP.FirstName, KP.LastName, KP.RepID, KP.Country, MA.Contact, MA.Date, MA.Time
from (FI)
where ((Upd>='$parValue2' or RepID='$parValue1' or Rep2ID='$parValue1') and Country='$parValue3')
plus (KP)
where (RepID='$parValue1' or Country='$parValue3')
plus (MA)
where (Upd>='$parValue2')
orderby (Company, KP.LastName, MA.Date, MA.Time)

```

The screenshot shows the AQL query editor with the following configuration:

- Company (FI):** Fields: Company (2), Country (5), Rep ID (24), Rep 2 ID (26), Updated on (4011), FI-StaNo (0). Sort: Company (2) (asc). Filter: Updated on (4011) >= \$parValue2, Rep ID (24) = \$parValue1, Rep 2 ID (26) = \$parValue1, Country (5) = \$parValue3.
- Person in Company (KP):** Fields: Form of address (0), First Name (3), Last Name (2), Account Manager 1 ID (14), Country (19). Sort: Last Name (2) (asc). Filter: Account Manager 1 ID (14) = \$parValue1, Country (19) = \$parValue3.
- Activity (MA):** Fields: Type (0), Starts on (2), Starts at (3), Updated on (4011), Status (1). Sort: Starts on (2) (asc), Starts at (3) (asc), Type (0). Filter: Updated on (4011) >= \$parValue2.

AQL syntax for the above example:

```

select (Company, Country, RepID, Rep2ID, Upd, FIKP.Sex, FIKP.FirstName,
FIKP.LastName, FIKP.RepID, FIKP.Country, FIMA.Contact, FIMA.Date, FIMA.Time,
FIMA.Upd) from (FI)
where ((Upd>='$parValue2' OR RepID='$parValue1' OR Rep2ID='$parValue1') AND
Country='$parValue3')
plus (KP as FIKP)
where (RepID='$parValue1' OR Country='$parValue3')
plus (MA as FIMA)
where (Upd>='$parValue2')
orderby (Company, FIKP.LastName, FIMA.Date, FIMA.Time)

```

Users only see one field per placeholder:

**Query FI.MultipleConditions**

**Enter Query Criteria:**

- Updated on (Company) >= 01.08.2014
- Rep 1 ID (Company) = Lisa Simpson
- Country (Company) = Italy

Max. 50 rows

**Query Result**

Company	Cou...	Rep 1 ID	Rep 2 ID	Updated on	F...	First Na...	Last Name	Rep ID	Country	Type	Starts on	Star...	Updated on
Fratelli Pasquini	Italy	Lisa Simpson		11.07.2014									
Gabrieli	Italy	Peter Parker		06.08.2014									
Merulo S.p.A.	Italy	Lisa Simpson		06.08.2014	Mr.	Girolamo	Frescobaldi		Italy				
Merulo S.p.A.	Italy	Lisa Simpson		06.08.2014	Ms.	Barbara	Strozzi	Lisa Simpson					
Query Test	Italy	John Doe	Lisa Simpson	06.08.2014	query	test	Lisa Simpson			Visit	20.08.2014 14:30		06.08.2014

The provided criteria are applied to all occurrences of the placeholder. Depending on the query definition (AND/OR, relation between info areas) the condition must be fulfilled by all fields using the same placeholder or not.

Make sure you do not apply a `$parValue<number>` placeholder to fields of different types. No validation takes place when saving the query.

Specifying `$parValue<number>` for a hierarchical catalog automatically adds an input field for the parent catalog. However, using the same `$parValue<number>` for parent and child catalog results in an error.



---

**Note:** The placeholder `$parValue` is not affected by this feature, i.e. each occurrence of `$parValue` creates an extra input field for the user.

---

## Background Queries

You can execute complex queries in the background and continue working on Aurea CRM.

By clicking  **Run Query in Background** users can execute queries asynchronously in the background in addition to executing queries immediately via  **Run Query**. When running a query in the background, the user can continue to work in Aurea CRM.Web. The query is executed on the server. When finished the user gets a notification that he can now access the query snapshot.

---

**Note:** Executing queries in the background is recommended for long-running complex queries.

---

To enable the processing of background queries, CRM.server must be up and running on the server. For further details, see [CRM.Server](#) on page 15.

The rep ID of the CRM.server user must be specified in the `UpdateServer.RepID` Web Configuration parameter because the to-dos for background processing are created for that rep. For further details, see [UpdateServer.RepID](#).

The Web Configuration parameter `Jobs.PollingInterval` defines the time interval in which the existence of new query snapshots is periodically checked. For further details, see [Jobs.PollingInterval](#).

The Web Configuration parameter `Query.BackgroundExecutionStart` allows you to specify a time in the format `hh:mm` for asynchronous queries to be processed on the server. For further details, see [Query.BackgroundExecutionStart](#). That way the processing of background queries can be scheduled for off-peak hours in case performance issues arise.

CRM.server is notified about new background queries via **To-Do** records that are automatically created by Aurea CRM.Web. The query is saved as a D1 document (\* .xml) linked to the to-do record. CRM.server processes the query with the rights of the rep that has sent the query and then saves the result in a query snapshot, and finally creates a **Message** record for the user. The message contains a link to the **To-Do** which now has the status "completed". The to-do contains a link to the query snapshot. The XML document is automatically deleted after completion.

Users can access the to-dos created for background queries via **Queries & Analyses > Background Queries**. For further details, see [ID\\_NET\\_TD\\_BACKGROUND\\_QUERIES](#).

In UPDATE\_DEFAULT the special headers `Query` and `QueryFooter` contain the **» Run Query in Background** button.

If you add the **» Create CSV File in Background** button to those headers, users can create a CSV file containing the query results. The file is stored in the **Document** info area (D1) and linked to the **Message** via a D3-link.

---

**Note:** With **Create CSV File in Background** the resulting records are **not** stored as a query snapshot but in the CSV file only.

---



---

**Note:** If the created CSV file exceeds a certain size, the decompression of the file (when transferring it from the database to the client) is not performed in memory but via the file system.

---

For information on the background query buttons and their configuration. For further details, see [Query Background Processing Buttons](#) on page 318.

## Global Definition of FindAsYouType

The FindAsYouType definition is used for all link fields that point to the respective info area.

You can define FindAsYouType globally by defining a query `<InfoAreaId>.FindAsYouType`. Examples in UPDATE\_DEFAULT: `FI.FindAsYouType`, `MA.FindAsYouType`, `AR.FindAsYouType` etc. This FindAsYouType definition is used for all link fields that point to the respective info area and for which no `Select-Function` has been defined. For further details, see [Select-Function](#) on page 240.

---

**Note:** FindAsYouType only supports text, number and variable catalog fields as search fields (Field Types C,L,S,K). For further details, see [Data Types in Aurea CRM](#) on page 877). Fields that are not supported are ignored and a warning is written to the client log. For output fields there is no restriction: All field types are valid.

---

---

**Note:** Link fields to which a global or specific **FindAsYouType** definition applies, are displayed as editable. If an invalid/erroneous FindAsYouType definition is defined for a field, this field is displayed as read-only.

---

**Note:** FindAsYouType is only supported for standard link fields, i.e. not for generic links.

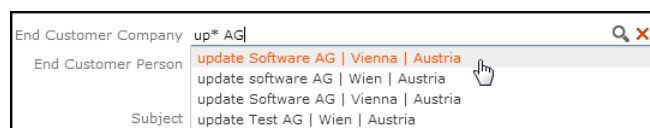
---

Text and catalog fields support a substring and wildcard search. Number fields only return exact matches, i.e. searching for "100" does not return "1000" or "3100". To search for number ranges you need to define a query with condition. For example, `where (Revenue>'$parValue')`.

### Example

```
select (Company, City, Country) from (FI)
where (Company="$parValue" OR Synonym="$parValue" OR City="$parValue" OR
Street="$parValue")
orderby (Company)
```

This FindAsYouType definition (`FI.FindAsYouType` in `UPDATE_DEFAULT`) is used e.g. in the **End Customer Company** field of the **Ticket (KM)** info area (if no other `Select-Function` has been defined for this field). If the user types part of a company's name and/or address in this field, a drop-down list containing all hits in the form **Company | City | Country** is displayed.



## Advanced Filter Criteria

Here is the list of Advanced Filter Criteria that you can define in Aurea CRM.Web.

The following advanced filter criteria are available:

### Comparing Field Values

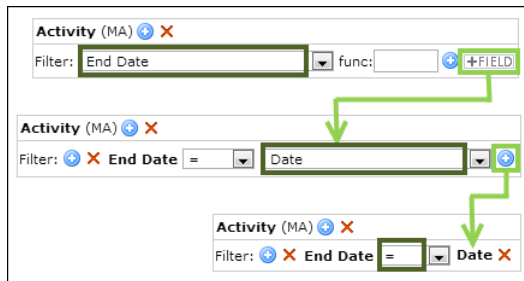
Filter criteria to compare field values.

You can compare the values of different fields within one info area provided these fields have the same data type. For further details, see [Data Types in Aurea CRM](#) on page 877.



To compare the values of different fields, do the following:

1. Select a field from the drop-down list.
2. Click **+FIELD** (**Add Field**).
3. Select the second field.
4. Click **+** (**Add**).
5. Select the desired operator.



The following comparisons are possible:

- date fields with date fields
- time fields with time fields
- rep fields with rep fields
- numerical fields with numerical fields
- text fields with text fields
- logical fields with logical fields
- catalog fields only if two or more fields are associated with the same catalog

**Note:** CRM.Designer does not perform any checks whether the compared fields are of the same type.

## Using Variable Expressions

Use of variable expression when defining filter criteria.

You can use variable expressions when defining filter criteria:

1. Select a field from the list
2. Click **+** (**Add**).
3. Select an operator.
4. Enter the desired variable in the field after the operator.
5. Click **+** (**Add**).

The following variable expressions are available:

- `$curDay`: today's date
- `$curfdYear`: the first day of the current year
- `$curfdQuarter`: the first day of the current quarter
- `$curfdMonth`: the first day of the current month
- `$curfdWeek`: the first day of the current week
- `$curYearFY`: the first day of the current business year
- `$curLastLogin`: the date of the current rep's last login

---

**Note:** `$curLastLogin` only works in Aurea CRM.Web, not in Aurea CRM win, and can therefore not be used for queries that are processed by CRM.server.

---

You can add or subtract days, weeks etc. from these dates, e. g. `$curDay+1y+1d` results in tomorrow one year from now. Available abbreviations: d (day), w (week), m (month), q (quarter), y (year).

- `$curTimeMin`: the current time in format HH:MM
- `$curTimeSec`: the current time in format HH:MM:SS
- `$curTimeMs`: the current time in format HH:MM:SS:mmm

---

**Note:** `$curTimeMin`, `$curTimeSec` and `$curTimeMs` can only be used depending on the time field's format. For further details, see [Aurea CRM Field Types](#) in the *CRM.Core Administrator Guide*.

---

- `$curRep`: the currently logged in rep
- `$curOrgGroup`: the **Org.group** of the currently logged in rep
- `$curGroup`: the **Org.group** plus group allocation(s) of the currently logged in rep
- `$curAllGroups`: Returns organizational group and all groups allocated to current logged in rep (without checking field datatype).

---

**Note:** `$curGroup` returns only organizational group if "supplied db field" is singular. `$curAllGroups` returns all group assignment & allocations in all conditions and ignores the field details in its query.

---

- `$curMyLeader`: the group leader of the **Org.group** of currently logged in rep

- `$curLeader`: all groups where currently logged in rep is group leader
- `$curMyDeputy`: the deputy of the currently logged in rep
- `$curDeputy`: all records where the currently logged in rep is the deputy
- `$curSuperior`: the supervisor of the currently logged in rep
- `$curSupLine`: the current superior line
- `$curExtGroup`: all groups in the rep's extended group, i.e. the group entered in the **Org. Group** field and all groups directly allocated to this group in the **Group Allocation** info area.

---

**Note:** `$curGroup`, `$curExtGroup`, `$curLeader` and `$curDeputy` cannot be used for default values since they potentially return more than one result. These variables are currently also not available for defining conditions in Aurea CRM.Web.

---

- `$curTenant`: the tenant of the currently logged in rep
- `$cur(<infoAreaId>,<fieldId>)`: You can access any field value of the current rep from the info areas **Rep** (ID) and **Station** (AS), as well as the info area **Rep Additional Info** (AZ) in the OTC vertical.

---

**Note:** Currently only rep fields are supported.

---

**Example:** `RepID = $cur(ID,72)` compares the content of the field **RepID** (e.g. in MA) with the superior of the currently logged in user as defined in the **Rep** info area.

In addition, you can use custom variables defined in the **Variables** info area (Z5) for defining filter criteria. For further details, see [Variables](#) on page 53.

**Syntax:** `$VAR(<variable name>)`

For more information, see the article “Current Expressions” at <https://support.aurea.com>.

## Using Wildcards in Date Fields

Using wildcards for defining conditions on date fields.

When using wildcards for defining conditions on date fields use the Aurea CRM format `YYYYMMDD` (e.g. “????0523”).

You can also define filters and queries for the current day, month or year. In this case, use the format `$YYYYMMDD`. Possible placeholders are # for the current date or ? for any date.

---

**Note:** You can use literal character values of ‘\*’ and ‘?’ using the escape sequences ‘\\*’ and ‘\?’.

---

## Examples

To get the current day and month regardless of year, use `$$$?##`. For Example - This filter can be used to determine all persons whose birthday is today. The filter `$$$?##??` could be used to find everyone whose birthday is in the current month.

You can also add or subtract values like you can for `$curDay`: Define `$$$?###+2w` to find people whose birthday is in 2 weeks.

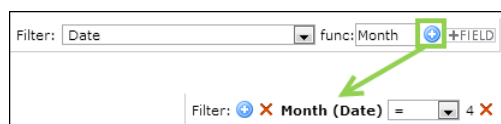
**Note:** It is also possible to perform wildcard searches like "date\_of\_birth>\$\$\$?## AND date\_of\_birth<\$\$\$?###+2w" because this structure cannot be represented in AQL.

## Using Date Functions

Defining conditions on date fields to evaluate specific properties of a date.

You can define conditions on date fields that evaluate specific properties of a date, like the weekday. To do this, follow the steps below:

1. Select a field from the list
2. Enter the desired function in the **func:** field.
3. Click + (**Add**).
4. Select an operator.
5. Enter the desired value in the field after the operator.
6. Click + (**Add**).



The following date functions are available when defining filter criteria in CRM.Designer:

- `Day`: Returns the day of the date (1...31)
- `Month`: the month of the date (1...12)
- `Year`: the year of the date
- `Quarter`: the number of the quarter the date is in (1...4)
- `Week`: the number of the date's calendar week (1...53)
- `Weekday`: the weekday of the date represented by a number (1=Monday, ..., 7=Sunday)
- `Weekend`: 1 if the date is on a weekend (i.e. Saturday or Sunday), otherwise 0

---

**Note:** Date functions cannot be used in conjunction with variable expressions like `$curDay`, `$curMonth`, etc. However, a filter to check e.g. if the month of a date field is equal to the current month can easily be realized using `?????##??` as the comparison value (for the operators "=" and "<>" only). For further details, see [Using Wildcards in Date Fields](#) on page 507.

---

## Dynamic Filters

This section lists the configuration units/areas where Dynamic Filters are available.

Configure dynamic filters to filter search results based on values retrieved from the parent record. For Example - to filter the item list of the offer quick add page based on a value from the offer.

Filter criteria from the dynamic filter and additional search criteria defined by the user are combined by a logical AND.

You can define dynamic filters for linked searches, record selectors and generic quick add pages. For further details, see [Examples](#).

## Configuring a Dynamic Filter

Learn to configure a dynamic filter.

To configure a dynamic filter:

1. Create a **field group** for the source info area, i.e. the info area from which the filter values are retrieved.
2. Create a List control and add the field(s) you want to use as filter values.
3. Enter a parameter name as **Function**. (You can enter any text (without spaces), but best practice is using the field's XML name.)
4. Define the **filter**.
5. Add the above defined parameter as filter value (using the syntax `$par<name>`).

---

**Note:** Date/time calculations with `$par` expressions are currently not supported.

---

6. Add the field group and filter to the **action** configuration:

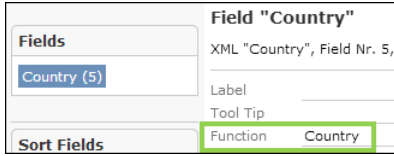
- the field group as `sourceCopyFieldName`
- the filter as `fixedFilterName`.

For more details, see the examples below:

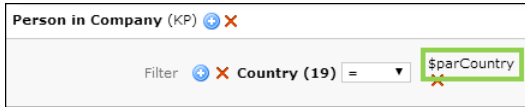
### Search

Searching for persons with the same **Country** as the parent company via a button in the `FI` Expand header:

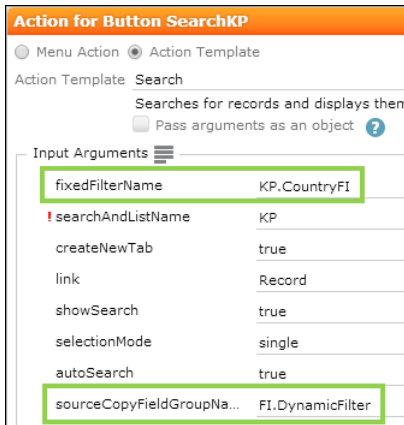
Field group for `sourceCopyFieldGroupName`: e.g. `FI.DynamicFilter` - List control:



Filter definition for `fixedFilterName`, e.g. `KP.CountryFI`:



Search action defined for a new button in the `FI` Expand header:



In this case, the parent record is defined by `link`. You can use the `sourceUId` parameter to pass another parent.

Clicking on the button opens a linked search that automatically displays only linked KP records with the same country as the parent FI record.

If the user enters additional search criteria, they are combined by logical AND.

### Related Data

The above example can also be implemented for sub-lists:

`FI SearchSublist` and `ExpandChildren` headers



Enter `sourceCopyFieldGroupName`, `FI.DynamicFilter` as `SubListParams`.

The KP sub-lists for FI are filtered by the country of the parent company.

Examples in `UPDATE_DEFAULT`:

- The **Text** (04) sub-lists display all available translations of a text record. Configured in the `SearchSublist` and `ExpandChildren` headers of header group 04 with `Filter` `04.BaseLanguageFilter` using field group `04.AllTexts`.
- The **Station configuration** (T0) sub-list contains only the record defined for the selected station (AS). Configured in the `SearchSublist` and `ExpandChildren`

headers of header group AS with Filter T0.ForStation using field group AS.ParametersForT0.

## Quick Add

Generic quick add page for offers - filtering the item list by the parent record's **Product Group**:

Field group for sourceCopyFieldGroupName: e.g. PR.DynamicFilter - List control:

Field "Product Group"	
<b>Fields</b>	XML "ProdGrp", Field Nr. 83, T
Product Group (83)	Label
	Tool Tip
<b>Sort Fields</b>	Function ProdGr

Filter definition for fixedFilterName, e.g. AR.ProdGrPR:

Item Master (AR)	
Filter	Product Group (3) = \$parProdGr

SerialEntry action e.g. defined for A\_SerialEntryAP:

Action for Menu A_SerialEntryAP	
Action Template	SerialEntry
Serial entry page.	
<input type="checkbox"/> Pass arguments as an object	
<b>Input Arguments</b>	
name	Offer(AR=>AP)
uid	Record
optionsFormName	SerialEntry.Default.Options
copySourceFields	ItemNo,ItemName,Departm
copyTargetFields	ItemNo,ItemName,Departm
autoSearch	true
fixedFilter	AR.ProdGrPR
recentFilterOptions	FI,OfferDate,3
sourceCopyFieldGroupNa...	PR.DynamicFilter

The item list displays only items (AR) with the same product group as the parent offer (PR).

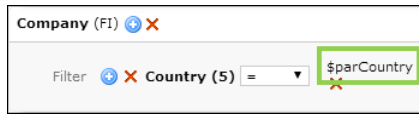
## Record Selector

Record selector for the **Event Location** field in the **Marketing Activity** (AK) info area - filtering for companies with the same **Country** as the marketing activity:

Field group for sourceCopyFieldGroupName: e.g. AK.DynamicFilter - List control:

Field "Country"	
<b>Fields</b>	XML "Country", Field Nr
Country (26)	Label
	Tool Tip
<b>Sort Fields</b>	Function Country

Filter definition for fixedFilterName, e.g. FI.CountryAK:



Select-Function defined for the **Event Location** field in the **AK Details** control:

```
selectRecord(
{
  from:
  {
    infoAreaId: "FI",
    useAutoSearch: true,
    fixedFilterName: "FI.CountryAK",
    sourceCopyFieldGroupName: "AK.DynamicFilter"
  }
});
```

Clicking on the record selector icon  opens a linked search that automatically displays only FI records with the same country as the current AK record.

If the user enters additional search criteria, they are combined by logical AND.

## Dynamic Filtering in the Tree View

Learn to configure a dynamic filter in the Tree View.

You can filter the child nodes displayed based on the field value of a parent node. To do this you have to perform the first four steps as described in [Configuring a Dynamic Filter](#) on page 509. Then set the parameter `$relationNameAndParameters` in the **Relation Name** field in the **Info Area Settings** of the child node. This parameter takes the relation name and the list of field values to compare with the parent info area. It can take any number of field values to compare. You can also pass linked parent records field values into the **RecordFilter** as parameters.

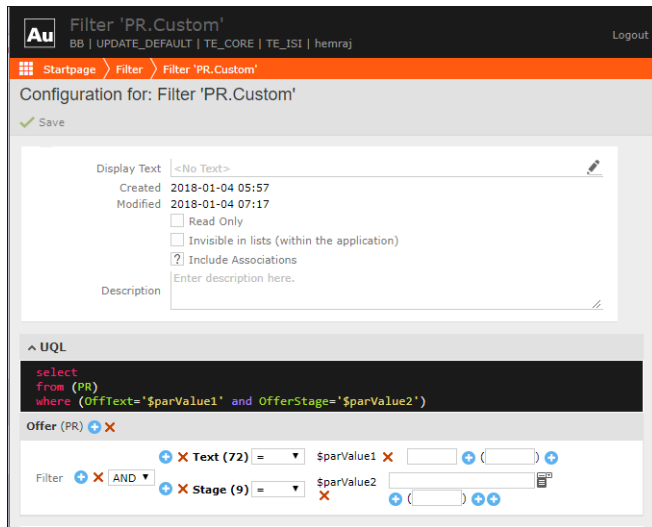
The following show the sample usage of the `$relationNameAndParameters` parameter:

- `$relationNameAndParameters=[, 51, 37]`—empty relation name and the field numbers as parameters.
- `$relationNameAndParameters=[$RelatedCompany, 51, 37]`—parentcompanyrelation name and the field numbers as parameters.

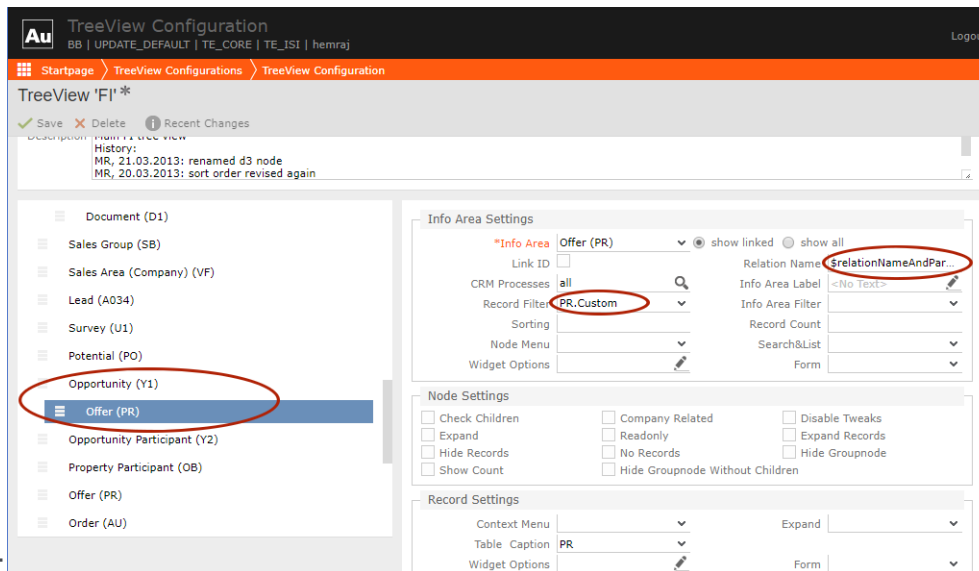
The following example shows the configuration of a dynamic filter to show only the Offer Item that is part of an Offer to a company. This ensures only the Offer Items linked to the Company record is be displayed under the Opportunity node and not all the offers made to the parent company.

1. Add a new filter for the PR tree node named **PR.Custom**.





2. Set the PR tree node under Opportunity Y1 as shown in the screenshot



below:

3. Set the **Record Filter** field to `PR.Custom` as shown in the screenshot above.

**Note:** You can also pass linked parent records field values into `RecordFilter` as parameters.




4. Set the **Relationship Name** field to the parameter `$relationNameAndParameters=[$RelatedCompany,51,37]` as shown in the screenshot in step 2.

## Analyses

Here is a guide on how to define analyses in update.CRM web.




























This topic describes the configuration of all elements related to Analyses in CRM.Designer. Analyses allow to aggregate query data, ad hoc, at the time of their execution and display graphics Example - the **Sales Funnel**.

To define/customize analyses in CRM.Designer:

1. Go to CRM.Designer main page.
2. Click **Analyses**  to display the **Analyses** page.
3. Click  to view an **Analyses**. Click  to edit an **Analyses**







**Analyses**

Name  Copy from Empty Analysis New

Name	Configuration
Company Buying Frequency	  my_config
Company Contact Frequency (last year)	  my_config
Contact and Buying Frequency	  UPDATE_DEFAULT
FI_Country_Revenue	  my_config
my_private_analysis	  my_config
my_public_analysis	  customized_parent_config
newTest	  UPDATE_DEFAULT
openissues	  UPDATE_DEFAULT
Opportunities_CurrentQuarter	 BASE/UPDATE_DEFAULT
OpportunityBubble	 BASE/UPDATE_DEFAULT
Questionnaire_SurveyAnswers	 BASE/UPDATE_DEFAULT
Questionnaire_SurveyResults	 BASE/UPDATE_DEFAULT
SalesFunnel	  UPDATE_DEFAULT
ServiceFocusProducts	  UPDATE_DEFAULT
ServiceWorkload	  UPDATE_DEFAULT
TicketBubble	 BASE/UPDATE_DEFAULT















**Explicit Analysis-Categories**

Name  Copy from Empty Category New

Name	Configuration
DateMonthly	  UPDATE_DEFAULT
DateQuarterly	  UPDATE_DEFAULT
Percentage_10Sliced	  UPDATE_DEFAULT

**Analysis Functions**

Name  New

Name	Configuration
BETWEEN	  UPDATE_DEFAULT
COUNTALL	  UPDATE_DEFAULT
CURRCONV	  UPDATE_DEFAULT
MIN	  UPDATE_DEFAULT
STATIC	  UPDATE_DEFAULT
SUMCAT	  UPDATE_DEFAULT
SUMCATX	  UPDATE_DEFAULT

Use the **SDK** page to explore the functionality of all analysis-related widgets: **SDK > Analysis**.

## Storing Analyses

This topic has information about storing of analyses.

Analyses are always saved in the CRM.Designer database. For further details, see [Format Types for Queries and Analyses](#) on page 492. Public analyses defined in Aurea CRM.Web are stored in the highest writable parent configuration.

---

**Note:** If the parent configuration of the user configuration is read-only, the user gets an error message when trying to save a public analysis.

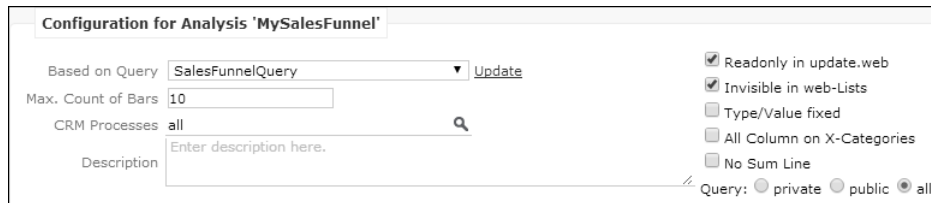
---

## Defining Analyses

Learn to define analyses in CRM.Designer.

To define an analysis in CRM.Designer:

1. Select an analysis or create a new one. For further details, see [Designing Configuration Units](#) on page 64.



2. Specify the following:

- **Based on Query:** Defines which query is executed for the analysis. This query is read-only if it is stored in the Aurea CRM database. It is recommended to store your queries in the Aurea CRM database (default). For further details, see [Storing Queries](#) on page 495. Click **Update** to load the latest version of the query.
- **Max. Count of Bars:** Defines how many bars (or pie segments) are contained in the analysis. If the selected category results in more values than defined in the **Max. Count of Bars**, the bars with the smallest values are grouped into an "other" value (unless otherwise specified in the definition for the category, i.e. **NoOther** option checked).
- **CRM Processes:** Assign one or more CRM processes to the analysis. For further details, see [Roles & CRM Processes](#) on page 727.
- **Description:** Add an internal description only visible in CRM.Designer.
- **ReadOnly in update.web:** If checked, users cannot edit the analysis in Aurea CRM.Web.
- **Invisible in web-lists:** If checked, users cannot select the analysis in Aurea CRM.Web.

---

**Note:** Use the **Read-only** and **Invisible** settings for analyses called via menu actions etc. Alternatively, you can use the `Analysis.Rights` Web Configuration parameter to determine whether users can execute and/or edit predefined analyses. For further details, see [Analysis.Rights](#).

---

- **Type/Value fixed:** If checked, the user cannot change the predefined result columns in Aurea CRM.Web.
- **All Column on X-Categories:** Check to display an additional column containing the sum total for additional categories. For further details, see [Using an Additional Category on the x-Axis](#).

- **No Sum Line:** Hides the sum total row in the analysis results.
  - **Query:** Defines where the query is searched depending on the format type. For further details, see [Format Types for Queries and Analyses](#) on page 492. Set it to "All" for predefined analyses.
3. Define the category (x-axis), value (y-axis) and filter fields for the analysis. The fields are taken from the selected query's output fields.

Field Name	CriteriaField	ValueField	AdditionalCriteria	ReadAll	DependentOnCurrency	WeightedField	DoNotSortBars	ShowAllBars	NoOther	MustSelect	AlternateCurrencyField	CategoryNotMandatory	XCategory	Defaultvalue	Default-End	Slices	Category	List-Nr.	List-Width		
Company (1)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>						4	10	%	
Country (5)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>									%
Opportunity (1)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>						1	15	%	
Level (9)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>						2	5	%	
Rep ID (12)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	\$curRep					3	10	%	
Group ID (14)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>									%
Priority (15)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>									%
Status (16)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>									%
ABC (17)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>									%
Completion Date (43)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	\$curDay	\$curDay+6m				5	15	%	
Total Prod. Value (145)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>						6	10	%	
Wtd. Total Prod. Value (146)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>									%
Opp. Wtd. Total Prod. Value (147)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>									%
Currency (4005)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>						7	10	%	

CriteriaDefault: Level (Opportunity)  
 Weight: Realization Probability (%) (Opportunity Product)  
 Currency: Currency (Opportunity)

- **Category fields (x-axis):** Check **CriteriaField** for all fields you want to use to aggregate data by (e.g. **Country**). These fields can be selected from the Category drop-down list and used for drill-down.

Category fields can be of type character, date, catalog or Boolean. If you select a date field, the data is not grouped by its exact value. The date range is sliced into a fixed amount of result rows, each containing a date range of the same size.

Specify additional settings if applicable. For further details, see [Settings for Category Fields](#) on page 519.

- **Value fields (y-axis):** Check **ValueField** for all fields that can be aggregated (e.g. **Total Prod. Value**). Only numeric fields can be selected.

Specify additional settings if applicable. For further details, see [Settings for Value Fields](#) on page 521.

- **Filter fields:** Check **AdditionalCriteria** for fields used to filter the analysis result (e.g. **Completion Date**). Filter fields can be of type character, date or catalog.

Specify additional settings if applicable. For further details, see [Settings for Filter Fields](#) on page 521.

#### 4. Specify defaults:

- **CriteriaDefault:** Select the default category field.
- **Weight:** Select the (percentage) field used for weighted values (defined by the **WeightedField** option).
- **Currency:** If your analysis contains currency-dependent value fields (defined by the **DependentOnCurrency** option), select a currency field. The currency specified in this field is used for the analysis. If the selected currency field contains another than the destination currency, values are converted using the currency conversion rate stored in the Aurea CRM database.

If the query contains more than one currency field, you can select different currency fields. These fields depend on the record containing the currency-dependent field. Select the **AlternateCurrencyField** option for a field of that info area. For example, the `SalesFunnel` analysis defined in `UPDATE_DEFAULT` has currency fields and currency dependent-fields in the **Opportunity** record and in the **Opportunity Product** record. The currency field in the **Opportunity Product** record is defined as **AlternateCurrencyField** and is used for currency-dependent value fields instead of the **Opportunities** currency field. Do not use more than one currency in an opportunity.

#### 5. Configure the list view of the analysis result.

Results (2 record(s))										
Opportunity	...	Rep ID	Comp...	Completion ...	Total ...	Curre...	Produ...	Prod....	...	...
Test8	Lev...	Miranda M...	Krist Aut. ...	23.01.2013	344.000,00	Euro				
Sell A Lot	Lev...		Eco Hotel ...	28.02.2013	30.000,00	Euro				

For each field you want to display in the results list define the following:

- **List-Nr.:** The position of the column in the list.
- **List-Width:** The column width in percent.

If undefined, no results list is available (menu item **Display Details** hidden).

#### 6. Define the default result columns for the analysis in the **Default Values** area:

Default Values	
Type	Column
↑ ↓ ✖ Average	Total Prod. Value (Opportunity)
↑ ↓ ✖ Sum	Opp. Wtd. Total Prod. Value (Opportunity)
↑ ↓ ✖ Sum	Wtd. Total Prod. Value (Opportunity)
⊕ Sum	Currency (Opportunity Product)

- Select the aggregation type (Sum, Count, Average, Minimum, Maximum).
- Select the desired value.
- Click + (**Add**).

The sum total row at the bottom of the results column in Aurea CRM.Web displays the total result according to the selected aggregation type (e.g. the sum total for "Sum", the highest maximum value for "Maximum" etc.

In the sum total row each result record is taken into account only once even if it is used in several result rows, i.e. if a category field from a child record is used. Example in UPDATE\_DEFAULT: `SalesFunnel` analysis - for Count (Opportunity) per Y5-Product Group the sum total does not necessarily have to correspond to the sum of the values of this column.

Results (10 record(s))		
Product Group	Count Opportunity	Average Total Prod. Value
#null	12	403.388,00
Laptops	1	9.150,00
Software	1	9.150,00
mobile	1	9.150,00
pda	1	9.150,00
phone	1	9.150,00
win	1	9.150,00
Bücher	2	4.575,00
Holzfiguren	2	4.000,00
Essen	1	0,00
	<b>17</b>	<b>285.753,29</b>

---

**Note:** The sum total row is not available in the drill-through details view.

---

**Note:** Default values are also used if **Type/Value fixed** is set for the analysis, or the displayed analysis in Aurea CRM.Web has no controller.

---

**Note:** If you remove a field you use as default value from the query, the reference is lost: ✖ Sum      ?0/1036

---

- Define additional values (calculations, parameters) if applicable. For further details, see [Additional Values](#) on page 522.
- Click **Save**.

To display an analysis in Aurea CRM.Web, create a menu action or include the analysis in a form e.g. a dashboard (`AnalysisResult` widget). For further details, see [AnalysisRun](#) on page 395 and [Forms](#) on page 621.

---

## Settings for Category Fields

Learn about the settings for category fields.

You can define the following additional settings for category fields **CriteriaField**:

- **DoNotSortBars**: Disables sorting the result rows in descending order. Used for date criteria where the date ranges are displayed in chronological order and for catalog values that have a chronological significance like opportunity levels.
- **ShowAllBars**: Used in conjunction with **DoNotSortBars** to display all bars, even if the value of the bar is "0". **ShowAllBars** disables the **Max. Count of Bars** setting.
- **NoOther**: Hides the "other" value (if **Max. Count of Bars** < available values). The values from "other" are taken into account in the sum total line however.
- **XCategory**: Defines a category field as additional category to be displayed on the x-axis, e.g. to display revenue data split by country in one analysis result instead of drilling down. For further details, see [Using an Additional Category on the x-Axis](#).
- **Slices**: Only valid for date and numeric fields. The value entered here overrides the **Max. Count of bars** default setting when aggregating the analysis by this column. (Use this setting to create e.g. quarterly slices of results. Alternatively, you can define Explicit Analysis-Categories. Example in UPDATE\_DEFAULT: `Percentage_10Sliced`. For further details, see [Explicit Analysis Categories](#) on page 527.)

---

**Note:** The "slicing" of date and numeric results only works if no controller is displayed in the analysis.

---

- **Category**: Applies an explicit analysis category to the field. Example - to split dates by year. For further details, see [Explicit Analysis Categories](#) on page 527.

### Using an Additional Category on the x-Axis

It is possible to specify an additional category that is used on the x-axis of an analysis. To do so,

- Check one second field in the **XCategory** column in the analysis definition.
- If you check **All Column on X-Categories**, an additional column containing the sum total for each value is displayed in the analysis.
- If **ShowAllBars** is checked, all values of the catalog or the explicit category are displayed whether they contain a value or not.
- If **DoNotSortBars** is checked, the sort order is determined by the order of the explicit category values or the catalog code. If these are not applicable, the sort order depends on the displayed text.

**Note:** This functionality is supported for catalogs and explicit categories, for all other field types field-specific conversions (e.g. from rep to rep id) are not implemented.

**Note:** Additional values are currently not supported as additional x-categories.

### Examples

Company analysis with values "Sum Revenue" and "Count Companies" per **Sector** and **Country**:

**Configuration for Analysis 'MyFICountrySector'**

Based on Query:

Max. Count of Bars:

CRM Processes:

Description:

Readonly in update.web  
 Invisible in web-Lists  
 Type/Value fixed  
 All Column on X-Categories  
 No Sum Line  
 Quadrant analysis only  
 Query:  private  public  all

Field Name	ValueField	CriteriaField	AdditionalCriteria	ReadAll	DependentOnCurrency	WeightedField	DoNotSortBars	ShowAllBars	NoOther	MustSelect	AlternateCurrencyField	Category/NotMandatory	XCategory	Defaultvalue	Default-End	Slices	Category List
Company (1)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
Company (2)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
Country (5)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>				
Rep ID (24)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
ABC (35)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
Revenue (36)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
Employees (37)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
Lead Status (38)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
Sector (29)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				

no Infoarea selected

CriteriaDefault:

Weight:

Currency:

**Default Values**

Type	Column
↓ Sum	Revenue (Company)
↑ Count	Company

Result in Aurea CRM.Web:

The header is split in two rows listing the values per sector (in the "all" column) and per country.



Sector	Sum Revenue						Count Company		
	all	Austria	Croatia	Ge...	UK	Л	all	Aust...	Cr...
Potatoes	8.508.566,00	8.508.566...	0	0	0	0	1	1	0
no value	5.800.000,00	0	0	5.800...	0	0	1	0	0
Shipping Agent	999.999,00	0	0	0	999...	0	1	0	0
Tourism	204.711,00	204.711,00	0	0	0	0	2	2	0
Mechanical Engineering	200.000,00	0	200.000,00	0	0	0	1	0	1
Software	29.300,00	29.300,00	0	0	0	0	3	3	0
Sector 1	20.000,00	0	0	0	0	20...	1	0	0
Software retailer	20.000,00	20.000,00	0	0	0	0	1	1	0
Construction	10.000,00	10.000,00	0	0	0	0	1	1	0
	<b>15.792.57...</b>	<b>8.772.5...</b>	<b>200.000,00</b>	<b>5.80...</b>	<b>999...</b>	<b>2...</b>	<b>12</b>	<b>8</b>	<b>1</b>

Company analysis with values "Sum Revenue" and "Count Companies" per country and half-year (using an explicit date category applied to **FreeD1**, see [Explicit Analysis Categories](#) on page 527).

Field Name																				XCategory			
Company (1)																				Defaultvalue	Default-End	Slices	Category L
Company (2)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
Country (5)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
Rep ID (24)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
ABC (35)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
Revenue (36)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
Employees (37)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
Lead Status (38)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
FreeD1 (82)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>				DateRange
Sector (29)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				

no Infoarea selected

CriteriaDefault Country (Company)

Result in Aurea CRM.Web:

Cou...	Sum Revenue			Count Company		
	all	H1 2012	H2 2012	all	H1 2012	H2 2012
Austria	507.300,00	500.000,00	7.300,00	2	1	1
USA	500.000,00	500.000,00	0	1	1	0
UK	4.711,00	0	4.711,00	1	0	1
	<b>1.012.011,00</b>	<b>1.000.00...</b>	<b>12.011,00</b>	<b>4</b>	<b>2</b>	<b>2</b>

## Settings for Value Fields

Learn about the settings for Value fields.

You can define the following additional settings for value fields (**ValueField**):

- **DependentOnCurrency:** Indicates a currency-dependent value and must be converted if the resulting currency does not match the currency in the selected **Currency** field.
- **WeighedField:** Indicates that the value must be weighted using the percentage value specified in the **Weight** field.

## Settings for Filter Fields

Learn about the settings for filter fields.

You can define the following additional settings for value fields (**AdditionalCriteria**):

- **ReadAll:** Even if the user selects a value for a criterion, the criterion is not sent to the server but evaluated on the client. Therefore, if the analysis was executed

once, and the criteria are changed afterwards, these changes do not result in another request to the server.

- **MustSelect:** Defines a mandatory additional criterion forcing the user to select a value to run the analysis.
- **Defaultvalue:** Defines the default value for an additional criterion. You can use the following variable expressions:
  - \$curRep
  - \$curDay
  - \$curWeek
  - \$curMonth
  - \$curYear

You can add or subtract days, weeks etc. from these dates, e. g. \$curDay+1y+1d results in tomorrow one year from now. Available abbreviations: d (day), w (week), m (month), q (quarter), y (year).

**Note:** For catalog values, you can either specify the catalog code(Example- 2) or \$<WebConfigParameterName> if you want to use catalog synchronization. For further details, see [Modifying Web Configuration Parameters](#).

- **Default-End:** Used to select the default end date in the date criterion. Supports the same variables as for **Defaultvalue**.

### Additional Values

Learn the use of additional values in defining Analyses.

Additional values are used to calculate the value(s) of one or more fields of a record.

Additional Values				
Valuename	Valuetype	Parameter	Display-Text	Fixed Stat-Type Options
<input type="text"/>	Computed	<input type="text"/>	all	<input type="text"/>

To define an additional value, specify the following:

- **Valuename:** The additional value is referenced with this name, e. g. in the drop-down list for default values.
- **Valuetype:** The type of the additional value. At present, only "computed" additional values are available.
- **Parameter:** The formula to calculate the value. Use \$<infoAreaID>[<field name or number>] to reference field values.

Example: \$Y1[FreeN1]\*2 multiplies field **FreeN1** of info area Y1 by 2.

Additional Values				
Valuename	Valuetype	Parameter	Display-Text	Fixed Stat-Type Options
AsValue	computed	\$Y1[FreeN1]*2	FreeN1*2	IsColumn

Level	Sum Total Prod. Value	Sum FreeN1*2
Level 0	2.503.000,00	1.800,00
Level 1	109.350,00	788,00
Level 2	551.000,00	376,00

The field must be part of the underlying query. If the query contains a field multiple times, reference it using `$Y1 [0] [FreeN1]` and `$Y1 [1] [FreeN1]`.

For more information on the syntax and the available functions. For further details, see [Analysis Functions](#) on page 533. For more examples, see [Examples](#) on page 525.

- **Display-Text:** An optional language-dependent text used in the value selection box in Aurea CRM.Web for this additional value. If undefined, **Valuename** is used.
- **Fixed Stat-Type:** Allows you to specify a fixed calculation type ("Sum", "Count", "Average" etc.). This disables the Type selection in Aurea CRM.Web (in the **Add a value** dialog).
- **Options:** Allows you to define options for the additional value. Separate multiple options by semicolon.

The following options are available:

- **Format:** An optional format definition. Use # for the value and `, <number>` for the desired number of decimal places. You can also add any desired character you want to display.

If no formatting is specified, # (i.e. the value) is assumed. If a formatting definition without # is present, the value is always added at the first place.

Example: `Format (#%, 1)` outputs values with one decimal place and a %-character after the value.

Additional Values				
Valuename	Valuetype	Parameter	Display-Text	Fixed Stat-Type Options
X ValuePercent	computed	\$PERCENT(\$Y1[TotalProd]	Total Prod. Value in %	IsColumn;Format

Level	Sum Total Prod. Value	Sum Total Prod. Value in %
Level 0	2.503.000,00	57,1%
Level 1	109.350,00	2,5%
Level 2	551.000,00	12,6%
Level 3	123.456,00	2,8%
Level 4	0,00	0,0%
Level 5	250.023,00	5,7%
Level 6	0,00	0,0%
Level 7	843.000,00	19,2%
Level 8	0,00	0,0%
Level 9	5.000,00	0,1%
	<b>4.384.829,00</b>	<b>100,0%</b>

**Note:** If a (numeric) value has no `Format` defined, it is displayed as floating point number with 2 decimal places. To output a value e.g. without decimal

places, you have to define it as additional value with `Format( #, 0)` in the **Options** field. Z-fields (e.g. 2nd currency fields) do not support number formatting.

---

In Aurea CRM.Web additional values can be selected as analysis categories and/or analysis value, depending on the following parameter(s) of the **Options** column:

- `IsColumn`: The additional value can be selected as a value.
- `IsCategory`: The additional value can be selected as a category.

You can also specify `true` or `false` for `IsColumn` and `IsCategory`, e.g. `IsColumn(false)`. Defaults: `IsColumn(true)` and `IsCategory(false)`. If you specify `IsCategory(true)`, `IsColumn` is automatically set to `false` unless explicitly defined otherwise.

---

**Note:** If you use additional values as categories, other options like **DoNotSortBars** have no effect. The sort order is always determined by the displayed text.

---

- `DefaultCategory`: The additional value is the default category.
- `NoChart`: The values for this column are displayed in the list but not in the chart view of the analysis.
- `Cumulate`: Displays accumulated values in the results column. The sum total row displays the highest accumulated value (instead of the sum total).
- `ComputeSum`: Calculates the sum total row for additional values (for all aggregation types) using the specified formula. (Without `ComputeSum` the formula is only applied to the result rows and the sum total row is calculated by the analysis engine.)

The following analysis functions can be used in combination with `ComputeSum`: `$SUM`, `$COUNT`, `$COUNTALL`, `$COUNTX`, `$STATIC`, `$SUMCAT`, `$SUMCATX`, `$SUMCATDISTINCT`, `$PERCENT`. For further details, see [Analysis Functions](#) on page 533.

If you specify `ComputeSum` for rolling date categories (Example in `UPDATE_DEFAULT: DateMonthlyRoll16`), each value is considered exactly once.

Additional values can also be used when working with array catalog fields: If an additional value with **Option** `IsCategory` returns an array of values, the record is counted for each of these category values (i.e. potentially the record is taken into account by multiple rows).

Example: "Sector" catalog of Company (in BTB): `[ $!FI[29], $!FI[30], $!FI[31] ]`

## To evaluate additional values once per row:

By default, additional values are evaluated per record. For values where all records in a category should deliver the same result (e.g. number of records or sum of a value across one category), you can change this behavior: Define a fixed aggregation by setting the **Fixed Stat-Type** to "Static" or specify `IsStatic` in **Options**.

Additional Values				
Valuename	Valuetype	Parameter	Display-Text	Fixed Stat-Type Options
✗ SumCompanies	computed	\$COUNT(\$FI)	Count Companies	isStatic
✗ SumEmployees	computed	\$SUMCAT(\$FI[37])	Sum Employees	Static

**Note:** Using **Fixed Stat-Type** "Static" disables the **Type** selection in Aurea CRM.Web (in the **Add a value** dialog).

**Note:** Additional categories (**XCategory**) are ignored when using static values.

## Examples

Some examples for your reference.

### Company analysis with "Revenue per Employee" as value (`IsColumn` option):

1. Define an additional value that divides **Revenue** (36) by **Employees** (37):

**Parameter** = `$FI[36]/$FI[37]`

Additional Values				
Valuename	Valuetype	Parameter	Display-Text	Fixed Stat-Type Options
✗ AsValue	computed	\$FI[36]/\$FI[37]	Revenue per Employee	isColumn

2. Define this value as default value.

Default Values	
Type	Column
↓ ✗ Sum	Revenue (Company)
↑ ✗ Sum	AsValue

Result in Aurea CRM.Web:

Category		Results (9 record(s))		
Category	Country	Country	Sum Revenue	Sum Revenue per Employee
Sum (Revenue)		Austria	360.908.791,00	2.606.330,25
Sum (Revenue per Employee)		German	5.800.000,00	1.657,14
		UK	999.999,00	12.499,99
		Albania	99.999,00	142,86
		Лд дава а оШжц дтдва тыва	20.000,00	66,67
		Andorra	4.985,00	6,34
		Ireland	0,00	0,00
		Switzerland	0,00	0,00
		USA	0,00	0,00
		<b>Add</b>	<b>367.833.774,00</b>	<b>2.620.703,25</b>

**Company analysis with ABC classification partitioned by the number of employees as category (IsCategory(true) option):**

1. Define an additional value for ABC by company size (splitting the number of employees in small, medium and large):

Additional Values				
Valuename	Valuetype	Parameter	Display-Text	Fixed Stat-Type Options
AsCategory	computed	$\$FI[37]<40?\$FI[35]+$ (sr) ABC per Company Size		IsCategory(true)

**Parameter** =  $\$FI[37]<40?\$FI[35]+$  (small:0-39) " :  $\$FI[37]<100?\$FI[35]+$  (middle:40-99) " :  $\$FI[35]+$  (large: >= 100) "

Result in Aurea CRM.Web:

Category "ABC per Company Size"		
Category	Results (6 record(s))	
Category ABC per Company ...		<b>Sum Revenue</b>
Values	A (large: >= 100)	60.000,00
Sum (Revenue)	A (middle:40-99)	999.999,00
	A (small:0-39)	0,00
	B (large: >= 100)	5.804.711,00
	B (middle:40-99)	200.000,00
	B (small:0-39)	8.515.866,00
		<b>15.580.576,00</b>

**Company analysis accumulating revenue data split by ABC classification (Cumulate option):**

1. Define the ABC field as (additional) XCategory. For further details, see [Using an Additional Category on the x-Axis](#).
2. Check **All Column on X-Categories**.
3. Define an additional value accumulating the revenue values:

Additional Values				
Valuename	Valuetype	Parameter	Display-Text	Fixed Stat-Type Options
CumValue	computed	$\$FI[36]$		Cumulate

4. Define this value as default value.

Default Values	
Type	Column
↓ X Sum	Revenue (Company)
↑ X Sum	CumValue

Result in Aurea CRM.Web:

Rep 1 ID	Sum Revenue				Sum CumValue			
	all	A	B	C	all	A	B	C
David	8.508.566,00	0	8.508.566,00	0	8.508.566,00	0	8.508.566...	0
Christoph Panther	999.999,00	999.999,00	0	0	9.508.565,00	999.999,00	8.508.566...	0
Imgard Sabat-Wainger	244.711,00	40.000,00	204.711,00	0	9.753.276,00	1.039.999,00	8.713.277...	0
Oliver Witvoet	200.000,00	0	0	200.000,...	9.953.276,00	1.039.999,00	8.713.277...	200.000,00
Administrator	20.000,00	20.000,00	0	0	9.973.276,00	1.059.999,00	8.713.277...	200.000,00
Simbrauf Putz	7.300,00	0	7.300,00	0	9.980.576,00	1.059.999,00	8.720.577...	200.000,00
Martin Putz	2.000,00	0	0	2.000,00	9.982.576,00	1.059.999,00	8.720.577...	202.000,00
	<b>9.982.576,00</b>	<b>1.059.999,00</b>	<b>8.720.577...</b>	<b>202.000...</b>	<b>9.982.576,00</b>	<b>1.059.999...</b>	<b>8.720.5...</b>	<b>202.000...</b>

## Explicit Analysis Categories

Use explicit analysis categories to aggregate data based on arbitrary criteria like number or date ranges or sets of catalog values.

Explicit analysis categories allow you to define "virtual catalogs" based on catalogs, dates and numeric data. These definitions can be hierarchical.

You can define/customize explicit analysis categories on the **Analyses** page. For further details, see [Defining Explicit Analysis Categories](#) on page 529.

**Explicit Analysis-Categories**

Name  Copy from Empty Category New

Name	Configuration
DateMonthly	UPDATE_DEFAULT
DateQuarterly	UPDATE_DEFAULT
Percentage_10Sliced	UPDATE_DEFAULT

You can define explicit analysis categories for the following field types:

- **Numeric fields:**

**Configuration for Analysis-Category 'ValueRange'**

**Analysis-Category ValueRange**

Label: Deal Size    
 display as other   
 Large

Other Items Mode:   
 Other Label:   
  Catalog-Browsing:

Value	Label	Operator	Value	Sub-Category
<input type="text" value="small"/>	<input type="text" value="Small"/>	<input "="" type="text" value="&lt;="/>	<input type="text" value="1000"/>	<input type="text"/>
<input type="text" value="medium"/>	<input type="text" value="Medium"/>	<input "="" type="text" value="&lt;="/>	<input type="text" value="10000"/>	<input type="text"/>

- **Date fields:** Use the date format YYYYMMDD. For more details. For further details, see [Analysis Date Categories](#) on page 532.

Configuration for Analysis-Category 'DateRange'

**Analysis-Category DateRange**

Label: (half-yearly)

Other Items Mode: display as other

Other Label:

Catalog-Browsing:

Value	Label	Operator	Value	Sub-Category
X YYYYH1	H1 {YEAR}	><	YYYY0101 - YYYY0630	
X YYYYH2	H2 {YEAR}	><	YYYY0701 - YYYY1231	

[New Value](#) [New Criteria](#) [New Criteria](#) [Save](#)

- **Catalog fields:** Check **Catalog-Browsing** and select the desired info area and catalog field. Enter a catalog value from a synchronized catalog either by clicking on the button or by entering the catalog code in the text field. For further details, see [Catalogs](#) on page 720.

**Note:** Multiple criteria are combined with OR.

Configuration for Analysis-Category 'Regions'

**Analysis-Category Regions**

Label: **Regions**

Other Items Mode: display as other

Other Label:

Catalog-Browsing: Country (LA) Country (0)

Value	Label	Operator	Value	Sub-Category
X DACH		=	Deutschland [0] CatCode = 2	
		=	Österreich [0] CatCode = 1	
		=	Schweiz [0] CatCode = 3	
X USUK		=	4 <input type="text"/>	
		=	USA (1_2_4) <input type="text"/>	
		=	--- (Cat #2) <input type="text"/>	

[New Value](#) [New Criteria](#) [Save](#)

- **Rep fields:** You can use `$curRep` for rep ID fields (not rep Z-fields).

**Examples:**

Deal Size: category based on a numeric value.



Small	Value < 1000
Medium	Value < 10000
Large	other

Regions: category based on the country catalog.

Benelux	NL, BEL, LUX
DACH	D, A, CH
CEE	CZ, SLO, H, PL, ...
other	other

## Defining Explicit Analysis Categories

Here you can learn how to define Explicit Analysis Categories.

To define an explicit analysis category:

1. Select an explicit analysis category or create a new one. For further details, see [Designing Configuration Units](#) on page 64.

Configuration for Analysis-Category 'DateRange'

**Analysis-Category DateRange**

Label: (half-yearly) display as other

Other Items Mode:

Other Label:

Catalog-Browsing:

Value	Label	Operator	Value	Sub-Category
X YYYYH1	H1 {YEAR}	><	YYYY0101 - YYYY0630	
X YYYYH2	H2 {YEAR}	><	YYYY0701 - YYYY1231	

[New Value](#) [New Criteria](#) [New Criteria](#) [Save](#)

2. Specify the following:

- **Label:** The multilingual name of the category. This name is displayed in Aurea CRM.Web (in the **Category** drop-down list and in **Drill down/up** menus) instead

of the underlying field name. The texts are saved in the `statcategory_texts` text group. For further details, see [Texts](#) on page 176.

- **Other Label:** Values that do not match the defined criteria are automatically aggregated in an "Other" column. You can define a label for this column here.
- **Catalog-Browsing:** Check to define explicit analysis categories based on catalog values.

<input checked="" type="checkbox"/> Catalog-Browsing:	Country (LA) ▼	Country (0) ▼
---	----------------	---------------

Select the desired info area and catalog field.

---

**Note:** As explicit analysis categories are not bound to a specific field, CRM.Designer does not automatically know for which field this category is defined and whether this field value is based on a catalog. With this option, you can manually select a catalog-bound field if you want to enter catalog-synchronized values.

---



---

**Note:** As soon as at least one catalog-synchronized value is stored in the category, CRM.Designer automatically displays one field that references this catalog. Even if it is not the field you selected, it is the same catalog!

---

### 3. Click + (**New Value**) to add the desired values.

Each value consists of:

- **Value:** Uniquely identifies the value.
- **Label:** The multilingual name of the value. The texts are saved in the `statcategory_texts` text group. This name is displayed as column header in Aurea CRM.Web. If you do not specify a **Label**, the value is omitted in analysis results based on this category.
- **Sub-Category:** Offered for drill-downs if this explicit analysis category is the current category, or the value for this virtual category is fixed due to a previous drill-down.

Each value of the underlying category field belongs to one defined category value or the "other" category. The category values are checked from top to bottom. A value belongs to the second category value if it matches the criteria for that value and it does not belong to the first category value.

### 4. Click + (**New Criteria**) to specify the range of values of the underlying category field for the value you are defining. Multiple criteria are combined with OR.

You can use variable expressions for dates (`$curDay`, `$curQuarter+1q` etc.), see [Using Variable Expressions](#). Use e.g. `YYYY` as a placeholder. (You cannot use wildcard expressions with `#####` here. For more information on defining date categories. For further details, see [Analysis Date Categories](#) on page 532.

5. Click **Save**.

## Using Explicit Analysis Categories

Here you can learn how to use Explicit Analysis Categories.

To apply an explicit analysis category to a field:

- Enter the name of the explicit analysis category in the **Category** column (e.g. the `DateRange` category from the above example): Users can select the category in the **Category** drop-down list in Aurea CRM.Web. Sub-categories and underlying catalogs can be used for drill-downs.

Field Name	CriteriaField	AdditionalCriteria	ReadAll	DependentOnCurrency	WeightedField	DoNotSortBars	ShowAllBars	NoOther	MustSelect	AlternateCurrencyField	CategoryNotMandatory	XCategory	Defaultvalue	Default-End	Slices	Category	List-Nr	
Company (1)																		
Activity (1)																		
Activity Type (0)	<input checked="" type="checkbox"/>																	
Date (2)	<input checked="" type="checkbox"/>											<input checked="" type="checkbox"/>	\$curDay-18m				DateRange	
Time (3)																		
Duration (4)	<input checked="" type="checkbox"/>																	
Unit (5)																		
Person (77)																		
Subject (103)																		

no Infoarea selected | 1

CriteriaDefault: Activity Type (Activity)

- If you check **CriteriaField** in addition, the explicit analysis category can be used for drill-down.
- If you check **XCategory** in addition, you can use the explicit analysis category as additional category to be displayed on the x-axis, see [Using an Additional Category on the x-Axis](#).
- If you set the explicit category as the default category in **CriteriaDefault**, users can only select the category specified in **Category**. To define an explicit category as the default category and still allow users to choose another category, additionally check **CategoryNotMandatory**.

## Analysis Date Categories

Learn to aggregate results into defined time spans.

In analyses that use a date field as root category users can not only analyze data by the actual date value of a field but also aggregate the result into defined time spans. Example - Date(monthly).

---

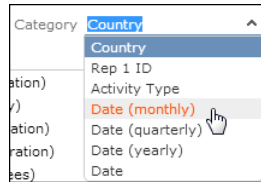
**Note:** A drill-down inside a date category (e.g. show quarters if this year) is not possible.

---

UPDATE\_DEFAULT contains the following explicit analysis categories for date segmentation:

DateMonthly, DateQuarterly, DateYearly.

The Web Configuration parameter `Analysis.DateCategories` defines which date segmentation options are available for users. For further details, see [Analysis.Date-Categories](#).



You can add custom date categories if required. For further details, see [Defining Explicit Analysis Categories](#) on page 529.

Configuration for Analysis-Category 'DateRange'

**Analysis- Category DateRange**

Label: (half-yearly)

Other Items Mode: display as other

Other Label:

Catalog-Browsing:

Value	Label	Operator	Value	Sub-Category
X YYYYH1	H1 {YEAR}	><	YYYY0101 - YYYY0630	
X YYYYH2	H2 {YEAR}	><	YYYY0701 - YYYY1231	

[New Value](#) [New Criteria](#) [New Criteria](#)

**Save**

Specify dates using the format YYYYMMDD.

Use YYYY as placeholder for comparison values.

Use {YEAR} as placeholder for the year in the label text.

## Analysis Functions

Learn about analysis function.

When defining additional values you can use special aggregation functions to calculate the value, see [Additional Values](#) on page 522.

Enter the formula to calculate the value in the **Parameter** field:

Additional Values				
Valuename	Valuetype	Parameter	Display-Text	Fixed Stat-Type Options
X ValuePercent	computed	\$PERCENT(\$Y1[TotalProd\	Total Prod. Value in %	IsColumn;Format

Syntax for referencing field values: \$<infoAreaID>[<field name or number>]

The field must be part of the underlying query.

Use \$Y1 [0] [FreeN1] and \$Y1 [1] [FreeN1] for multiple occurrences of a field.

UPDATE\_DEFAULT contains the following analysis functions:

- \$BETWEEN([cmpval, minval, maxval, trueval, falseval]): Returns records within a certain range. A cmpval between minval and maxval (including the bounds) returns trueval, everything else returns falseval.

### Examples:

Sales opportunities within the first half-year of 2012:

```
$BETWEEN([u8.services.typeConverter.dateToString(($!Y1[CompleDate]),
{format: "YYYYMMDD"}), "20120101", "20120630", $Y1[TotalProdValue], 0])
```

For more information on the dateToString function, see the article *"The Type Converter Service"* at <https://support.aurea.com>.

>Contact records with costs between 1000 and 1500: \$BETWEEN([ \$MA[7], "1000", "1500", \$MA[59], 0])

---

**Note:** Note: If a query calculates average values (e. g. average costs), the costs are divided by the total costs of all records in the category.

---

- `$COUNT(value)` : Returns the sum of different record for a record (e.g. number of opportunities per company).
- `$COUNTX(value)` : returns the number of records per XCategory and Category.
- `$CURRCONV(value)` : Returns the value corrected by a conversion factor (if a currency column is specified).
- `$MAX(param)` : Returns the maximum value of all values passed in `param`. `param` is a JavaScript array of `n` values with the syntax `[value1,value2,...]`.

**Example:** `$MAX([$Y1[TotalProdValue], $Y1[WtdTotalProdValue], $Y1[OPWtdTotalProdValue]])`

- `$MIN(param)` : Returns the minimum value of all values passed in `param` (same syntax as `$MAX`).
- `$PERCENT(value)` : Returns the percentage of the value in the current row compared to the summed value over all rows. `$PERCENT(value)` is short for `value*100/$SUM(value)`.
- `$SUM(value)` : Returns the sum of the value over all rows independently of the category.
- `$SUMCAT(value)` : Returns the sum of a value within a category.

**Example:** Analysis of AK with MA with Y1 calculating the average marketing activity cost per opportunity for each category: `Average $SUMCAT($AK[37])/Math.pow($COUNT($Y1), 2)`

- `$SUMCATDISTINCT(param1, param2)` :

Where `param1` is a JavaScript array of `N` values with the syntax `[value1,value2,...,valueN]` and `param2` is a category.

The function returns the sum of distinct values in an array for a category. It works the same way as `$SUMCAT` if the values are numeric and returns the sum of a value within a category.

If the argument is an array and the array has only one element, then the value is added as value for key 'const', and indicates a static value.

If the argument is an Array with more than one element, the function tries to add values for distinct keys or categories defined in `param 2`.

**Example 1:** `$SUMCATDISTINCT([$FI[5], $FI])` returns the sum of all unique FI records for field #5, and if a `n:1` linked table like MA is used in the same formula with multiple MA records pointing to the same FI, then each FI is summed just once.

**Example 2:** If the search result set were to have a list of 3 records containing the item name and number of items in an order: {25, item1}, {30, item2}, {40, item1}. Then applying the function on the array returns a result set containing {65, item1}, {30, item2}.

- `$SUMCATX(value)` : returns the sum per XCategory and Category.
- `$WEIGHTCONV(value)` : Returns the value corrected by a weight factor (if a weight column is specified).

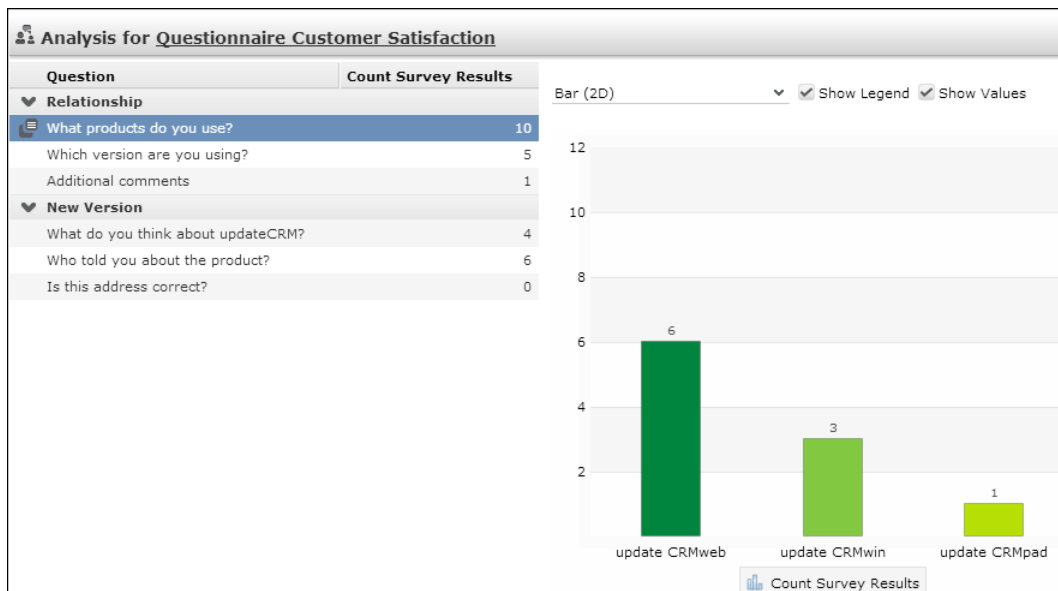
You can add custom analysis functions on the **Analyses** page:

Analysis Functions	
Name	<input type="text"/> <input type="button" value="New"/>
Name	Configuration
BETWEEN	UPDATE_DEFAULT
COUNTALL	UPDATE_DEFAULT
CURRCONV	UPDATE_DEFAULT
MIN	UPDATE_DEFAULT
STATIC	UPDATE_DEFAULT
SUMCAT	UPDATE_DEFAULT
SUMCATX	UPDATE_DEFAULT

Analysis functions are defined by a name and its JavaScript class name.

## Questionnaire Analysis

UPDATE\_DEFAULT contains the configuration of a survey results analysis (answers per question).



The left part of the page lists all questions of the selected questionnaire. Question records with **New Section** checked are displayed as group headers. Question records with **Do not display** (F2:23) checked are not listed.

**Note:** You may need to adjust the value of the `Analysis.MaxRecords Web Configuration` parameter to ensure that all results are displayed, see [Analysis.MaxRecords](#).

Clicking on a question displays the analysis chart of the answers. For textual answers (**Edit Answer** checked) a list of answers is displayed instead of the analysis chart. This list is configured by the List control of field group U2.Analysis.

For information on defining questionnaires and conducting surveys, see **Surveys** in the *Web Business Logic User Guide*.

**Questionnaire Analysis** consists of the following configuration units:

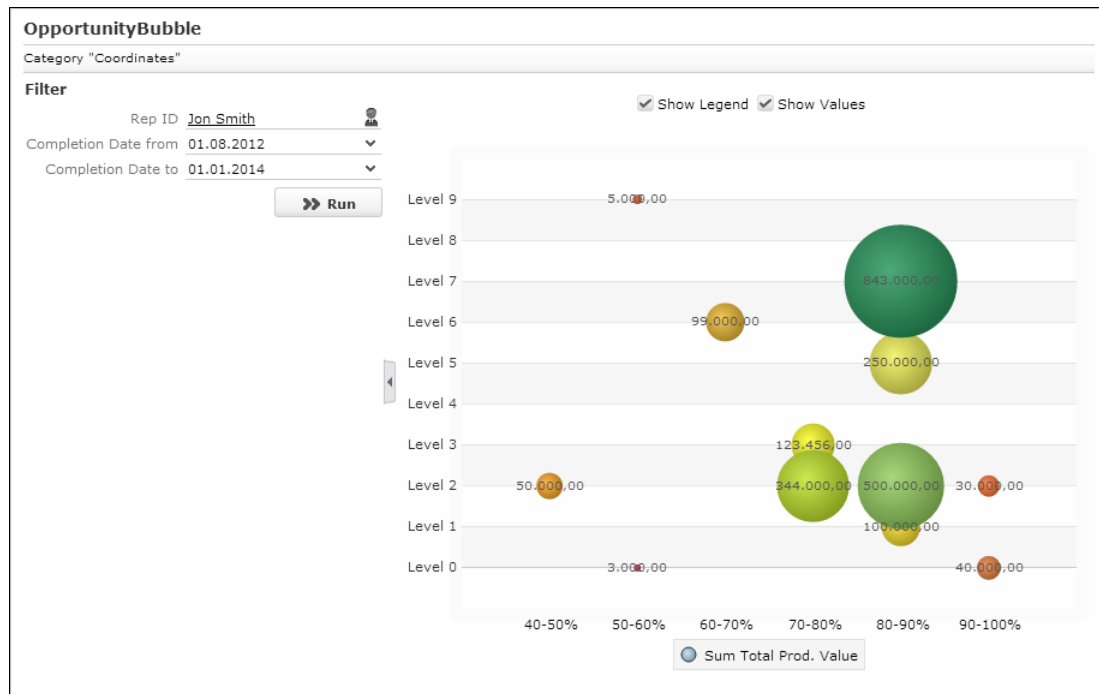
- The **Questionnaire Analysis** is called by menu item A\_QuestionnaireAnalysis in the M\_QuestionnaireView context menu (**Collaboration > Questionnaire > Questionnaire Analysis**). This calls a ShowForm action displaying the QuestionnaireAnalysis form.
- The QuestionnaireAnalysis form consists of a header and a QuestionnaireAnalysis widget containing the configuration of the list of questions and the analysis chart.

You can configure/change defaults for the following chart properties via the Form Designer (answerAnalysis parameters for widget QuestionnaireAnalysis): chartType, fixedChartType, hideSelector, showLegend, showValues, showEditor. The fieldGroupName parameter allows you to specify another field group for the textual answers list.

- The QuestionnaireAnalysis widget uses the following queries:
  - Questionnaire\_SurveyResults providing the data for the list.
  - Questionnaire\_SurveyAnswers providing data for the analysis chart.

The QuestionnaireAnalysis widget uses the following analyses:

### Bubble Chart Analyses





You can configure analyses to use Fusion Bubble charts as chart type. This chart type is not visible/selectable in Aurea CRM.Web. Display-Order = -1). For further details, see [Chart Types](#) on page 545.

UPDATE\_DEFAULT contains two bubble chart analyses `OpportunityBubble` and `TicketBubble` which can be displayed via the application menu entries `ID_NET_OpportunityBubble` and `ID_NET_TicketBubble`.

Each bubble is defined by an X,Y coordinate pair and the bubble size (z-axis). These values must be provided by the analysis definition, see [Defining Analyses](#) on page 515.

Example: `OpportunityBubble`

Default Values			
Type	Column		
✖ Sum	Total Prod. Value (Opportunity)		
⊕ Sum	Coordinates		
Additional Values			
Valuename	Valuetype	Parameter	Display-Text Fixed Stat-Type Options
✖ Coordinates	computed	(Math.min(90,Math.floor(\$	IsCategory;Defau

- The bubble size is calculated based on the 1st analysis value defined under **Default Values**, in this example the **Total Prod. Value**.
- The X,Y coordinates pair must be defined as Additional Value of type "computed", >> [Additional Values](#) on page 522. Separate the values for X and Y by semicolon.

```
(Math.min(90,Math.floor($Y1[ProbabilityP]/10)*10))+';'+$Y1[Level]
```

In this example, X is calculated based on the **Realization Probability** (in steps of 10 via the explicit analysis category `Percentage_10Sliced`), Y based on the **Level**.

## Timelines

Use timeline to visualize customer history in a graphical representation.

The screenshot shows a CRM timeline for 'Company TimeLine Inc.' covering the period from 23.04.2013 to 02.05.2013. The timeline is a horizontal axis with various activity bubbles. A bubble for '25.04 - 26.04.2013' is highlighted in blue, containing '2 Activities' and '1 Opportunity'. Below the timeline, a detailed view of a 'Telephone' activity is shown for '25.04.2013 15:07: Telephone for TimeLine Inc.'. The detailed view includes fields for Subject, Date, Time, Activity Type, Duration, Text, Rep ID, and Participant ID.

A timeline is a tool for visualizing a customer history. For example - a timeline can display marketing activities, visits and offers linked to a company or person. It consists of a graphical representation of several records from different info areas linked to a parent record.

The Timeline view is called using the `Timeline` action template. For further details, see [Timeline](#) on page 475. Examples in `UPDATE_DEFAULT: A_AccountTimeline` in `M_Company` and `M_Person`, `A_MarketingTimeline` in `M_Person`.

The `Timeline` widget is also available in the Forms Designer. For further details, see [Forms](#) on page 621.

The timeline requires a valid record `uid` used as parent when querying for the records to be shown on the timeline. The timeline can display records from all info areas that contain at least one field of type "date".

A timeline page consists of the actual timeline with the child records displayed as "bubbles", a `RecordListBox` listing all records contained in the selected bubble and an Expand view of the record selected from the list. For further details, see [Configuration Units](#) on page 539. If the selected bubble contains only one record, an Expand view of this record is displayed instead of a list.

The type of child records and their graphic representation (icons, borders) are determined via the Timeline configuration (referenced in the action). For further details, see [Configuring a Timeline](#) on page 540.

---

Use the **SDK** page to test your timelines: **SDK > Timeline**.

## Configuration Units

This topic explains about Configuration Units in Aurea.CRM web.

A timeline consists of the following configuration units:

- a page header is a special header `Timeline` in `UPDATE_DEFAULT`, containing the zoom options and the `TimelineAddInsight` button allowing users to add notes. These notes are saved as **Note** records (NO) linked to the timeline's parent. Example - FI. (Make sure that the parent record of your timeline is linked to the NO info area.)
- a Timeline configuration. For further details, see [Configuring a Timeline](#) on page 540
- Expand configurations for the info areas contained in the Timeline configuration used to display selected records in the Related Data area.
- table captions for the records displayed in the bubbles. If the table caption contains an image, this is displayed as well.
- table captions for the records displayed in the list

The names of these configuration units are derived from the name of the Timeline configuration. For further details, see [Naming Conventions](#) on page 539.

## Naming Conventions

This topic explains about Naming Conventions in Aurea.CRM web.

Configuration units for timelines need to follow special naming conventions using the name of the Timeline configuration which is referenced by the action parameter `name`.

The following table lists default names and action parameters (if available) for all configuration units:

Configuration unit	Name	Examples
Header	Header group / Header name: <InfoAreaId> / <Name>Timeline Fallbacks: <InfoAreaId> / Timeline Special / <Name>Timeline Special / Timeline	FI AccountTimeline KP AccountTimeline KP Timeline AccountTimeline Timeline
Timelineconfiguration	<Name>Timeline	AccountTimeline MarketingTimeline
Expandconfiguration	<InfoAreaId>.<Name>Timeline	MA.AccountTimeline BR.MarketingTimeline
Table caption (bubble)	<InfoAreaId>.<Name>Timeline Fallbacks: <InfoAreaId>.Timeline <InfoAreaId>	MA.AccountTimeline Y1.Timeline
Table caption (list)	<InfoAreaId>.<Name>Timeline.Related Fallbacks: <InfoAreaId>.Timeline.Related <InfoAreaId>	MA.AccountTimeline.Related Y1.Timeline.Related

## Configuring a Timeline

This topic explains how to configure the timeline in Aurea.CRM web.

The Timeline configuration defines the content of the timeline, info areas of child records, filters, bubble style, icons.

On the CRM.Designer main page in the **Filter | Query | Analysis** area click



**Timelines** to display all available Timeline configurations and create new ones. For further details, see [Designing Configuration Units](#) on page 64.

Timeline 'AccountTimeline'

Save Delete

Description Account Timeline

Info Area	Link ID	Date Field	Filter	Criterion Field	Image	Css/Border Color
Activity (MA)	<input type="checkbox"/>	Date (2)	MA.Contacts...	Activity Typ... global	CAT_CONTACT_BT...	
Offer (PR)	<input type="checkbox"/>	Date (6)		Status (8) global	CAT_ZF_1	open
				Status (8) = 2	CAT_ZF_4	closed
				Status (8) = 3	CAT_ZF_1	open
				Status (8) = 4		
Opportunity ...	<input type="checkbox"/>	Completion Date (...)				

To configure a timeline:

1. Select an info area from the drop-down list and click + (**Add**). The timeline can display records from all info areas that contain at least one field of type "date".
2. For each info area you can define the following:
  - **Link ID:** Check **Link ID** to specify another than the default link. For further details, see [Link IDs](#) on page 19.
  - **Date Field** (mandatory): Select the date field that is used to calculate the record's position on the timeline. Only date fields from the selected info area are available.
  - **Filter:** Select a filter to filter timeline items, e.g. `MA.Contacts_Done`. Only filters defined for the selected info area are available.
  - **Criterion Field:** Define criteria for displaying images in the bubble and/or applying a specific border color depending on a field value, e.g. activity type images for activity bubbles, red border for opportunity bubbles if Status = lost etc.

The input method and available operators depend on the field type and the selected operator:

- **Catalog fields:** If the catalog is synchronized (see [Catalog Value Synchronization](#)), you can toggle between text input (📝) and catalog input (📄), otherwise you need to enter the value's code as text. In this example MA>Type has been synchronized, PR:Status has not:

Info Area	Link ID	Date Field	Filter	Criterion Field
Activity (MA)		Starts on (1002)	MA.Contacts...	Type (1000) = Visit
				Type (1000) = 3
				+ Status (8) = 3
Offer (PR)		Date (6)		Status (8) = 2

- Number fields: If you select a number field, you can use the "range" operator:

Budg.curr.Y...	range	from		to
----------------	-------	------	--	----


You can use variable expressions like `$curDay` etc. For further details, see [Using Variable Expressions](#) on page 505.

- **Image:** Specify an image to be displayed in the bubble (in addition to an info area image configured in the table caption). Depending on the selected operator you can
  - set an image for a specific field value, e.g. if **Status** = 2 show `CAT_ZF_1`.
  - If you select "global" as operator you can use the `Small Image Template` syntax: `<image_name>_{VALUE}`  
 Example: Defining `CAT_CONTACT_BTBTB_{Contact}` for the **Type** field in MA displays the activity type image in the bubble.
- **Css/Border Color:** Specify the bubble border color using the hexadecimal color code e.g. (`#ff0000`) or by providing a CSS class (e.g. `selected`, `open` etc.) or `.`  
 For a list of available predefined bubble styles (CSS classes), see the article "How to Configure the Timeline" at <https://support.aurea.com>.

---

**Note:** For a selected criterion field and its comparison value you can define both an image and a border color (in one row).

---

To reorder items, click  (**Drag&Drop**) in the respective row and drag it to the desired position.

---

**Note:** Both info areas and criterion fields can be added more than once.

---

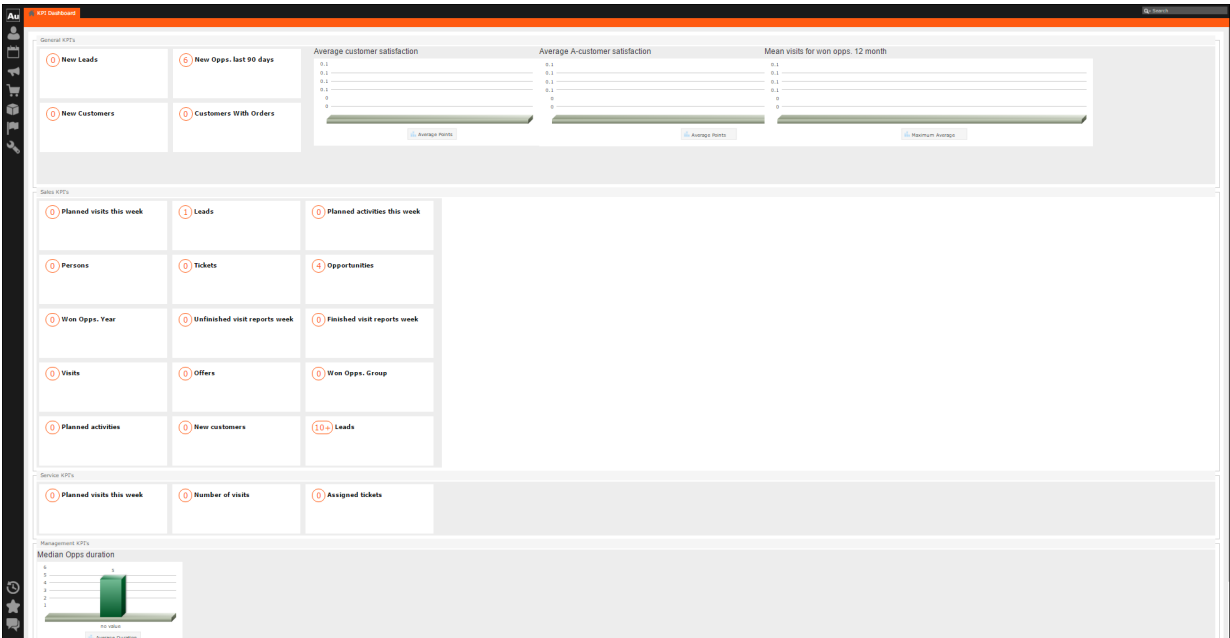
To make the timeline available in Aurea CRM.Web configure a `Timeline` action referencing the Timeline configuration. For further details, see [Timeline](#) on page 475.

# Key Performance Indicators (Aurea Enterprise)

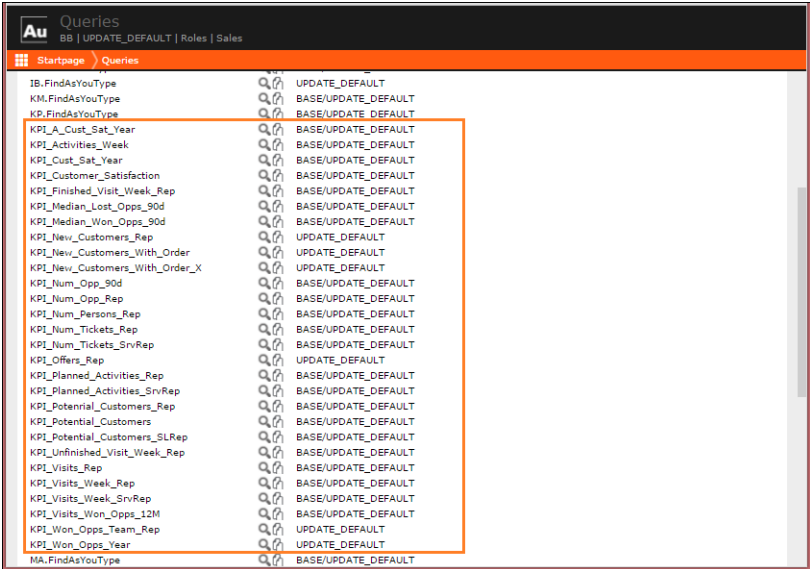
An overview of the Key Performance Indicators dashboard.

The KPI dashboard provides a quick measure of your business metrics primarily focusing on customer development and management. It shows the current state of your sales for existing and new customers. This would be very useful in monitoring open calls, new customers of interest and other related information without having to be informed by email.

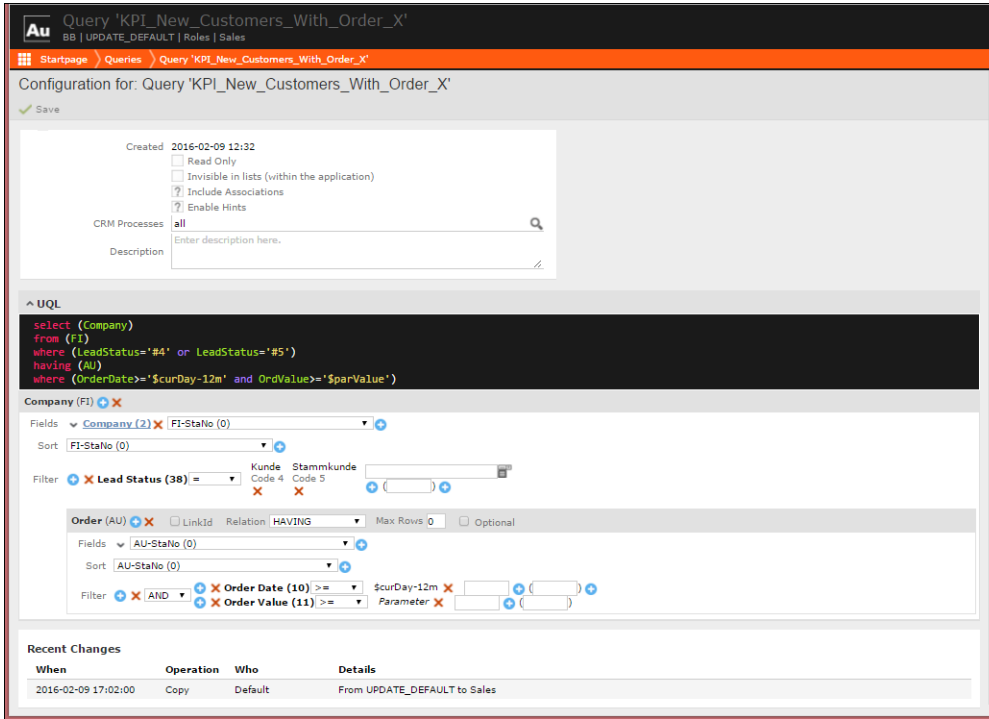
The following screenshot shows the KPI dashboard.



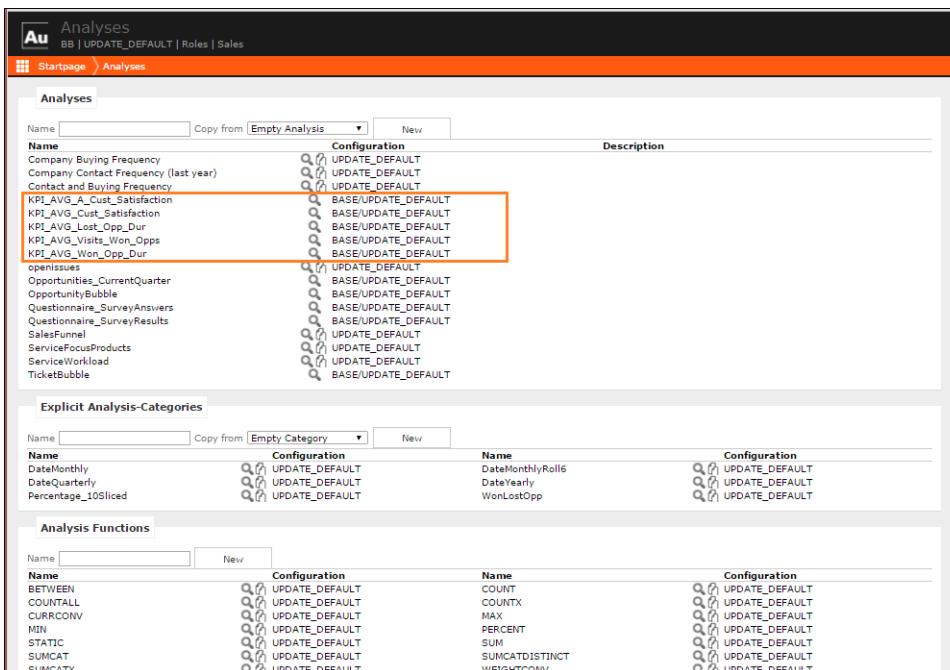
The queries that correspond to the KPI metrics displayed to the CRM users are defined in the CRM.Designer as shown in the screenshot below:



You can configure the KPI Dashboard queries according to the requirements of your users and customers. For example, you may have to set the values of some parameters that can vary for different users. The following screenshot shows the order size value that can be set as required for the **Customers With Orders** metric displayed on the dashboard.



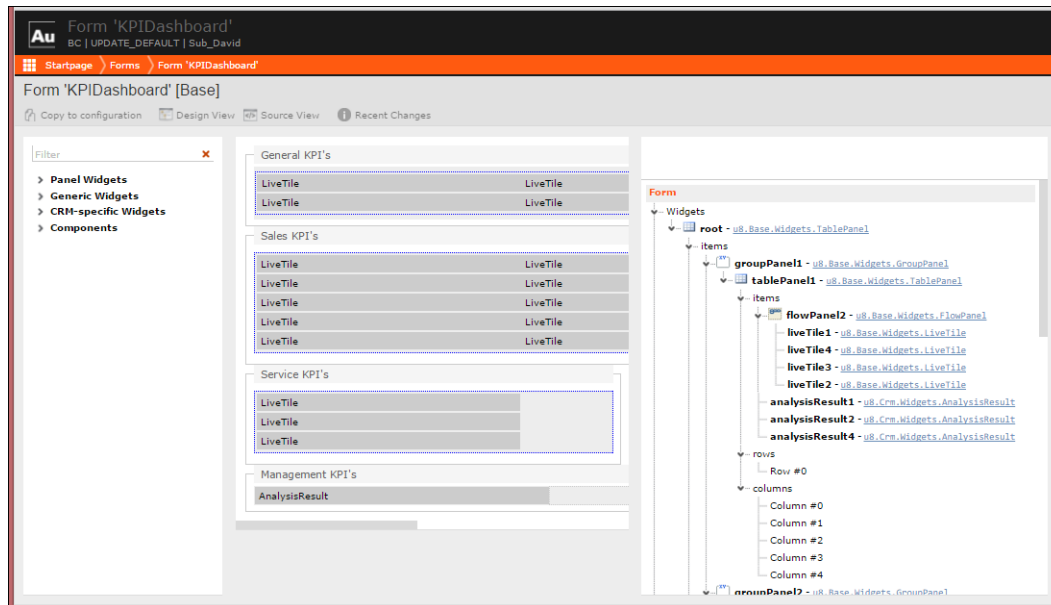
The following screenshot shows the KPI related Analyses available for use out of the box.





To develop a new Analysis for the KPI dashboard using the existing KPI Analyses as templates, perform the following:

1. Create a copy of the required KPI Analyses you want use as a suitable template.
2. Edit and change the Analyses according to the new requirements.
3. Save it with a new name.
4. Apply the new Analyses into the KPIDashboard form. The following screenshot shows the KPIDashboard form. For more information on working with forms, see [Forms](#) on page 621.



**Note:** All items in the KPIDashboard form are by default linked to the proper CRM processes, but they can be configured and adapted.

## Chart Types

Define a new chart type suitable for every purpose.

Aurea CRM.Web uses FusionCharts for the graphical display of analysis results. For further details, see [AnalysisRun](#) on page 395.

On the CRM.Designer main page in the **Filter | Query | Analysis** area click

Chart Types

**Chart Types** to display all available chart types.

Name	Type	Position in List	Configuration
AreaStacked2D	FusionChart	9	UPDATE_DEFAULT
Bar2D	FusionChart	1	UPDATE_DEFAULT
Bar3D	FusionChart	2	UPDATE_DEFAULT
BarHorizontal2D	FusionChart	3	UPDATE_DEFAULT
BarHorizontal3D	FusionChart	4	UPDATE_DEFAULT
BarStacked2D	FusionChart	5	UPDATE_DEFAULT
BarStacked3D	FusionChart	6	UPDATE_DEFAULT
Bubble	FusionChart	not selectable	UPDATE_DEFAULT
Combination3D	FusionChart	25	UPDATE_DEFAULT
Funnel	FusionChart	not selectable	UPDATE_DEFAULT
Line2D	FusionChart	9	UPDATE_DEFAULT
Line3D	FusionChart	not selectable	UPDATE_DEFAULT
LineZoom	FusionChart	10	UPDATE_DEFAULT
Pie2D	FusionChart	7	UPDATE_DEFAULT
Pie3D	FusionChart	8	UPDATE_DEFAULT
Radar	FusionChart	12	UPDATE_DEFAULT

You can edit existing chart types or create new ones. For further details, see [Designing Configuration Units](#) on page 64.

Label **Bar (2D)**

Renderer **FusionChart**

RendererOptions `crm/fusionCharts_template.json.js`

Display-Order `1`

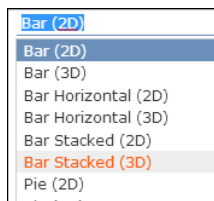
Options 

```
{
  url: "MSColumn2D.swf"
}
```

These options will extend or override the renderer options!

The following settings are available:

- **Label:** The text in the chart type drop-down list.



The texts are saved in the `net_charttype_text` Text Group. For further details, see [Texts](#) on page 176.

- **Renderer:** The charting engine used: "FusionChart".
- **RendererOptions:** The default settings for all chart types are saved in the `fusionCharts_template.json.js` file located in the `..\web\data` directory. Enter another file name to provide your own settings file.

Alternatively you can use the Options field to specify renderer options (see below).

Information on all available settings and renderer options. For further details, see <http://docs.fusioncharts.com/charts/> (section "Chart XML API").

- **Display-Order:** Determines the order in which the chart types are displayed in the chart type drop-down list. Set the Display-Order to "-1" to hide a chart type from the drop-down list. Charts with Display-Order =-1 are labeled as "not

selectable" on the **Chart Types** page (e.g. Funnel, Bubble). Such chart types cannot be selected by users in Aurea CRM.Web.

- **Options:** Displays the file used for the selected chart type. Here you can enter additional renderer options or change existing ones thus overruling the definition in the file specified in **RendererOptions**.

Detailed information on defining chart types, see the article “How to Define a New Fusion Chart Type” at <https://support.aurea.com>.

Use the **SDK** page to further explore the charts functionality: **SDK > Charts**.

## Processes

Use processes to guide users through complex task in Aurea CRM.Web.

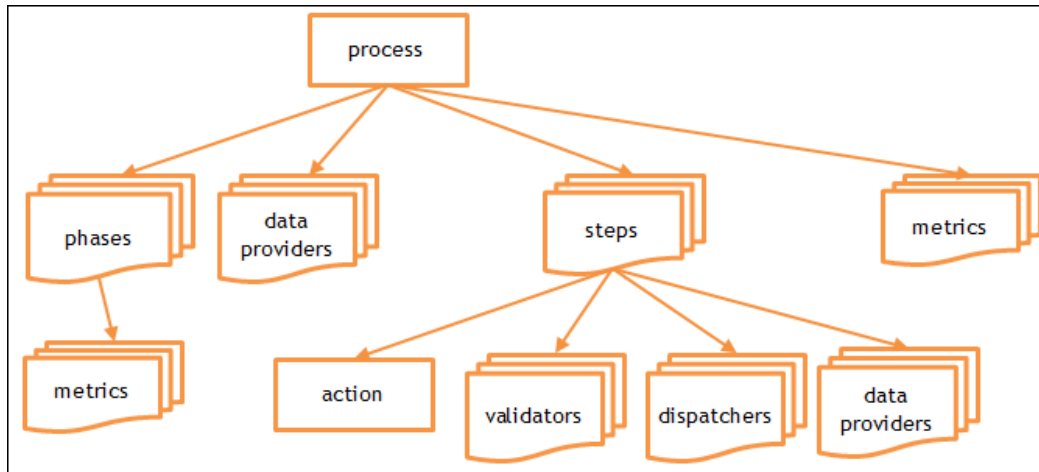
Processes are defined as:

- a simple task completed within one user session, like creating a new record with selecting a parent record first (example in UPDATE\_DEFAULT: `NewAppointment`).
- a complex task spanning many user sessions, e.g. the opportunity process which starts with identifying an opportunity and ends weeks or months later with creating an offer (example in UPDATE\_DEFAULT: `MergeRecords`).

You can either use and customize the processes and steps defined in UPDATE\_DEFAULT or create your own custom processes. For further details, see [Process Elements](#) on page 548 and [Designing a Process](#) on page 572.

## Process Elements

Learn about the process elements.



A process consists of the following elements:

- The **process** itself containing at least the process's name and (if the process does not contain phases) an initial step. For further details, see [Process](#) on page 550.
- To make the process available in Aurea CRM.Web you need to define an action (e.g. a menu action or a button) **calling the process** via the `Process` action template. For further details, see [Calling a Process](#) on page 575.
- You can divide complex processes into process **phases** allowing the user to exit the process and resume it at a later time. For further details, see [Phase](#) on page 551.

A process can consist of phases or just some steps. A phase always consists of one or more steps.

- Each process or phase consists of one or more process **steps** containing the action to be executed, e.g. `ChooseInfoAreaForContact->SelectRecord` being the **initial step** of the `NewAppointment` process. For further details, see [Step](#) on page 555.
- You can create **sub-steps** that execute a number of steps themselves, e.g. the `SelectPerson` step called by the `ExecuteSelectRecordSubStep` action allowing the user to search for or create a person record in an existing or new company. For further details, see [ExecuteSelectRecordSubStep](#) on page 581.

---

**Note:** Steps and sub-steps can be (re-)used in multiple processes.

---

- Each step calls an **action** (e.g. a `ProcessAsk`, `Search` or `Expand` action). For further details, see [Action](#) on page 559.

---

Any supported action can be used (**Callable from** setting in the action template).

- For each step you need to define **dispatching** rules, i.e. conditions or events leading to the **next step**, e.g. button A calls step X, button B calls step Y. For further details, see [Dispatching](#) on page 560.

A step automatically dispatches to the next step, if the defined conditions are met.

You can allow the user to cancel the process at any step and roll back to state before the start of the process. For further details, see [Cancelling a Process](#) on page 575.

- You can define **data providers** containing values from process-internal or external sources, e.g. a parent record for a newly created record. These data providers can be accessed at various points during the execution of a process (e.g. process storage, output message etc.). For further details, see [Data Providers](#) on page 562.
- You can define **validation** criteria that must be met before the next step can be executed, e.g. at least one record must be selected for the `MergeRecords` process to start (as defined in the process's initial step `PrepareForeMerge->DisplayMergeOverview`). For further details, see [Validation](#) on page 571.
- For each process element you can add
  - a **Description** for internal use in CRM.Designer. For further details, see [Adding Descriptions](#) on page 66
  - a **Help Text** displayed as a tool-tip providing user assistance during the process's execution. The texts are saved in the `ProcessHelp` Text Group. For further details, see [Texts](#) on page 176. You can use the Markdown syntax to format your help text, see <http://en.wikipedia.org/wiki/Markdown>.

---

**Note:** To get correct line breaks in your help texts, make sure you use the following syntax:

- - for closing a paragraph, enter two newlines, i.e. press return,
  - - to create a line break, enter at least two spaces and press return.
- 

- For processes and phases you can define **metrics** to graphically (or textually) display the progress of the process or phase. For further details, see [Metrics](#) on page 571.

The sample processes described at the end of this section contain examples of most of the process elements. For further details, see [Examples](#) on page 591.

## Process

The process is the starting point for a series of steps that need to be executed to complete a task.

**Process 'NewOpportunityForCompany'**

Save Delete Show Usage Graph Steps...

Process: NewOpportunityForCompany

CRM Process: all

Header: ProcessHeader\_Y1

Description: Creates an opportunity.

Help Text: Create an opportunity and/or add opportunity participants.

Initial Step: (None)

**Process Phases**

CreateY1 AddY2 +

**Data Providers**

Name	Action
Input Message	
Process Storage	
curTexts	UsingDataProvider

+ New Name

**Process Storage Variables**

Name	Data Provider	Value
curFIUID	none	0
curY1UID	Input Message	\$.get('uid')
curRepName	none	u8.session.identity.repName
curY2Count	none	0
repMetric	curTexts	\$(('TXT_Process_TextGroup_0'))

+ New Name

**Process Metrics**

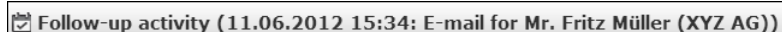
View	Data Provider	Value	Help Text
Text	Process Storage	\$.get('repMetric')+\$.get(...	The name of the curr...

+ New Name

A process's configuration consists of the following settings:

- **Process:** The name of the process. Used as input argument `name` in the `Process` action template. For further details, see [Calling a Process](#) on page 575.
- **CRM Process:** Assign the process to a CRM process. For further details, see [Assigning CRM Processes](#).
- **Header:** An optional header for the process. Define an info area-independent special header. For further details, see [Special Headers](#) on page 342. Supported placeholders:
  - {INFOAREAIMAGE}
  - {RECORD}
  - {RECORDLINK}
  - {INFO}
  - {INFOS}

**Example:** If you define `{INFOAREAIMAGE} Follow-up activity ({RECORD})` as header text for the special header `NewFollowUpAppointment`, the header of the `NewFollowUpAppointment` process is displayed as follows:



- **Description** (optional): Add an internal description (only visible in CRM.Designer).
- **Help Text** (optional): Add a description of what the process is supposed to do. The **Help Text** is displayed as a tool-tip when positioning the mouse over the header text or clicking on the header.
- **Initial Step:** The process step to be called when the process is launched.  
If a process contains phases, the initial step from the active phase is used. If no active phase is found, the initial step from the process is used as a fallback.
- **Process Phases** (optional): The phases of which the process consists. For further details, see [Phase](#) on page 551.
- **Data Providers** (optional): Used to access values that are used during the execution of the process. For further details, see [Data Providers](#) on page 562. Available internal data providers: Input Message, Process Storage. You can add custom external data providers. For further details, see [External Data Providers](#) on page 564.
- **Process Storage Variables** (optional): Variables that are available for all phases, steps and metrics during the entire life cycle of the process. For further details, see [Process Storage Variables](#) on page 570.
- **Process Metrics** (optional): Use metrics to display the progress of a process. For further details, see [Metrics](#) on page 571.

## Phase

Learn to divide complex process into phases.

You can divide complex processes into several phases, e.g. "Identification", "Qualification", "Cover Buying Center" etc. in the opportunity process. The user can exit multi-phase processes and resume at the same point at a later time.

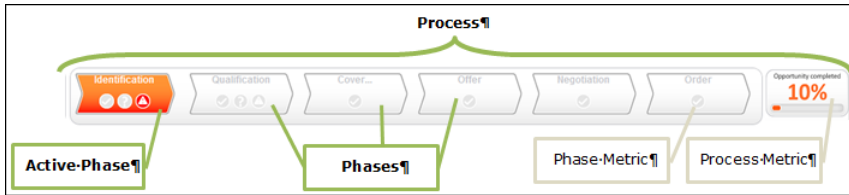
---

**Note:** Phases can not be reused.

---

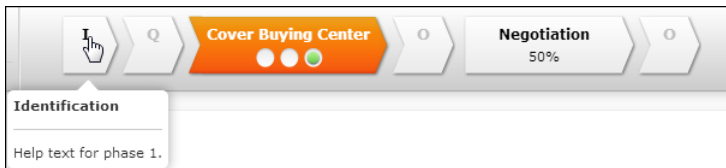
For examples, see [Phase CreateY1](#) and [Phase AddY2](#).

Phases are displayed (as arrows) in the process header. The user can click on the arrows to switch between phases (if configured). Each phase arrow contains the phase name and metrics (if defined). Help texts (phase, step and metrics help) are displayed a tooltip.



If a process contains a lot of phases, you can display past and/or inactive phases in a "collapsed" state by setting the `collapseInactivePhases` and `collapsePastPhases` parameters of the `Process` action. For further details, see [Process Action Parameters](#). Thus users do not have to scroll to access all phase arrows.

**Example:** Both `collapseInactivePhases` and `collapsePastPhases` are set to `true`.



A collapsed phase arrow only displays the first character of the phase label. The complete label and the phase help are displayed in the tool tip. The phase metrics and metrics help is not displayed.

### Defining Phases

To define a phase:

1. Switch to the desired process.
2. Under **Process Phases** click on + (Add).



3. Define the phase:



A phase's configuration consists of the following settings:

- **Phase Name:** The name by which the phase is referenced in the process (only visible in CRM.Designer).
- **Disable navigation to preceding phases:** Check to disable navigation to a previous phase in a process regardless of the conditions defined under **This phase can be activated if**.
- **Description** (optional): Add an internal description (only visible in CRM.Designer).
- **Help Text** (optional): Add a description of what happens in this phase. The **Help Text** is displayed as a tool-tip when positioning the mouse over the phase arrow. If a **Help Text** for the current step exists and the step belongs to the phase, the step help is displayed below the phase help.
- **Phase Label:** The text displayed on the phase arrow in Aurea CRM.Web.
- **Initial Step:** The process step to be called when the phase is initially activated.

At process startup, the property **Initial Step** is used to determine the initial step of a process: The first phase where both the conditions defined under **This phase is active if...** and **This phase can be activated if...** evaluate to true determines the actual initial step of the process.

- **View Template** (optional): currently not implemented
- **This phase is active if...:** Conditions that determine whether the phase is active. Multiple conditions are combined using AND.

A phase is active if all conditions defined under **This phase is active if...** evaluate to true.

The currently active phase is displayed colored with a bold label: 

Inactive phases that cannot be activated are displayed in gray: 

To get information about the active process phase (e.g. for a dispatching condition), use the `getActivePhase()` method. You can use the method `getActivePhase()` wherever the process editor allows entering JavaScript expressions, for instance in the dispatching condition of a process step.

The return value of this method (`getActivePhase()`) is an object containing the properties and methods shown in the following table:

**Note:** The return value can also be "null" if no phase is currently active.

Field	Description
description	A description of the phase (non localized - for documentation only).
helpText	A text containing help for users executing the process.
metrics	An array of metrics to be evaluated for this phase.
name	The name of the phase (non localized - for documentation only).
nr	A number used to order the phases.
stepName	The name of the process step associated with this phase.
text	A text describing this phase.
view	The name of a widget used to display this process phase.
activateConditions	The conditions determining if this phase is currently active.
canActivateConditions	The conditions determining if this phase can become the active phase.
isActive()	returns true if the phase is currently active.
canActivate()	returns true if the phase can be activated.

Example for a dispatching condition based on the name of the active phase:

```
process.getActivePhase().name == "SelectCustomer"
```

- **This phase can be activated if...:** Conditions that determine whether the phase can be activated (i.e. clicked by the user). Multiple conditions are combined using AND.

During the execution of a process, the user can click on a phase. This navigates the process to the **Initial Step** associated with this phase, if all conditions defined under **This phase can be activated if...** evaluate to true. The step receives the original process input as its input message (`$in`).

Inactive phases that can be activated are displayed with a bold label: 

---

**Note:** For conditions in phases only values from the Process Storage and from custom data providers are available. For further details, see [Data Providers](#) on page 562.

---

- **Phase Metrics** (optional): Use metrics to display the progress within a phase (e.g. display "<#>% completed"). For further details, see [Metrics](#) on page 571. Phase metrics are displayed below the phase label. You can add an additional **Help Text** explaining the meaning of the metric.

## Step

Steps are the building units of each phase or process.

A process step is one part in a sequence of steps that define a [Process](#) on page 550.

Each process or process phase consists of one or more process steps. Steps are displayed in the work frame of the current tab.

A process step contains the actual action to be performed:

- a user interface representation, e.g. "Show from xy"
- a background operation, e.g. "Execute workflow z"
- a dispatching operation, e.g. "if x do A, else do B"

A process step is not linked to a process and can therefore be reused in multiple processes. For further details, see [Examples](#) on page 591.

**Process Step 'ProcessAskY1'**

Save Delete Usage Graph Processes...

Step: ProcessAskY1  
 Description: Process Ask page for Process NewOpportunityForCompany  
 Help Text:

**Data Providers**

Name	Action
Process Storage	
Input Message	
Output Message	
Step	

+ New Name

**Action**

Action Template: ProcessAsk  
 Asks the user a question and lets him choose from a predefined list of answers.

Pass arguments as an object  
 End process with this action

Input Arguments

Source	Value	Name
Constant Value	ProcessAskY1	Name
Process Storage	\$.get('curY1Uid')	uid
Process Storage	\$.get('curY2Count')	Y2Count
Process Storage	\$.get('curY2Count')==0?0:1	Y2State

+ New Argumentname

ActionTemplate Output Arguments

command	The command selected by the user.
selectedUid	The uid of the selected record.

Validation

Source	Condition	Error Text
+ New Condition		

Output Arguments

Value	Target	Name
+ New Argumentname		

**Dispatching**

Source	Condition	Next Step	Arguments
Step	\$.get('command')== 'Event1'	SelectKPForY2	
Step	\$.get('command')== 'Event2'	ViewRecord->Exit	uid = \$.get('curY1Uid')

+ New Condition

A step's configuration consists of the following settings:

- **Name:** The name of the step.
- **Description** (optional): Add an internal description (only visible in CRM.Designer).
- **Help Text** (optional): Add a description of what happens in this step. The **Help Text** is displayed as a tool-tip below the phase help. If the process does not contain phases, the **Help Text** is not displayed.
- **Data Providers:** Used to access values used by the step. For further details, see [Data Providers](#) on page 562. All internal data providers are available.

You can add custom external data providers, to get e.g. the result of a query or values from a record. For further details, see [External Data Providers](#) on page 564.

- **Action:** The action to be executed by this step. For further details, see [Action](#) on page 559.

Use the action template `$nop` to define an invisible "dispatching step". For further details, see [\\$nop](#) on page 581. The purpose of a dispatching step is to determine which step is to be executed next. For further details, see [Dispatching](#) on page 560.

- **Input Arguments:** The input arguments for the selected action. For further details, see [Action](#) on page 559.
- **Validation:** Define conditions that must be met to complete the step and call **Dispatching**. For further details, see [Validation](#) on page 571. Multiple conditions are combined using AND. You can define an **Error Text** to be displayed in Aureka CRM.Web if validation fails.
- **Output Arguments:** Output arguments defined in the action are displayed read-only (argument name plus description) in the **Action Template Output Arguments** section.

You can define additional output arguments in the **Output Arguments** section, e.g. to update the process storage. Target data providers: same as for **Input Arguments**.

Value	Target	Name
✗ step.get('uid')	→ Process Storage	curY1Uid
+ New Argumentname		

- **Dispatching:** Defines what happens after the step is completed, i.e. the next step. For further details, see [Dispatching](#) on page 560. The step associated with the first condition that evaluates to true is executed.

If no dispatching conditions are defined or if none of the defined dispatching conditions evaluates to true, the process ends with this step.

## Defining Steps

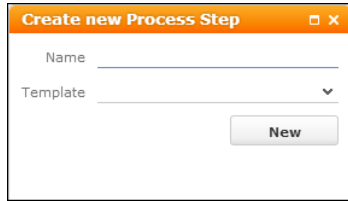
You can define new process steps

- on the **Step** page
- directly in the dispatching area of an existing step

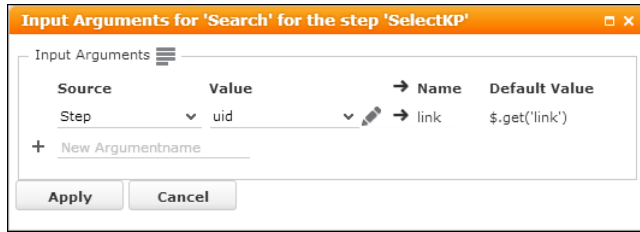
Source	Condition	Next Step	Arguments
✗ Step	\$.get("command")=="Select"	→ SelectKP	+ link = \$.get('uid')
+ New Condition			

To create a new step directly in the dispatching area:

1. Click + (New Step). Enter **Name** and **Template**.



2. Click **New**. Or close the popup to cancel step creation.
3. Click (Edit step arguments) to view and edit the input arguments for the next step.

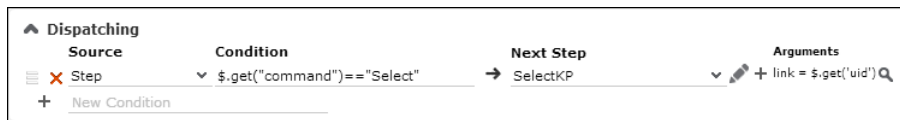


4. Enter an argument name and click + (Add) or click to display all available default arguments.
5. For the arguments you want to pass from the current step (\$out) to the next step (\$in) select "Step" as **Source**. For example, to pass the selected uid from the current step as link for the next step.

In the action definition of the next step you can get these values by simply selecting "Input Message" as source and "as is" as value. For further details, see [Action](#) on page 559.

**Note:** The next step's input arguments are not automatically retrieved from the previous step. If you do not select "Input Message" and "as is", the action template's defaults (source = Constant Value) is used, e.g. `selectionMode = multi` for a `Search` action.

6. Enter the desired **Value**: You can toggle between text input () and a drop-down list of available values ().
7. Click **Apply**.



8. Save the step definition.
9. Click (**Go to step**) to further define the next step (data providers, action definition, dispatching). For steps that do not exist in the database (listed with "not defined yet"), the Edit icon is grayed out.

## Action

Actions are defined on a step to be performed as a part of a process.

In a process step you define an action to be executed in this step. For further details, see [Defining an Action](#) on page 287.



An action's configuration consists of the following settings:

- **Action Template:** The action template used for this action. You can use any available action template.
- **End process with this action:** If checked, the process ends when this action is completed.
- **Input Arguments:** The action's input arguments. Once you select an **Action**, the **Input Arguments** section is populated with the selected action template's parameters. For further details, see [Action Templates](#) on page 350 and [Action Templates within Processes](#) on page 576.

An input argument is defined by **Name**, **Source** and **Value**.

An argument value can be

- a value from a data provider: You can access values from data providers using JavaScript snippets (e.g. `$.get('uid')`). for further details, see [Syntax](#) and [Examples](#) on page 591.
- a constant value: Specify e.g. the name of a configuration unit, the `maxRows` value for a list etc. Constant values are not treated as JavaScript, therefore you do not need to put strings in quotes.


You can toggle between text input () and a drop-down list of available values ()

Available data providers (**Source**):

- Constant Value, e.g. the name of a Process Ask configuration
- Process Storage, e.g. a link record to be used as parent in the current step
- Input Message, e.g. an argument passed from the previous step

- Step
- custom data providers, to get e.g. a field value from a record, see [Step CreateY1](#)

If the data provider "Input Message" is selected, the fixed text "as is" is displayed, indicating that whatever is passed as argument from a previous step (`$out`) is taken as it is and turned into an argument name (`$in`). For further details, see [Step CreateY2](#).

Clicking  in an "as is" row allows you to enter an expression (default: `$.get('<argumentname>')`).

To define additional input arguments (e.g. to perform naming conversions): Enter a **Name** and click + (**Add**).

- **Output Arguments:** The action's **Output Arguments**. The output of the action is always written to `$step`.

If you need additional output arguments, define them in the Output Arguments section of the step definition. For further details, see [Step](#) on page 555.


## Dispatching

Learn to define rules for steps using Dispatching.

In the **Dispatching** section of the step definition you define rules for the next step to be executed. For further details, see [Defining Dispatching Rules](#).

Dispatching			
Source	Condition	Next Step	Arguments
  Constant Value	 <code>process.isSubStep()</code>	 <code>\$return</code>	 
  Process Storage	 <code>!!\$.get('CopyDirectFieldIndices')</code>		 
  Constant Value	 <code>true</code>	 <code>ViewRecord-&gt;Exit</code>	  <code>uid = \$.get('uid')</code>
+ <input type="text" value="New Condition"/>			

The conditions defined here are checked one after the other. The step associated with the first condition that evaluates to true is executed.

To reorder the dispatching rules, click  (**Drag&Drop**) in the respective row and drag it to the desired position.

Dispatching rules can be based on

- a method, e.g. `process.isSubStep()`
- any condition, e.g. a specific record must be available: `!!$.get('companyUId')`
- a command, i.e. the user selects a record or clicks a button in the page header or a button, record or link on a Process Ask page. For further details, see [Dispatching Based on Commands](#).



## Defining Dispatching Rules

To define a dispatching rule and next step:

1. **Condition:** Enter a JavaScript expression to be evaluated. If there is only one possible next step, just enter "true".

If your condition requires values from multiple data providers, you can access these values using the predefined variables `process` or `step`.

**Example:** `$.get('uid') || process.get('myDataProvider.uid')`

2. Click + (**Add**).
3. **Source:** Select the data provider containing the values used by the condition. Available data providers:
  - Constant Value
  - Process Storage
  - Input Message
  - Step: For commands you must always select "Step".
  - custom data providers
4. **Next Step:** Select or create the step to be called if the dispatching condition is met. For further details, see [Defining Steps](#).

## Dispatching Based on Commands

To define a dispatching rule for **default commands** use the following syntax (**Condition**):

```
$.get('command') == 'Select'
$.get('command') == 'Save'
$.get('command') == 'New'
$.get('command') == 'CancelProcess'
etc.
```

If you use a form with a `RecordListBox` widget in a process, use the `command` property of the widget: `command = Proceed<...>` calls the next step.

For buttons, records or links on a **Process Ask** page you need to enter an **Event No.** in the [Process Ask Configuration](#) on page 582. In the dispatching definition these events are referenced as

```
$.get('command')== 'Event1'
$.get('command')== 'Event2'
```

If you want to use **custom buttons** in a process, you need to use the `proceed` method, i.e. the command for the button must follow the naming convention `'Proceed<Name>'`.

Button definition:

```
Action Template: Command
Command: Proceed.ButtonA
```

Dispatching condition:

```
$.get('command') == 'Proceed.ButtonA'
```

## Data Providers

Use data providers to gather the necessary information for a process and to pass information from one process step to another.

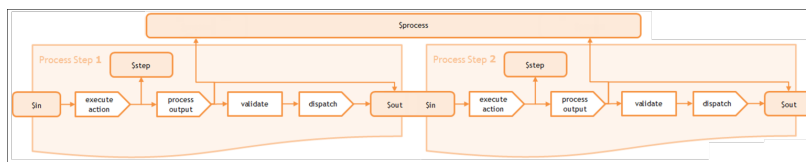
Data providers can access values from process-internal or external sources. During the execution of the process these values can be used

- to get arguments for actions
- in dispatching and phase conditions
- to define validation criteria
- to display process (and phase) metrics

## Internal Data Providers

Learn about the built-in Internal Data Providers available for process, phase or step.

Internal data providers are not configurable. They are created automatically during the execution of the process.



Depending on the context (process, phase, step) the following built-in data providers are available:

- **Process Storage** (`$process`): The process storage is available during the whole process. Use the process storage to share data between process steps. For further details, see [Process Storage Variables](#) on page 570.
- **Step Storage** (`$step`): The step storage is a temporary storage available only within a step. The step storage contains the output of the action executed in this step and builds the basis for decisions made during validation and dispatching.
- **Input Message** (`$in`): The input message contains the process step's input, e.g. the `uid` of the record from which the process was launched.
- **Output Message** (`$out`): The output message contains the step's output, e.g. the `uid` of a newly created record.

Usually the input of a process step is the output of its predecessor step. For further details, see [Dispatching](#) on page 560.

The initial process step receives the input from the process, which in turn forms the initial input message from the arguments passed to the process action template.

## Syntax

To access the values stored in a data provider when defining phases/steps/metrics, select the data provider from the **Source** drop-down list and enter a JavaScript expression like `$.get('uid')`.

`$` always represents the selected data provider.

Example: `$.get('curY1Uid')` gets the record stored in the variable `curY1Uid` in `$process`.



Use JavaScript to define conditions based on these values, e.g.

```
$.get('curY2Count')==0?0:1
```

For more syntax examples, see [Examples](#) on page 591.

The process engine provides access to the values in data providers through small JavaScript expressions like the following:

```
$.get("uid")
```

Javascript expressions are not limited to accessing values in data providers but are also in variety of other situations, such as:

- Conditions for dispatching
- Conditions for process phase activation (`canActivate/isActive`)
- Input-/output-arguments for actions
- Validators
- Metrics
- Process storage entries

**Exception:** For output arguments of an action, `$` always refers to data provider `$step` (which contains the output of the action). The data provider selected in the drop-down list acts as the **target** data provider, that receives the value of the expression.

In addition to the variable `$` accessing the selected data provider there are two predefined variables that can be used in expressions:

- `process`: Refers to the process storage
- `step`: Refers to the current step (only available if the expression is executed in the context of a process step e.g. for dispatching)

Use `process` and `step` to access the values of **all** data providers when defining output arguments and Process Ask pages, >> [Process Ask Configuration](#) on page 582.

Examples:

#### Internal Data Provider

- `process.get('curOpportunityUid')`
- `step.get('uid')`

#### ReadRecordDataProvider

- `step.get('curCompanyInfo.$').get('Name')`

Gets the company name of a record specified in `curCompanyInfo`.

#### ExecuteQueryDataProvider

- `step.get('curActivityCount.$').getNumberOfRows()`

Gets the number of result rows from a query defined in `curActivityCount`.

---

**Note:** For external data providers, you need to specify `.$` at the end of the string to "simulate" the `$` that represents the source.


---

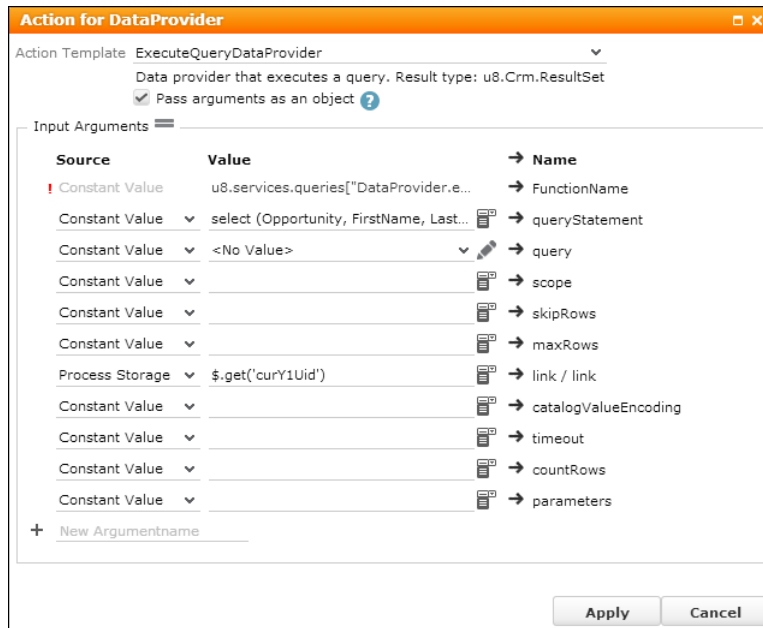
## External Data Providers

Learn about the customized External Data Providers for process, phase or step.

You can create custom data providers to get values from external sources, e.g. the result of a query or the value of a field in the database.

To create a custom data provider:

1. In the **Data Provider** section of a step or process enter a **Name**.
2. Click + (**Add**).
3. Click  (**Edit**).
4. Select an **Action Template**. The available action templates are described in the following chapters.



5. Define the **Input Arguments**, e.g. the `queryString` for an `ExecuteQueryData-  
aProvider`.
6. Click **Apply**.
7. Save the step or process.

For custom data providers using `ExecuteQueryDataProvider` and `ReadRecordDat-  
aProvider` you can define conditions by adding a `$if` argument. If the condition de-  
fined in `$if` evaluate to false, no data is provided.

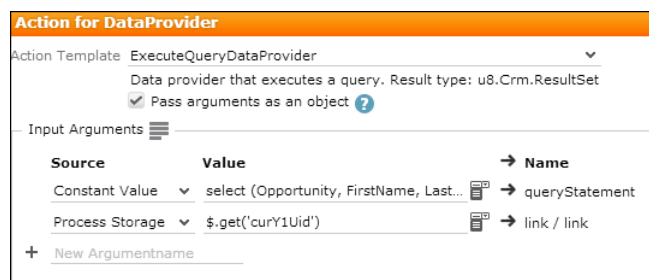
Example: `$if = !!$.get('uid')`

In this case the data provider only returns a result, if the `uid` is passed to the process.  
Create a process storage variable containing a fallback value if the condition eval-  
uates to false: `$.get('<value>') || <fallback value>`

### **ExecuteQueryDataProvider**

Learn about the External Data Providers - `ExecuteQueryDataProvider`.

The `ExecuteQueryDataProvider` executes a query and returns the query results.



The `ExecuteQueryDataProvider` action template supports the following input argu-  
ments:

### query

The query to execute. The query must be saved in the CRM.Designer.

### queryStatement

The query statement to execute. Use the AQL syntax to define the query. For further details, see [AQL Syntax](#) on page 882.

---

**Note:** You need to specify either `query` or `queryStatement`.

---

### maxRows

The maximum number of rows to read.

### skipRows

The number of rows to skip.

### link

An optional link record to read child records, e.g. the opportunity participants (Y2) of the opportunity (Y1) specified by `link`.

### catalogValueEncoding

Indicates how catalog values are encoded: `Numeric`, `Text` or `TextInBaseLanguage`.

### timeout

Timeout (in ms) for this query. If the query takes longer, it is aborted.

### countRows

If set to a non-empty value, the result only contains the number of rows instead of the actual query result.

---

**Note:** You need to explicitly initialize this data provider in the step where you want to use it: Add an input argument (with value 'null') that uses the data provider as source. For example, see [Step EvalY2Count](#).

---

The result of a data provider using the `ExecuteQueryDataProvider` action consists of a list of rows containing values from the Aurea CRM database. You can access the information stored in the result to use in your process:

```
$.getNumberOfRows()
```

Gets the number of rows currently in the result set. The maximum number of result rows is determined by `maxRows`.

```
$.getRowAt(0).values[0]
```

Gets the 1st value from the 1st row in the result set.

```
$.getTotalNumberOfRows();
```

Gets the **total** number of rows in a result set. (You need to set `countRows = true` for this to work.)

For an example, see [Step EvalY2Count](#).

### **ReadConditionalRightsDataProvider**

Learn about the External Data Providers - ReadConditionalRightsDataProvider.

Use the `ReadConditionalRightsDataProvider` to determine which conditional access rights (info area and field rights) are currently applied to a record.

The `ReadConditionalRightsDataProvider` only checks for **conditional** rights. To determine the currently applied "fixed" rights as well, you need to use methods like `u8.Crm.Schema.InfoAreaSchema.canUpdate()`, `u8.Crm.Schema.FieldSchema.canDelete()` etc.

The `ReadConditionalRightsDataProvider` action template supports the following input argument:

#### **uid**

The record for which the conditional rights should be determined.

#### **fieldIds**

A comma-separated list of fields (XML names or numbers) for which the conditional rights should be determined. If undefined, the conditional rights are checked on info area level only.

#### **types**

A comma-separated list of rights "types" to be checked: `CheckViewRight`, `CheckUpdateRight`, `CheckDeleteRight` or `CheckNewRight`. If undefined, the data provider checks for conditional rights of all types.

Use the following syntax to access the results:

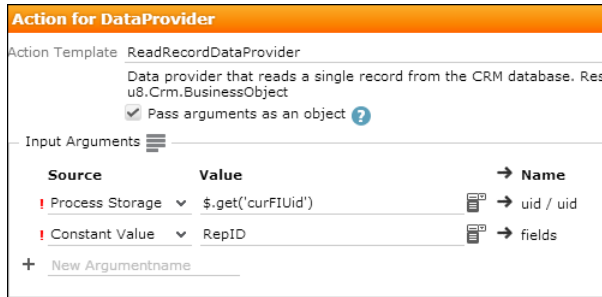
<code>\$.canUpdate(12)</code>	Determines the conditional update right on field no. 12.
<code>\$.canView("Gender")</code>	Determines the conditional view right on field "Gender".
<code>\$.canDelete()</code>	Determines the info area conditional delete right.
<code>\$.getUpdateRight(5).message</code>	Retrieves the message ( <b>Description</b> field from the rights definition) for the conditional update right on field no. 5.

**Note:** In order to be able call the methods `canCreate`, `canView`, `canUpdate` and `canDelete`, make sure that all rights and fields have been loaded , otherwise the results may be inaccurate.

### **ReadRecordDataProvider**

Learn about the External Data Providers - ReadRecordDataProvider.

The `ReadRecordDataProvider` reads data from a single record in the Aurea CRM database.



The `ReadRecordDataProvider` action template supports the following input arguments:

#### **uid (mandatory)**

The uid of the value of the record to be read.

#### **fields (mandatory)**

A comma-separated list of fields (`<field name or number>`) to be read.

#### **linkName (optional)**

Name of a link relation. If defined, the specified link is followed from the given `uid` and the linked record is read.

Examples for `linkName`:

<code>\$Link[FI]</code>	reads the linked company
<code>\$Link[KP]</code>	reads the linked person
<code>\$Link[KP:2]</code>	reads the linked person using link ID 2, see <a href="#">Link IDs</a> on page 19
<code>\$Link[KP;FI]</code>	reads the linked person, or if the person does not exist the linked company

The result of a data provider using the `ReadRecordDataProvider` action consists of the value(s) of the field(s) specified in `fields`. To access these values, enter `$.get('<field name or number>')`.

For an example, see [Step CreateY1](#).



**Note:** You can not mix XML names and field IDs: If you defined your data provider using XML names, you can not get the values by specifying field IDs and vice versa.

### **ReadVariablesDataProvider**

Learn about the External Data Providers - ReadVariablesDataProvider.

The `ReadVariablesDataProvider` reads the values of variables defined in the Z5 info area. For further details, see [Variables](#) on page 53.

The `ReadRecordDataProvider` action template supports the following input argument:

#### **names (mandatory)**

The (comma-separated) list of Z5 variables to be read.

A data provider using the `ReadVariablesDataProvider` action returns the corresponding Z6 variable values as a Javascript object containing a single property called `values`. The values are output in the same order as the requested `names`.

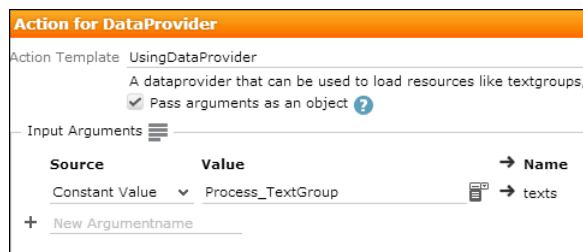
Use the following syntax to access the values:

<code>\$.values[0].value</code>	Gets the value of the first requested variable.
<code>\$.values[1].catNo</code>	Gets the catalog number of the second requested variable.
<code>\$.values[2].catNumericValue</code>	Gets the numeric catalog value (i.e. the catalog code) of the third requested variable.

### **UsingDataProvider**

Learn about the External Data Providers - UsingDataProvider.

The `UsingDataProvider` loads resources like text groups, catalogs and database schema information. These resources can then be used throughout the process. Use the `UsingDataProvider` for processes in a multilingual environment. For further details, see [Texts](#) on page 176.



The `UsingDataProvider` action template supports the following input arguments:

#### **texts**

A comma separated list of text groups to be loaded.

## schema

A comma separated list of info area IDs.

The result of a data provider using the `UsingDataProvider` action consists e.g. in access to all texts of the specified text group(s). To display such texts within the process (e.g. in metrics or as labels on a Process Ask page), you have to read it from the data provider by entering `$(('TXT_<text group>_<text number>'))`.

Example: `$(('TXT_MyTextGroup_3'))` loads text number 3 from text group `MyTextGroup`.

For an example, see [Process NewOpportunityForCompany](#)

## Process Storage Variables

Learn about the Process Storage Variables

When a process is launched, it creates the process storage as defined in the **Process Storage Variables** section of the process configuration. For further details, see [Process](#) on page 550.

Process Storage Variables ?		
Name	Data Provider	Value
✗ curFUID	none	▼ 0
✗ curY1UID	Input Message	▼ \$.get('uid')
✗ curRepName	none	▼ u8.session.identity.repName
✗ curY2Count	none	▼ 0
✗ repMetric	curTexts	▼ \$('TXT_Process_TextGroup_0')
+ New Name		

A process storage variable is defined by **Name**, **Data Provider** and **Value**.

You can define variables for any type of data: record uids, query results, field values, the answer from a web service etc.

The process storage is always created and updated at runtime. Exception: Constant values (Data Provider "none") are only loaded at process startup. (To access "historical" data in your process, you need to write it to a field in the database.)

Since the process storage is reloaded with every process step, defining a lot of process storage variables may affect the performance. Therefore, use the process storage only for values that you really need throughout the entire process.

Available data providers:

- Input Message (`$(in)`): All process storage variables with source `$(in)` must have a value. If a value is missing the process is terminated with an error.
- Custom external data providers, if defined. For further details, see [External Data Providers](#) on page 564
- none: fixed values or functions

Accessing the process storage:

- Phases have read-only access to the process storage: You can use values from the process storage to determine the phase's current state and for displaying phase metrics. For examples, see [Phase CreateY1](#) and [Phase AddY2](#)
- Steps can both read from and write to the process storage. For an example, see [Step CreateY1](#).

## Validation

In the Validation section of a step definition you can define conditions that must be met to complete the step.

Source	Condition	Error Text
Process Storage	<code>\$.get("uids").length &gt; 0</code>	Can not start merging without records.
Process Storage	<code>\$.get("uids").map(u8._.attr("infoAreaId"))...</code>	Can not start merging records from diff...
Process Storage	<code>\$.get("uids").length &lt; 3</code>	Can not start merging more than two re...
Process Storage	<code>!\$.get("defaultMergeOption")    \$.get("mer...</code>	Default merge option must be containe...

The conditions are checked one after the other. If all are met, then [Dispatching](#) on page 560 is called. If a condition evaluates to false, the **Error Text** is displayed.

To define a validation condition:

1. Enter a condition and click + (**Add**).
2. Select the data source used by the condition. Available data providers:
  - Process Storage
  - Input Message
  - Output Message
  - Step
3. Define an **Error Text** to be displayed in Aurea CRM.Web if the condition evaluates to false.

---

**Note:** For steps calling an `Expand` action you need to define mandatory fields via the `Must-Field` field attribute or the Aurea CRM win **Rights** module, since the validation is only called after saving the record.

---

## Metrics

Learn to provide a context for a process and phase metrics.



You can define process and phase metrics (text and/or graphics) to provide a "where am I and what's my status" context for the user. E.g. if the goal of a phase is to create activities, the number of needed activities to reach the next phase can be displayed as a phase metric of type "Text".

Process Metrics ?			
View	Data Provider	Value	Help Text
Text	Process Storage	<code>\$.get('repMetric')+\$.get('c...</code>	The name of the curren...
+			

To define metrics:

1. Select a **View**, i.e. the type of metric you want to display.
2. Select a **Data Providers** on page 562. Available data providers:
  - for process metrics: none, Input Message, Process Storage, custom data providers
  - for phase metrics: none, Process Storage, custom data providers
3. Enter a **Value** or a condition using JavaScript, e.g. `!!$.get('curY1Uid')?100:0` for a "PercentageTrafficLight" metric
4. You can define an (optional) **Help Text** to provide an explanation of the displayed metrics. The **Help Text** is displayed on mouse-over for metrics contained in phase arrows, and on both mouse-over or click for phase metrics displayed on the right side of the header.

Each defined metric is displayed in an extra row.

The following views are available:

- **Html**: Displays text with HTML tags e.g. `<BR>`.
- **Image**: Displays images depending on a condition.
- **Percentage**: Displays percentage values.
- **PercentageTrafficLight**: Displays a traffic light image based on percentage values (red: 0-33 | yellow: 34-66 | green: >66).
- **Text**: Displays text.
- **TrafficLight**: Displays a traffic light image (0: red | 1: yellow | 2: green | -1: no traffic light).

To display multilingual texts (in **Html** and **Text** metrics) define and use a custom data provider with the `UsingDataProvider` action template. For further details, see [UsingDataProvider](#) on page 569.

## Designing a Process

Learn how to design a process.

### Before you start

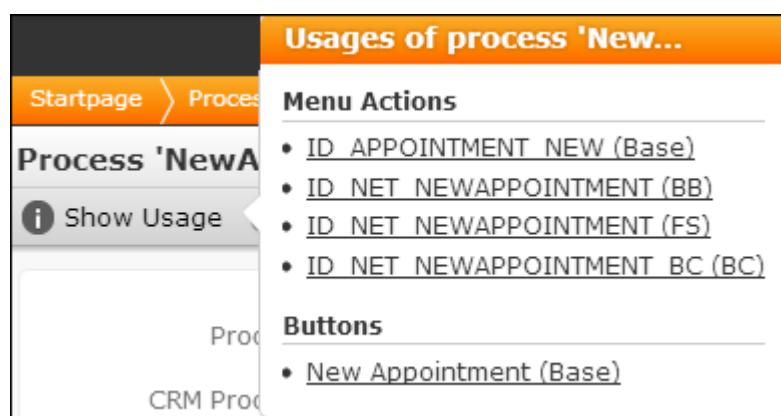
Design the flow of the entire process "on paper":

- Define the goal of the process.
- Identify the necessary steps.

- Define the result of each step.
- Decide if the process should be completed in one session or if you want users to be able to exit the process and resume it at a later time.
- If the process spans multiple user sessions, define the necessary phases.
- If you split the process into phases, define the entry and exit criteria for each phase.
- If using phases, define how users get from one phase to the next (automatically or by clicking on a phase)
- Identify the necessary input arguments (e.g. a parent record) and user input for the process, the phases and steps.
- Decide on adequate data providers: Identify the variables you need to store in the process storage. Define custom data providers instead of entering complex expressions at various point of the process (extendability, performance).
- For each step identify the user interaction needed for step dispatching (Select, Save, custom buttons).
- Decide on helpful metrics to display for the process and the phases.

Identify reusable configuration units:

- Use process steps already defined in UPDATE\_DEFAULT
- If you need to create a new step, identify an existing step to copy from
- Click **Show Usage** on the Processes and Steps pages to display the usage of a process or step.

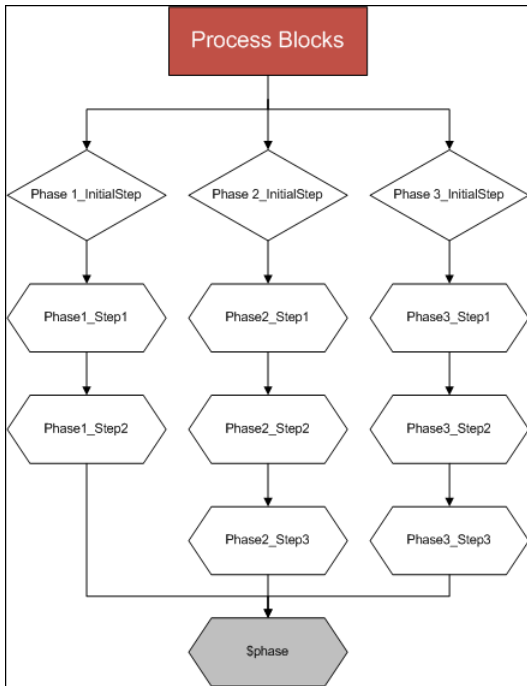


For processes a list of buttons and menu actions calling the process is displayed. For steps a list of processes, phases, steps and sub-steps calling the step is displayed. Click on a link to switch to the respective configuration unit.

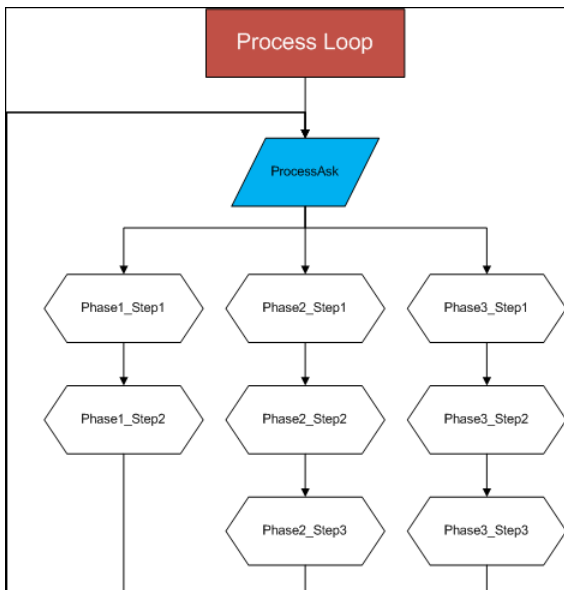
### For Multi-Phase Processes

Decide on the general flow.

You can define phases as blocks returning to an evaluation step  $\$phase$  where the currently active phase is (re-)evaluated:




You can define a loop with each phase returning to a central process ask page which either allows the user to select the next phase or just displays the items relevant for the next phase:



---

## While designing the process

- Click  **Graph** on the **Processes** and **Steps** pages to display the elements and flow of a process (starting with the current step).
- Use the **Description** fields of the various configuration units to document their purpose, open issues, changes etc.

For information on actually defining a process, phases and steps, see [Examples](#) on page 591.

Errors occurring during the execution of a process (syntax errors, faulty storage definition, circular dependencies between data providers etc.) are written to the client log (debug mode). Enter "+process" as filter to display only process-related entries.

## Calling a Process

Learn how to call a process to launch it.

For users to be able to launch a process in Aurea CRM.Web, you need to assign an action using the `Process` action template to a menu item or button. For further details, see [Process](#) on page 550. For examples, see [Examples](#) on page 591.

You can also start a process from within another process by calling the `Process` action from a process step. For further details, see [Step](#) on page 555.

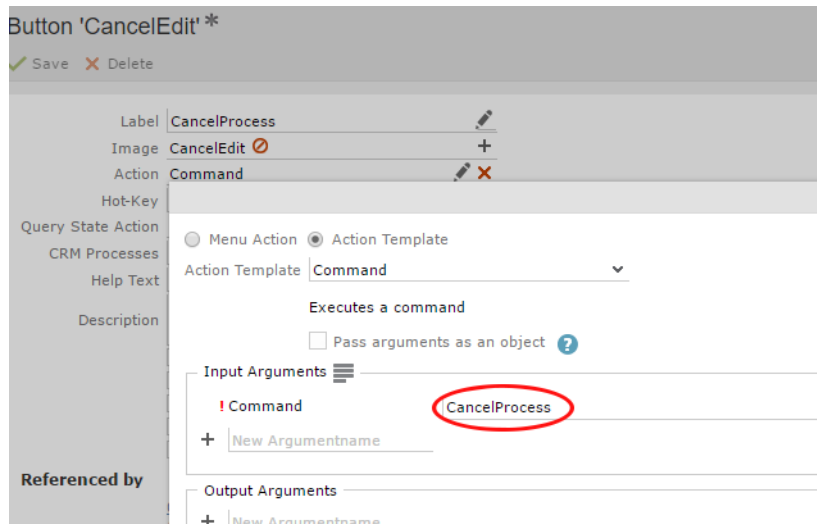
## Cancelling a Process

Learn how to cancel a process in this topic.

You can allow your users to cancel a process at any step of the process. The system is returned to the state it was in before the process is called. To do this you need to bind the command `cancelProcess` to the **Process** action template of a menu or button. Then call the dispatch condition `$.getCommand('command')='CancelProcess'` on a process step to allow the user to cancel the process.

The following screenshot shows the configuration of a button to cancel a process.

Binding the `CancelProcess` command to a **Process** action template of the button.



The following screenshot shows the configuration of the dispatch rule on a process step to cancel the process.

Dispatching		Next Step
Source	Condition	
Input Message	\$.get('viewRecord')	ViewRecord->Exit
Step	step.isSubStep() && \$.get('uid')	\$return
Step	process.page.url.get("onProceed")	\$exit
Step	\$.get('command') == 'Save'	ViewRecord->Exit
Step	\$.get('command') == 'CancelProcess'	\$cancel

For information on assigning an action to the **Process** action template, see [Process](#) on page 576. For more information on dispatching, see [Dispatching](#) on page 560.

## Action Templates within Processes

Learn about the Action Templates within Processes

The following chapters describe the usage of action templates within processes.

### Process

Learn about ProcessStep action template within processes.

Use the `Process` action template (`process.aspx`) to call a process (via a menu item or button). For further details, see [Calling a Process](#) on page 575.

You can also start a process from within another process by calling the `Process` action from a process step. For further details, see [Step](#) on page 555.

### Process Action Parameters

The `Process` action template supports the following input arguments:

#### name

The name of the process to be launched, e.g. `NewAppointment`, `MergeRecords` etc.



## uid

The record from which the process is launched. The record's `uid` is passed to the process's input message and used as input argument for the initial step. For further details, see [Data Providers](#) on page 562.

`Record`: the current record.

`Linked`: the parent record of the current record.

## link

An optional link record passed to the process's input message (`Record` or `Linked`).

## processState

Set to "RequireEnabled". Cannot be changed.

The process called by this action must be available the current user's role, otherwise it can not be executed. For further details, see [Roles & CRM Processes](#) on page 727.

## createNewTab

See [createNewTab](#).

## createNewPopup

Starts the process in a popup. For further details, see [createNewPopup](#).

## recordSet

When starting a process from a list, query or the calendar you may want to pass more than one record to the process. To do this define the `recordSet` attribute. Available values:

The record set is written to the Input Message (`$in`) of the process.

To get the contents of the record set (e.g. as process storage variable) use  
`$.get('recordSet')`

To filter the contents of a record set, you can specify one or more info area IDs:

```
$.get('recordSet').getRecordSet('MA') // only activity records
```

```
$.get('recordSet').getRecordSet('MA', 'KM') // only activities and tickets
```

To get a list of all uids stored in the record set enter:

```
$.get('recordSet').getUids()
```

To read a single record from the record set use the following syntax:

```
$.get('recordSet').getUids()[<recordIndex>]
```

Example:

```
$.get('recordSet').getUids()[0]
```

You can list records from a record set on a Process Ask page by using the Records item. For further details, see [Records](#) on page 589.

### collapseInactivePhases

If set to `true`, phases that can not be activated by the user are displayed collapsed.

**Default:** false.

### collapsePastPhases

## Search

Learn about search action template within processes.

The screenshot shows the 'Search Persons' form within an 'Opportunity Process'. The form includes a header with 'Create Opportunity' and 'Add Participant' buttons, and a 'Rep: Susanna Mann' dropdown. The search fields include:
 

- Name: First Name, Last Name
- Department: + v
- + Add more ... v
- Rep ID: [input field]
- Company: [input field]
- With me:

 At the bottom, there is a 'max. 15 Records / Page' dropdown and 'Reset' and 'Start Search' buttons.

Use the `Search` action template within a process to let users select an existing record from a specific info area (optional: linked to a parent record). You need to call the `Search` action from a process step, see [Step](#) on page 555.

## Expand

Learn to use Expand Action template within processes.

The screenshot shows the 'Expand' action template within an 'Opportunity Process'. The form displays a list of opportunity details:
 

- Opportunity: My Opportunity
- Opp. Type: Sales
- Rep ID: Test User
- Status
- Completion Date
- Level: Level 4
- Priority
- Completion

Use the `Expand` action template within a process to let users view/edit an existing record or create a new one. You can call the `Expand` action from a process step or (read-only) from within a Process Ask configuration. For further details, see [Step](#) on page 555 and [Expand View](#) on page 585.

## ProcessAsk

Learn to use ProcessAsk Action template within processes.

The screenshot displays the 'ProcessAsk' interface for an 'Opportunity Process'. At the top, there are buttons for 'Create Opportunity' and 'Add Participants', along with the text 'Rep: Lisa Simpson'. The main content area is divided into two sections: 'Current State' and 'Add Participants'. The 'Current State' section shows a table of fields: Opportunity (My Opportunity), Opp. Type, Rep ID, Status, Completion Da..., Level (Level 0), Priority, and Completion. The 'Add Participants' section lists two participants: Andrea Gabrieli (Gabrieli) as Contact Person and John Doe (update software AG) as External Consultant. Below the list, a message states 'You have 2 participant(s) linked to your opportunity.' and there are buttons for 'Add Participant' and 'Show Opportunity'.

Use the `ProcessAsk` action template to

- provide information about previous events
- display the current state of a record
- provide the user with a list of to-dos for the current process step or phase
- provide the user with a choice of follow-up steps in a process
- ask the user a question and fire different events depending on the answer

A Process Ask page consists of

- a process step calling the `ProcessAsk` action. For further details, see [ProcessAsk Action Parameters](#)
- a Process Ask configuration. For further details, see [Process Ask Configuration](#) on page 582.

For an example, see [Step ProcessAskY1](#) and [Process Ask Configuration ProcessAskY1](#).

### ProcessAsk Action Parameters

The `ProcessAsk` action template supports the following input arguments:

#### Name

The name of the Process Ask configuration defining the items and functionality of the Process Ask page. For further details, see [Process Ask Configuration](#) on page 582.

This parameter is mandatory.

### uid/uid1-3

The uid of a record. Use `uid1-3` to pass additional records.

The first `uid` specified in the `ProcessAsk` action is automatically used for an Expand View item in the Process Ask configuration. (If you have more than one Expand View item on your Process Ask page, you need to specify the required uids as parameter of each Expand View. For further details, see [Expand View](#) on page 585.)

To pass more than four records to a Process Ask page use the `recordSet` argument described below.

To use the records passed with `uid/uid1-3` in texts on the Process Ask page (see [Process Ask Configuration](#) on page 582) the following placeholders are available:

- `{RECORD}` the table caption of the record passed with `uid`
- `{INFOAREA}` the name of the record's info area
- `{INFOAREAS}` the plural of the info area name

The values of `RECORD`, `RECORD1` etc. (i.e. the table captions) are passed to the output message of the Process Ask: You can get these values (e.g. to display in metrics) from `$step` via the expression `$.get('record')`, `$.get('record1')` etc.

### FormName

The name of a form to be displayed instead of the Process Ask page. If both `Name` and `FormName` are specified, `FormName` is used.

### NewInfoArea

The info area ID to be used for the placeholders `{INFOAREA}` and `{INFOAREAS}`.

If undefined, `uid` is used as a fallback.

### Scripts

Scripts to be loaded on the page.

### recordSet

If you want to display more than four records on a Process Ask page in a Records item use the `recordSet` argument.

At process startup the record set is written to the Input Message using the `recordSet` argument of `Process`, described in [ProcessAsk Action Parameters](#). You can then create a process storage variable containing the record set. In the `ProcessAsk` step get the records from the process storage by entering e.g. `$.get('myRecordSet')`.

For more information, see [Records](#) on page 589.

For an example, see [Passing Multiple Records to a Process Ask page](#) on page 620

---

## ExecuteSubStep

Learn to use ExecuteSubStep Action template within processes.

Use the `ExecuteSubStep` action template to call sub-steps for executing routine tasks (e.g. creating a child record) or to call another process.

The `ExecuteSubStep` action template supports the following input arguments:

### name

The sub-step to be executed.

### processName

The (sub-) process to be launched.

You must specify either `name` or `processName`.

`ExecuteSubStep` has no default output arguments.

## ExecuteSelectRecordSubStep

Learn to use ExecuteSelectRecordSubStep Action template within processes.

Use the `ExecuteSelectRecordSubStep` action template to call sub-steps for executing routine tasks like selecting a record and creating it if it does not exist.

Example: the `SelectCompany->Return` step called by the `ExecuteSelectRecordSubStep` action allowing the user to search for or create a company, see [Step SelectFIForY1](#).

The `ExecuteSelectRecordSubStep` action template supports the following input argument:

### name

The sub-step to be executed.

Default output argument of `ExecuteSelectRecordSubStep`:

### uid

The uid of the selected record.

## \$nop

Learn to use \$nop Action template within processes.

Use the `$nop` action template to define background steps (without user interface) to:

- recalculate values (evaluation step) for dispatching purposes
- initialize a data provider

For an example, see [Step EvalY2Count](#).

## Process Ask Configuration

The Process Ask Configuration defines the items to be displayed on a `ProcessAsk` page.

The Process Ask Configuration defines the items to be displayed on a `ProcessAsk` page.

To define a Process Ask configuration:

1. On the CRM.Designer main page in the **Processes** area click **Process Ask**.

All available Process Ask configurations are listed. You can add, copy, delete or view configurations. For further details, see [Designing Configuration Units](#) on page 64.

2. Define all necessary items for your Process Ask page in the order you want to display them.

Process Ask 'ProcessAskY1'

✓ Save ✗ Delete

Description

Modify the items that will be displayed on the process ask page:

Type	Event No.	DisplayText	Action	TableCaptionName	CRM Process	Max. rows	Parameters
✗ Caption		Current State			all	Q	Parameters
✗ Expand View					all	Q	{expandName:'Y1.Process'}
✗ Separator					all	Q	
✗ Caption		Add Participants			all	Q	Parameters
✗ Linked Records	0	Display text		Y2.Process	all	Q 5	{linkName:'\$Children[Y2].showGridHeader:fals...
✗ Text		You have {url:Y2Count} par...			all	Q	{image:'CAT_ZF_{url:Y2State}'}
✗ Button	1	Add Participant			all	Q	Parameters
✗ If							{url:Y2Count}>0
✗ Button	2	Show Opportunity			all	Q	Parameters
✗ End If							

The available items and their mandatory and optional settings are described in the following chapters.

---

**Note:** The maximum length of the **Parameters** field is 128 characters.

---




---

**Note:** The maximum length of the **TableCaptionName** field is 28 characters.

---

You can assign one or more **CRM processes** to each item (except for if...end if items). For further details, see [Roles & CRM Processes](#) on page 727

To prevent empty lists from being displayed on a `ProcessAsk` page, use the `hideIfEmpty` property. Example: Define `{infoAreaId: "KP", hideIfEmpty: true}` as **Parameters** for a favorites item, to completely hide this item if no KP favorites are available.

To reorder items, click  (**Drag&Drop**) in the respective row and drag it to the desired position.

3. Click **Save**.

4. Enter the Process Ask configuration as `Name` in the `ProcessAsk` action definition.

To access parameters and values from the process and step storage:

- `{url:param}`: All input arguments from the `ProcessAsk` action are written to the URL. You can access these parameters via `{url:'urlParamName'}`. You can use URL parameters wherever text can be entered in the Process Ask configuration.

Example: `{url: uid1}` gets the uid specified by input argument `uid1`.

URL parameters can contain process variables, Web Configuration parameters etc. - anything that is passed from the `ProcessAsk` action template. For more examples, see [Process Ask Configuration ProcessAskY1](#)

---

**Note:** Using URL parameters makes a Process Ask configuration reusable and data provider-independent.

---

- To access the process or step storage use `process.get('param')` and `step.get('param')`, where `param` is the name of a parameter in the process or step storage. For further details, see [Syntax](#).
- You can use the following placeholders within texts or parameters:
  - `{RECORD}`: Displays the table caption of the record specified by input argument `uid`. It is available for all other uids as well: `{RECORD1}` is replaced by the table caption of `uid1` etc.
  - `{INFOAREA}`: Displays the name of the info area of `uid`, e.g. "Company" if the record specified by `uid` is a company record. Also available for `uid1-3`.
  - `{INFOAREAS}`: Same as `{INFOAREA}` but in plural, e.g. "Companies".

## Button

You can define possible answers as buttons.

Users choose an answer by clicking on a button.

Person	Person in Company	Company	Favorites	Property Participant	Opportunity Participant
--------	-------------------	---------	-----------	----------------------	-------------------------

Specify the following:

- **Event No.:** The number of the event to be fired if the user clicks the button. The actual events are defined in the **Dispatching** area of the step definition, see [Dispatching](#) on page 560.

Source	Condition	Next Step	Arguments
Step	\$.get('command')== 'Event1'	→ SelectPerson->CreateContact	
Step	\$.get('command')== 'Event2'	→ SelectCompany->CreateContact	
Step	\$.get('command')== 'Event3'	→ SelectFavorite->CreateContact	
Step	\$.get('command')== 'Event4'	→ SelectProperty->CreateContact	
Step	\$.get('command')== 'Event5'	→ SelectOpportunity->CreateContact	
Step	\$.get('command')== 'Event6'	→ CreateContact->ViewRecordOrReturn	link = \$.get('selectedUid')

- **DisplayText:** The button's label.
- **Parameters** (optional): The following parameters are available:
  - `image`: You can specify an image to be displayed and its minimum size, e.g.:  
`{image: "MyImage", minWidth: "120px"}`.

The image must be registered in CRM.Designer, see [Images](#) on page 717.

- `tooltip`: You can specify a tooltip for the button:

```
{tooltip: "This is a help text."}
```

Or you can reference a (multi-language) CRM.Designer text:

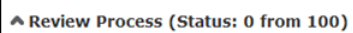
```
{tooltip: "TXT_<textgroup>_<number>"}
```

The default text group for button tooltips is `ButtonHelp`.

You can use the Markdown syntax to format your help text, see <http://en.wikipedia.org/wiki/Markdown>.

## Caption

Caption is the heading of a section (e.g. the purpose of the process, the actual question for the user etc.)



Caption items display an **Expand/Collapse** toggle button to show/hide all items below that caption. Use this item to organize the **Process Ask** page.

Specify the following:

- **DisplayText:** The caption text.
- **Parameters** (optional): Specify an image to be displayed with the caption: `{image: 'my_image'}`

The image must be registered in CRM.Designer. For further details, see [Images](#) on page 717.



## Expand View

Displays the details of a record (e.g. to provide information on the context of the process).

Company	ABC AG	City	Wien
ZIP	1010	Country	Austria
Street		Office R...	
Rep ID			

The Expand view is read-only, all editing functions are disabled. You can combine information from multiple records and info areas by defining multiple **Expand View** items.

If you passed a `uid` to the `ProcessAsk` action, you don't need to specify it here. The first `uid` from the `ProcessAsk` is used. Otherwise you must specify the `uid` in the **Parameters** field.

## Syntax

```
{model: {uid: <uid>, <param>: <value>, ...}}
```

If you specify only parameters for the `model`-property, i.e. only `Expand` action parameters (e.g. `uid` and `expandName`, see [Expand Action Parameters](#) on page 356) you can use

```
{uid: <uid>, <param>: <value>, ...}
```

Example:

```
{uid: <uid>, expandName: '<nameOfAExpandConfiguration>'}
```

You can specify `<uid>` as one of the following:

- as URL-Parameter, e.g. `{url:uid}` or `{url:uid2}`
- as value from the process storage, e.g. `process.get('myUid')`
- hard-coded, e.g. `{infoAreaId: "FI", recordId: "x00000064000009f9"}` or `"FI;x00000064000009f9"`

## Favorites

Displays a list of **Favorites** records for the user to choose from. Double-clicking a record fires the event.

Select from "Favorites"
Justsharp
OTC Testfirma
Praxis Ville
Cork City FC

Specify the following:

- **Event No.:** The number of the event to be fired if the user selects a record. The actual event is defined in the **Dispatching** area of the step definition.
- **DisplayText:** The heading of the record list.

- **TableCaptionName:** The name of the table caption to be used for displaying the records.
- **Max. rows:** The maximum number of records to be displayed.

---

**Note:** If you don't specify **Max. rows** for an item, the value specified in the `Search.MaxResults` Web Configuration parameter is used instead. For further details, see [Search.MaxResults](#).

---

- **Parameters** (mandatory): The info area from which to display the favorites.

## If...End If

Allows you to define conditional parts of the Process Ask action template (e.g. to use the same Process Ask configuration for multiple process phases).

If	0		{url:Y2Count}>0
Button	2	Show Opportunity	Parameters
End If	0		

All items between an **If** and **End If** item are displayed only if the expression passed in the **Parameters** field evaluates to true. If you don't define any condition in the **Parameters** field, this evaluates to true as well.

You can embed if-loops within other if-loops.

## Syntax

```
{url:<variable>} >= <value>
process.get('<variable>') == true
```

## Link

You can define possible answers as hyperlinks. Users choose an answer by clicking on a link.

You can define possible answers as hyperlinks. Users choose an answer by clicking on a link.

>  [current status: 100 \(from 100\)](#)

You can specify an event or an action for a Link item. If you specify an event, you can use Link items to call process steps (allowing the users to navigate between steps).

Specify the following:

- **Event No.:** The number of the event to be fired if the user clicks on the link. The actual event is defined in the **Dispatching** area of the step definition.
- **DisplayText:** The hyperlink's text.
- **Action:** The action to be executed if the user clicks on the link.

---

**Note:** If both **Event No.** and **Action** are specified for a Link item, only the action is used.

---

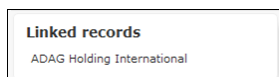
- **Parameters** (optional): Specify an image to be displayed with the link: `{image: 'my_image'}`

The image must be registered in CRM.Designer. For further details, see [Images](#) on page 717.

You can define variable images depending on some value (e.g. `{image: 'COMPLETION_{url:part1Image}'}`) or directly enter conditions for an image in the **Parameters** field.

## Linked Records

Displays a list of linked records for the user to choose from. Double-clicking a record fires the event.



Specify the following:

- **Event No.:** The number of the event to be fired if the user selects a record. The actual event is defined in the **Dispatching** area of the step definition.
- **DisplayText:** The heading of the record list.
- **TableCaptionName:** The name of the table caption to be used for displaying the records. (If not defined, the default table caption of the info area specified by the link relation in **Parameters** is used.)
- **Max. rows:** The maximum number of records to be displayed.

---

**Note:** If you don't specify **Max. rows** for an item, the value specified in the `Search.MaxResults` Web Configuration parameter is used instead, see [Search.MaxResults](#).

---

- **Parameters** (mandatory): The link to be used, e.g. `$Link[FI]` or `$Children`. For possible values of `<linkName>`, see [Link Names in JavaScript](#) on page 22.

**Note:** The link is always read in relation to the first `uid` from the `ProcessAsk` action.

In addition to the `linkName` parameter you can specify any parameter supported for the `RecordListBox` widget. You can e.g. use `filterName` to filter the list of linked records.

### Examples:

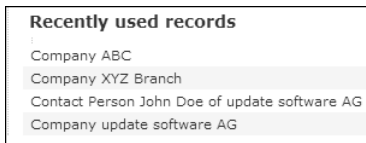
```
{linkName: '$Children[Y2]', showGridHeader: false, grid:{emptyGridTextAlign: 'left'}}
```

```
{linkName: '$Children[KM]', filterName: "KM.myFilter"}
```

For the complete example, see [Step ProcessAskY1](#) and [Process Ask Configuration ProcessAskY1](#).

## Most Recently Used

Lists the records from the **History** for the user to choose from. Double-clicking a record fires the event.



Specify the following:

Type	Event No.	DisplayText	Action	TableCaptionName	Max. rows	Parameters
Caption	0	Add a new activity			0	Parameters
Step Caption	0	With whom do you want to c...			0	Parameters
Most Recently Used	1	Recently used records		TableCaption name	20	KP

- **Event No.:** The number of the event to be fired if the user selects a record. The actual event is defined in the **Dispatching** area of the step definition.
- **DisplayText:** The heading of the record list.
- **TableCaptionName:** The name of the table caption(s) to be used for displaying the records. If undefined, the info area's default table caption(s) are used.
- **Max. rows:** The maximum number of records to be displayed.

**Note:** If you don't specify **Max. rows** for an item, the value specified in the `Search.MaxResults` Web Configuration parameter is used instead, see [Search.MaxResults](#).

- **Parameters** (optional): The info area from which to display records. If undefined, all records from the **History** are displayed.

## Records

If multiple records are passed to the `ProcessAsk` via `uid`, `uid1-3` and `recordSet` you can use the `Records` item to list these records on the `Process Ask` page.

RecordSet:
Alpenbank Nord AG1
American Express
ARCANA chem.-pharm. Fabrik Gesellschaft m.b.H.
Arkansas Consulting GmbH

Specify the following:

- **DisplayText:** The heading of the record list.
- **TableCaptionName:** The name of the table caption to be used for displaying the records. (If undefined, the default table captions are used.) If no table caption exists for a record, the record is not displayed.
- **Max. rows:** The maximum number of records to be displayed.

---

**Note:** If you don't specify **Max. rows** for an item, the value specified in the `Search.MaxResults` Web Configuration parameter is used instead. For further details, see [Search.MaxResults](#).

---

- **Parameters:** Specify which records are listed (no brackets, no quotes):
  - `RECORDS` The records from `uid`, `uid1`, `uid2` and `uid3` are listed.
  - `RECORDLIST` The records from `recordSet` are listed.
  - `RECORDLISTS` Both the records from `uid`, `uid1-3` and `recordSet` are listed.

If undefined, `RECORDLISTS` is used.

For an example, see [Passing Multiple Records to a Process Ask](#) page on page 620.

## Separator

Inserts a separator line.

## Step Caption

Displays bold text e.g. to highlight important information or to ask the user a question.

How do you want to continue?

Specify the following:

- **DisplayText:** The text.
- **Parameters** (optional): Specify an image to be displayed with the step caption:
 

```
{image: 'my_image'}
```

The image must be registered in CRM.Designer, see [Images](#) on page 717.

## Text

Any kind of text (e.g. the description of an action to be performed by the user).

Click on "Select Company" to proceed ...

Specify the following:

- **DisplayText:** The text.

You can use URL-Parameters to access variable values, e.g. "You have {url:Y2Count} participant(s) linked to your opportunity.". You can **not** access the process storage in text items. Therefore you need to define all required information as input arguments of the `ProcessAsk` action.

You can also use the following placeholders: `{RECORD}`, `{INFOAREA}` and `{INFOAREAS}`, `>>` .

- **Parameters** (optional): Specify an image to be displayed with the text: `{image: 'my_image'}`

The image must be registered in CRM.Designer. For further details, see [Images](#) on page 717.

You can define variable images depending on some value (e.g. `{image: 'CAT_ZF_{url:Y2State}'}` shows a different smiley depending on the current value of `Y2State`). For a complete example, see [Step ProcessAskY1](#) and [Process Ask Configuration ProcessAskY1](#).

You can also directly enter conditions for an image in the **Parameters** field.

Use images (bullets and numbers) if you want to create a formatted list. (Entering the bullets or numbers directly in the text diminishes the re-usability of texts.)

---

**Note:** Use empty **Text** items to control the vertical spacing on your page.

---

## Standard Process Steps

UPDATE\_DEFAULT contains some standard process steps useful for navigating within a process.

Those standard process steps are:

### \$cancel

Use `$cancel` to end a process: Aurea CRM.Web navigates back to the page from which the process was started. If the process was executed in a popup, the popup is closed, if the process is executed in a new tab, the tab is closed.

## \$exit

Use `$exit` to end a process without navigation: The last (interactive) process step is displayed.

## \$phase

Use the `$phase` step to navigate to the initial step of the active phase. It automatically re-evaluates all phase conditions and thus determines the active phase.

## \$return

Use `$return` to return from sub-steps: Define `$return` as **Next Step** to call the step from which the current step is called. Example in `UPDATE_DEFAULT`: Dispatching of step `SelectCompany->Return`.

## ViewRecord->Exit

Use this step to end a process.

This step calls the `ViewRecord` action with input argument `uid` executing the default action for the record and ending the process.

## Examples

Examples for Processes.

Below are few examples for your reference:

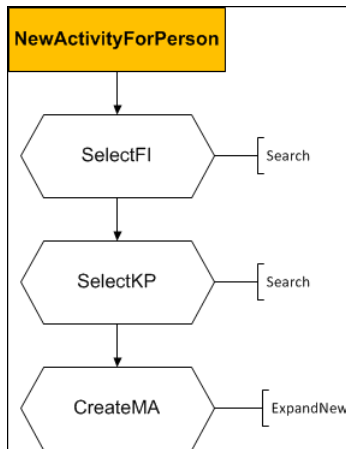
### NewActivityForPerson

An example for a simple process.

This is an example for a simple process that creates an activity record (MA) for a person (KP) in a company (FI). The process contains no header, no phases, metrics etc.

The process is started from the application menu, i.e. without context.

The following graphic displays the process flow, the steps and which actions they call.



The following chapters list the necessary configuration units.

## Process NewActivityForPerson

Processes

On the CRM.Designer main page, click **Processes** to access all available process configurations and define new processes.

To define a process:

1. Select the **Initial Step** of the process. If you divide your process in multiple phases, this initial step is used as a fallback if no active phase is found.

For this example create a new process called `NewActivityForPerson`.




2. Select the **Initial Step** `SelectFI`.

---

**Note:** You need to create the steps before you can select them.

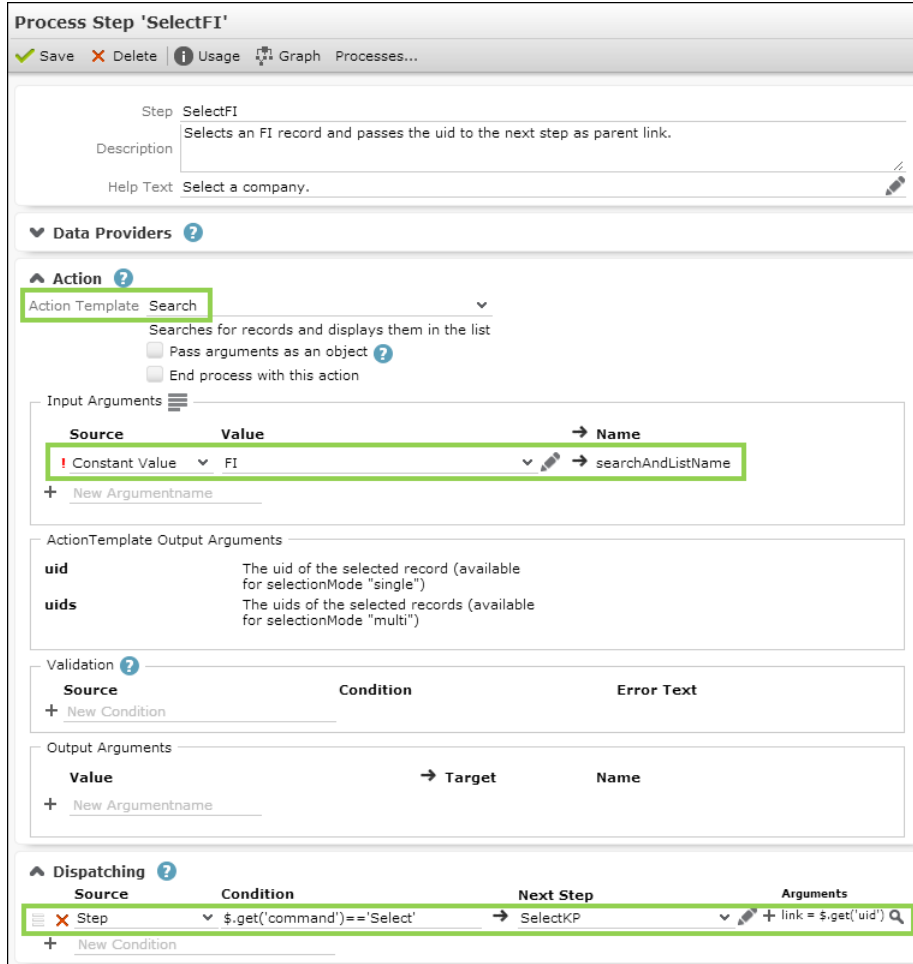
---

Once you have selected a step you can click  (**Edit**) to edit the step.



## Step SelectFI

1. Create a step called `SelectFI`.



2. Select the action (`Search`) and define its input arguments.

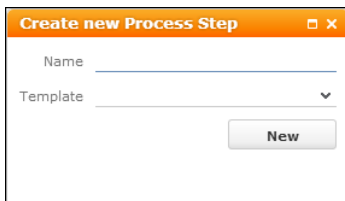
The search results automatically use the context menu `M_Wizard` containing the action `A_Select` which allows users to select a record.

3. Define the **Dispatching**:


- a. Enter the condition: `$.get('command')==\'Select\'`  
Selecting a record navigates to the next step.
- b. Select "Step" as data provider (**Source**).


4. Select or create the **Next Step**:

- a. Under **Next Step** click + (**New Step**).



- b. Enter "SelectKP" as **Name**.

- c. Select the `Search` action template as **Template**.
  - d. Click **New**.
5. Define the input arguments for `SelectKP`.
- To pass the `uid` of the selected FI record to the next step as `link`:
- a. Under Arguments click  (**Edit step arguments**).



Source	Value	Name	Default Value
Step	uid	link	\$.get('link')

+ New Argumentname

Apply Cancel

- b. For the arguments you want to get from `$out` select "Step" as **Source**.
  - c. Select the desired **Value** (in this case `uid`) from the drop-down list.
  - d. Click **Apply**.
6. Save the step definition.

## Step SelectKP

1. Create a step called `SelectKP`.

**Process Step 'SelectKP'**

Save Delete Usage Graph Processes...

Step: SelectKP  
 Description: Selects a KP record and passes the uid to the next step as parent link.  
 Help Text: Select a person.

**Data Providers**

**Action**  
 Action Template: Search  
 Searches for records and displays them in the list  
 Pass arguments as an object  
 End process with this action

**Input Arguments**

Source	Value	Name
Constant Value	KP	searchAndListName
Input Message	as is	link

**ActionTemplate Output Arguments**

**uid**: The uid of the selected record (available for selectionMode "single")  
**uids**: The uids of the selected records (available for selectionMode "multi")

**Validation**

Source	Condition	Error Text
+ New Condition		

**Output Arguments**

Value	Target	Name
+ New Argumentname		

**Dispatching**

Source	Condition	Next Step	Arguments
Step	\$.get('command')== 'Select'	CreateMA	+ link = \$.get('uid')
+ New Condition			

Since the step's input argument `link` comes from the dispatching definition of the previous step, you only need to select "Input Message" as **Source** and select "as is" from the **Value** drop-down list.

2. Define the dispatching as described for `SelectFI`.

## Step CreateMA

1. Create a step called `CreateMA`.

**Process Step 'CreateMA'**

Save Delete Usage Graph Processes...

Step: CreateMA

Description: Creates an activity record for the selected person. The process ends with this step.

Help Text: Create an activity record for the selected person. The process is completed once you save the reco...

**Data Providers**

**Action**

Action Template: ExpandNew

Creates a new record in expanded view.

Pass arguments as an object

End process with this action

Input Arguments

Source	Value	Name
Constant Value	MA	infoAreaId
Input Message	as is	link
Constant Value	none	history

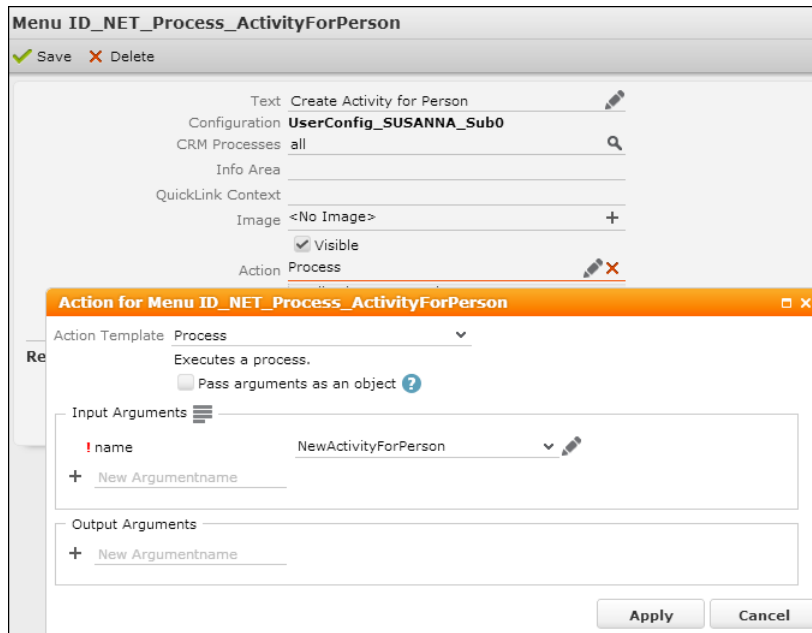
+ New Argumentname

2. Select the action (`ExpandNew`) and define its input arguments (again taking `link` "as is" from the previous step).
3. Check **End process with this action**.

Once the user saves the new MA record the process ends with displaying an Expand view of the record.

## Menu Action ID\_NET\_Process\_ActivityForPerson

1. To call your process create an application menu action `ID_NET_Process_ActivityForPerson` in `ID_NET_APPOINTMENT_MGMT`. For further details, see [Configuring a Menu](#) on page 292.



2. Select `Process` as **Action Template**.
3. Enter the name of the process as input argument `name`.

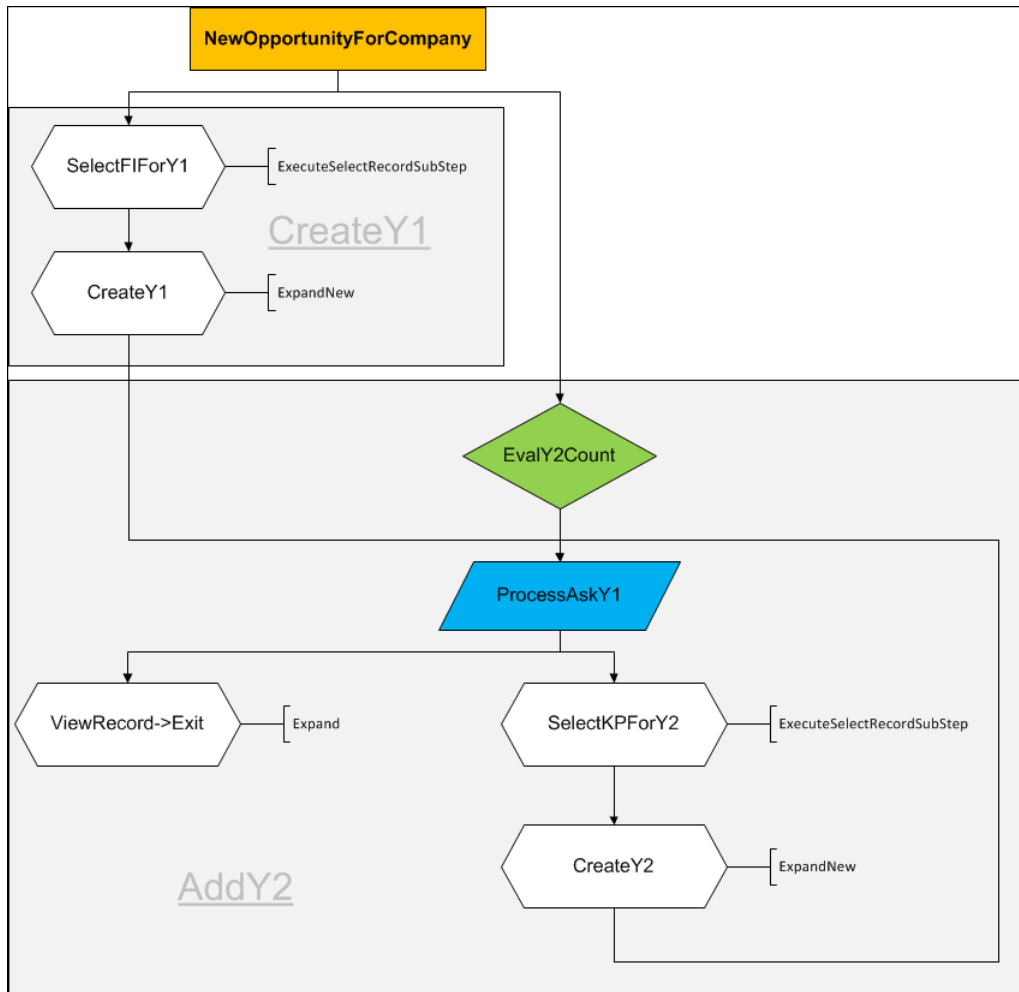
## NewOpportunityForCompany

An example for a slightly complex process.

This is an example for a slightly more complex process that creates an opportunity (Y1) for a company (FI) and allows users to add opportunity participants (Y2) at a later time. The process consists of two phases, it contains a header, a Process Ask configuration and (phase) metrics and uses different data providers and process storage variables.

The process can be called from both the application menu and an existing opportunity's context menu.

The following graphic displays the process flow, the phases and steps and which actions they call.



The following chapters list the necessary configuration units.

## Process NewOpportunityForCompany



## 1. Create a new process called `NewOpportunityForCompany`.

Process 'NewOpportunityForCompany'

CRM Process all

Header: `ProcessHeader_Y1`

Description: Creates an opportunity.

Help Text: Create an opportunity and/or add opportunity participants.

Initial Step: (None)

**Process Phases**

- `CreateY1`
- `AddY2`

**Data Providers**

Name	Action
<code>curTexts</code>	UsingDataProvider

**Process Storage Variables**

Name	Data Provider	Value
<code>curFUID</code>	none	0
<code>curY1uid</code>	Input Message	<code>\$.get('uid')</code>
<code>curRepName</code>	none	<code>u8.session.identity.repName</code>
<code>curY2Count</code>	none	0
<code>repMetric</code>	<code>curTexts</code>	<code>\$(('TXT_Process_TextGroup_0'))</code>

**Process Metrics**

View	Data Provider	Value	Help Text
<code>Text</code>	Process Storage	<code>\$.get('repMetric')+\$.get('curRepName')</code>	The name of the current rep.

Since your process contains phases you do not need to specify an **Initial Step**.

- The **Header** drop-down list lists all info area-independent special headers.
- For the process metrics define a custom **Data Provider** `curTexts` loading the text group containing the multilingual texts used in the process.
  - Enter a **Name**.
  - Define an **Action** using the built-in external data provider `UsingDataProvider`. For further details, see [UsingDataProvider](#) on page 569.

**Action for DataProvider**

Action Template: `UsingDataProvider`

A dataprovider that can be used to load resources like textgroups.

Pass arguments as an object

Input Arguments

Source	Value	Name
Constant Value	<code>Process_TextGroup</code>	<code>texts</code>

- Enter the name of the desired text group as argument `texts`.
  - Click **Apply**.
- Define following process storage variables:
    - `curFUID`: The uid of the company record for which the opportunity is created. If the process is started from the context menu of an existing opportunity the company uid is not needed and not passed to the process.

If the process is started from the application menu, the uid comes from the initial step where the user selects a company. For further details, see [Step SelectFIForY1](#).

This is handled automatically by the process, you do not need to define conditions in this case.

- `curY1Uid`: The uid of the opportunity record.

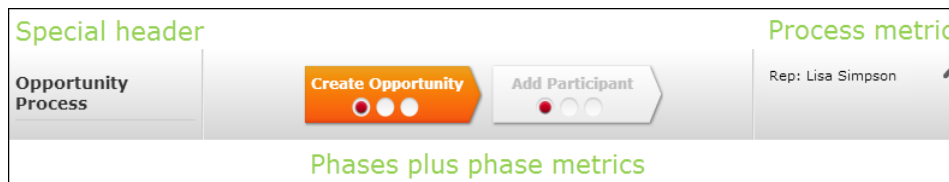
If the process is started from the context menu, the uid comes from the current opportunity, therefore **Data Provider** = Input Message (`$.get('uid')`).

If the process is started from the application menu, the uid comes from the step where the user creates a new opportunity. For further details, see [Step CreateY1](#).

- `curY2Count`: This variable is calculated by an evaluation step and used for the ProcessAsk page. For further details, see [Step EvalY2Count](#) and [Step ProcessAskY1](#). Therefore initial value = 0. Since `curY2Count` is accessed at different points of the process, it must be defined in the process storage.
- `repMetric`: `$( 'TXT_Process_TextGroup_0' )` gets text number 0 of text group `Process_TextGroup`. Syntax, . For further details, see [Referencing Texts in JavaScript](#) on page 178. The text group has been loaded by the custom **Data Provider** `curTexts`.
- `curRepName`: The name of the current rep. Available functions for identifying the current rep. For further details, see [u8.session.identity](#).

5. To display the text "Rep: <current rep>" as a process metric, a multilingual label and a variable value are combined as `$.get('repMetric')+$$.get('curRepName')`. In an English-only environment you could just define "Rep: "+`$.get('curRepName')` instead.

`repMetric` has been defined as a process storage variable because texts from different data providers can not be combined in one value definition.



Alternatively, you could define two text metrics, one getting the label directly from data provider `curTexts` and the second getting the rep name from the process storage.

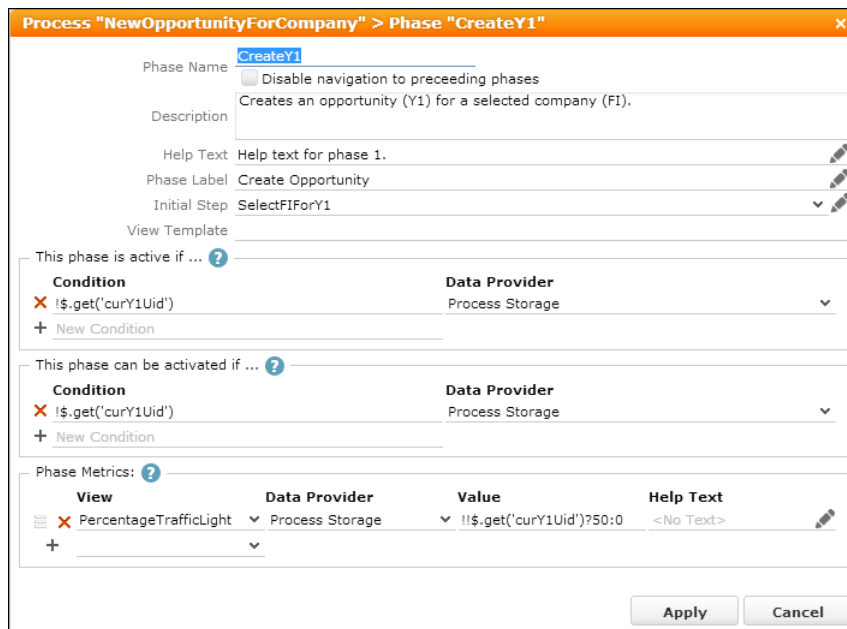
View	Data Provider	Value
Text	curTexts	\$( 'TXT_Process_TextGroup_0' )
Text	Process Storage	\$.get('curRepName')

In this case the texts are displayed in two lines.

## Phase CreateY1

Define the first phase (selecting a company and creating an opportunity):

1. Under **Process Phases** click + (Add).



Process "NewOpportunityForCompany" > Phase "CreateY1"

Phase Name: CreateY1

Disable navigation to preceding phases

Description: Creates an opportunity (Y1) for a selected company (FI).

Help Text: Help text for phase 1.

Phase Label: Create Opportunity

Initial Step: SelectFIForY1

View Template

This phase is active if ... ?

Condition	Data Provider
✗ !\$.get('curY1Uid')	Process Storage
+ New Condition	

This phase can be activated if ... ?

Condition	Data Provider
✗ !\$.get('curY1Uid')	Process Storage
+ New Condition	

Phase Metrics: ?

View	Data Provider	Value	Help Text
✗ PercentageTrafficLight	Process Storage	!!\$.get('curY1Uid')?50:0	<No Text>
+ New Metric			

Apply Cancel

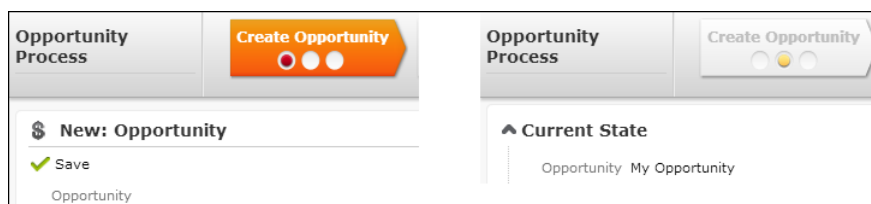
2. Enter a **Phase Name**.
3. Enter a **Phase Label**. This is displayed on the phase arrow.
4. Select the phase's **Initial Step**. For further details, see >> [Step SelectFIForY1](#).

**Note:** You need to create the steps before you can select them.

Once you have selected a step you can click  (**Edit**) to edit the step.

5. Define the conditions for the phase to be active: `!$.get('curY1Uid')`  
Phase `CreateY1` is only active/can only be activated if **no** opportunity exists, i.e. if called from the context menu of an opportunity, the process automatically starts with phase `AddY2`.
6. Define a phase metric using the **PercentageTrafficLight View** with **Value**  
`!!$.get('curY1Uid')?50:0`

Once an opportunity `uid` is present in the process storage, the traffic light changes to yellow.



7. Click **Apply**.
8. You need to (re-)save the process, otherwise the phase definition is lost.

## Phase AddY2

Define the second phase (adding opportunity participants):

The screenshot shows the configuration for Phase "AddY2" within the process "NewOpportunityForCompany". The interface includes the following fields and sections:

- Phase Name:** AddY2
- Description:** Adds opportunity participants (Y2) to the opportunity.
- Help Text:** Help text for Phase 2.
- Phase Label:** Add Participant
- Initial Step:** EvalY2Count
- View Template:** (Link)
- This phase is active if ...**
  - Condition:** `!!$.get('curY1Uid')`
  - Data Provider:** Process Storage
- This phase can be activated if ...**
  - Condition:** `!!$.get('curY1Uid')`
  - Data Provider:** Process Storage
- Phase Metrics:**

View	Data Provider	Value	Help Text
PercentageTrafficLight	Process Storage	<code>\$.get('curY2Count')==0?0:100</code>	<No Text>

Buttons for "Apply" and "Cancel" are located at the bottom right of the configuration window.

- **Initial Step:** For further details, see [Step EvalY2Count](#)

- **Condition:** `!!$.get('curY1Uid')`

Phase `AddY2` is active/can be activated if an opportunity exists.

- **Phase metric:** `$.get('curY2Count')==0?0:100`

The traffic light changes to green as soon as at least one opportunity participant has been added (`curY2Count > 0`).



Each time the phase is activated, the process storage variable `curY2Count` is (re-)calculated by the phase's initial step `EvalY2Count`.

## Step SelectFIForY1

1. Create a step called `SelectFIForY1` allowing the user to select (or create) a company for which to create the opportunity.

The screenshot shows the configuration for a process step named 'SelectFIForY1'. The description is 'Selects the company for which the opportunity is created.' The 'Data Providers' section is collapsed. The 'Action' section is expanded, showing the 'Action Template' set to 'ExecuteSelectRecordSubStep'. Below this, there are checkboxes for 'Pass arguments as an object' (checked) and 'End process with this action' (unchecked). The 'Input Arguments' section contains one argument: 'Constant Value' with a value of 'SelectCompany->Return' and a name of 'name'. The 'ActionTemplate Output Arguments' section shows an output argument 'uid' with the description 'The uid of the selected record.' The 'Validation' section is empty. The 'Output Arguments' section contains one argument: 'uid' with a target of 'Process Storage' and a name of 'curFIUid'. The 'Dispatching' section contains one condition: 'Constant Value' with a condition of 'true' and a next step of 'CreateY1'.

2. Define the action: In this case a sub-step is called (`ExecuteSelectRecordSubStep`) that uses the existing step `SelectCompany->Return`.

`ExecuteSelectRecordSubStep` returns the `uid` of the selected record. For further details, see [ExecuteSelectRecordSubStep](#) on page 581.

3. To pass the `uid` of the selected FI record to the next step you need to define an output argument for the step: The `uid` is written to the process storage variable `curFIUid`. This is necessary since the process can be started with and without context, i.e. parent record. If the process is started from an existing opportunity, `curFIUid` comes from the Input Message. For further details, see [Process NewOpportunityForCompany](#).
4. Create the **Next Step**, see [Step CreateY1](#). Since there is only one possible next step creating the opportunity, enter "true" as dispatching "condition".

## Step CreateY1

1. Create a step called `CreateY1` allowing the user to create a new opportunity.

**Process Step 'CreateY1'**

Save Delete Usage Graph Processes...

Step: CreateY1  
Description: Creates an opportunity for the selected company.  
Help Text:

**Data Providers**

Name	Action
curFIRep	ReadRecordDataProvider

**Action**

Action Template: ExpandNew  
Creates a new record in expanded view.  
 Pass arguments as an object  
 End process with this action

Input Arguments

Source	Value	Name
Constant Value	Y1	infoAreaId
Process Storage	\$.get('curFIUID')	link
Constant Value	none	history
Constant Value	RepID	defaultFields
curFIRep	\$.get('RepID')	defaultValues

Validation

Source	Condition	Error Text
+ New Condition		

Output Arguments

Value	Target	Name
step.get('uid')	Process Storage	curY1Uid

Dispatching

Source	Condition	Next Step	Arguments
Step	\$.get('command')==\'Save\'	ProcessAskY1	

2. Select the action (`ExpandNew`) and define its input arguments:
  - `infoAreaID: Y1`
  - `link`: Since the process can be started with and without context, the opportunity's parent (FI) must come from the process storage: `$.get('curFIUID')`
3. To read the `RepID` from FI and write it to `RepID` in Y1 you need to define a custom data provider `curFIRep` for this step:
  - a. Enter a **Name**.
  - b. Define an **Action** using the built-in external data provider `ReadRecordDataProvider`. For further details, see [ReadRecordDataProvider](#) on page 568.

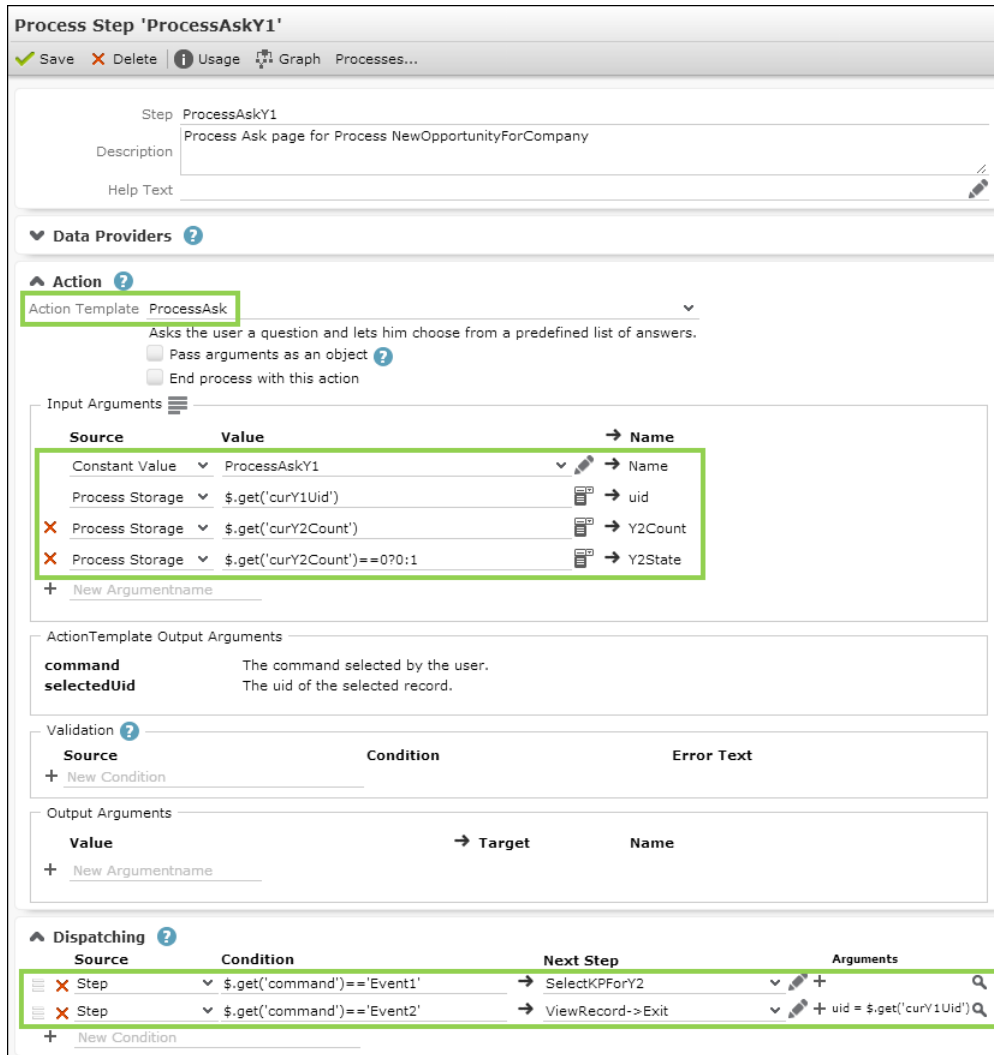
Source	Value	Name
Process Storage	\$.get('curFIUID')	uid / uid
Constant Value	RepID	fields

- c. Get the `uid` of the record to read from the process storage: `$.get('curFIUID')`
  - d. Enter the IDs or XML names of the fields to be read.
  - e. Click **Apply**.
4. Define the input arguments `defaultValues` and `defaultFields`. For further details, see [defaultFields/defaultValues](#).
    - Get `defaultValues` from the custom data provider `curFIRep: $.get('RepID')`
    - Enter the IDs or XML names of the target fields as `defaultFields`
  5. To pass the `uid` of the created Y1 record to the next step you need to define an output argument for the step: The `uid` is written to the process storage variable `curY1Uid`. This is necessary since the process can be started with and without context, i.e. parent record. If the process is started from an existing opportunity, `curY1Uid` comes from the Input Message. For further details, see [Process NewOpportunityForCompany](#).
  6. Define the **Dispatching**:
    - a. Enter the condition: `$.get('command')== 'Save'`  
Saving the record navigates to the next step.
    - b. Select "Step" as data provider (**Source**).
    - c. Create the **Next Step**, see [Step ProcessAskY1](#).



## Step ProcessAskY1

1. Create a step called `ProcessAskY1` displaying the previously created opportunity and ask the user to create opportunity participants. For further details, see [ProcessAsk](#) on page 579. As soon as at least one participant is added, the user can exit the process.



2. Select the action (`ProcessAsk`) and define its input arguments:

- The `name` of the Process Ask configuration `ProcessAskY1`. For further details, see [Process Ask Configuration ProcessAskY1](#)

Get the necessary parameters for the Process Ask page from the process storage:

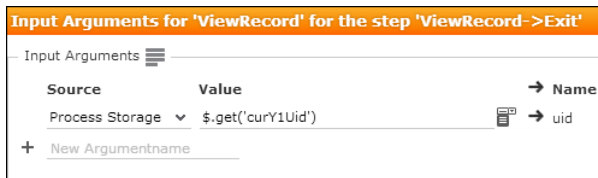
- the opportunity record's `uid` for the Expand view
- the current participants count (`Y2Count`) for the text
- `Y2State` evaluating if the minimum number of participants has been reached - only then the process can be completed.

3. Define the **Dispatching** depending on the buttons defined in the Process Ask configuration:

- Event 1: The user clicks the **Add Participant** button. This calls `CreateY2` as **Next Step**.

- Event 2: The user clicks the **Show Opportunity** button. This calls `ViewRecord->Exit` and ends the process. For further details, see [ViewRecord->Exit](#).

Since a `ProcessAsk` step does not provide `uid` as output argument you need to get the `uid` for `ViewRecord->Exit` from the process storage:



## Step SelectKPForY2

1. Create a step called `SelectKPForY2` allowing the user to select (or create) a person to be added as opportunity participant.

The screenshot shows the configuration for a process step named 'SelectKPForY2'. The description is 'Selects the person (KP) which is created as opportunity participant (Y2)'. The 'Action' section is expanded, showing the 'ExecuteSelectRecordSubStep' template. The 'Input Arguments' table has one entry: 'Constant Value' with 'SelectPerson->Return' as the value and 'name' as the name. The 'ActionTemplate Output Arguments' section shows 'uid' with the description 'The uid of the selected record.'. The 'Validation' section is empty. The 'Output Arguments' section is empty. The 'Dispatching' section has one entry: 'Constant Value' with 'true' as the condition, 'CreateY2' as the next step, and arguments 'link = \$.get('uid')' and 'link2 = \$.get('curY1Uid')'.

2. Define the action: Analogous to `SelectFIForY1` a sub-step is called (`ExecuteSelectRecordSubStep`) that uses the existing step `SelectPerson->Return`. It returns the `uid` of the selected person record.
3. Define the **Dispatching**: Since there is only one possible next step (creating the opportunity participant), enter "true".
4. Create the next step, see [Step CreateY2](#). Pass the `uid` of the selected person and the opportunity (parent of opportunity participant) to `CreateY2`:
  - a. Click **q** (**Edit step arguments**).

The screenshot shows the 'Input Arguments for ExpandNew' dialog for the step 'CreateY2'. The 'Input Arguments' section has two entries: 'Step' with 'uid' as the value and 'link' as the name, and 'Process Storage' with '\$.get('curY1Uid')' as the value and 'link2' as the name.

You get the person record from the current step.

You need to get the opportunity from the process storage because the process can be started with and without context.

- b. Click **Apply**.

5. Save the step definition.

## Step CreateY2

1. Create a step called `CreateY2` allowing the user to create a new opportunity participant.

**Process Step 'CreateY2'**

Save Delete Usage Graph Processes...

Step: CreateY2  
 Description: Creates an opportunity participant (Y2). On save the number of existing Y2 records is (re)calculated.  
 Help Text:

**Data Providers**

**Action**  
 Action Template: ExpandNew  
 Creates a new record in expanded view.  
 Pass arguments as an object  
 End process with this action

**Input Arguments**

Source	Value	Name
Constant Value	Y2	infoAreaId
Input Message	as is	link
Input Message	as is	link2

**Validation**

Source	Condition	Error Text
+ New Condition		

**Output Arguments**

Value	Target	Name
+ New Argumentname		

**Dispatching**

Source	Condition	Next Step	Arguments
Step	\$.get('command')==\'Save\'	EvalY2Count	
+ New Condition			

2. Select the action (`ExpandNew`) and define its input arguments:

- `infoAreaID`: Y2
- `link`: the link to the person
- `link2`: the link to the opportunity

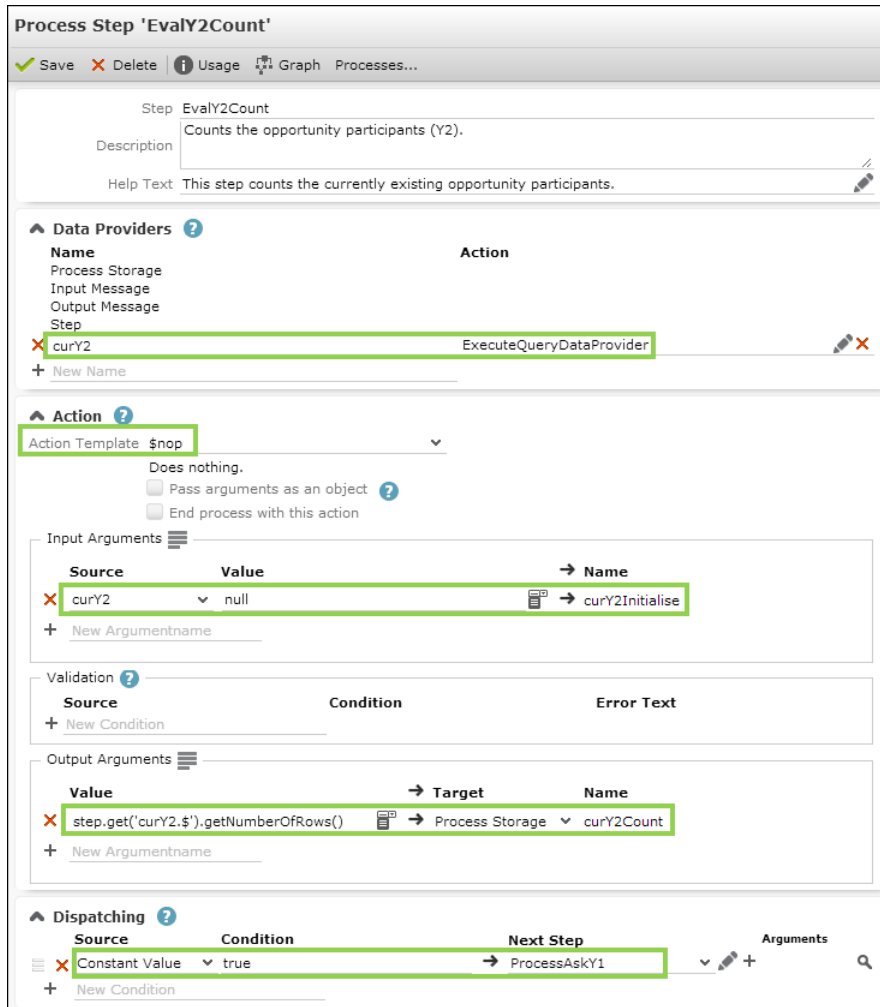
Since `link` and `link2` come from the dispatching definition of the previous step, you only need to select "Input Message" as **Source** and select "as is" from the **Value** drop-down list.

3. Define the **Dispatching**:

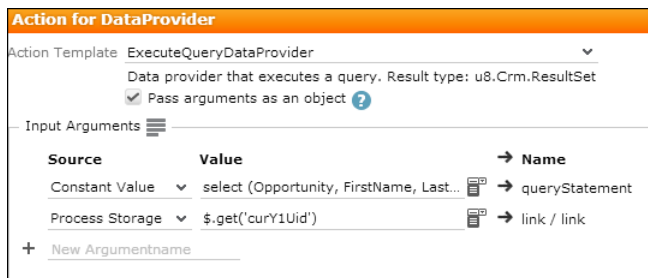
- Enter the condition: `$.get('command')==\'Save\'`  
 Saving the record navigates to the next step.
- Select "Step" as data provider (**Source**).
- Create the **Next Step**, see [Step EvalY2Count](#).

## Step EvalY2Count

1. Create a step called `EvalY2Count` to calculate the current number of opportunity participants. As soon as at least one participant is added, the user can exit the process.



2. Select `$nop` as **Action Template**: `EvalY2Count` is an evaluation step executed in the background. For further details, see [\\$nop](#) on page 581. It just recalculates the current participant count and leads back to the Process Ask page which is updated with the result.
3. To get the current participants count you need to run a query. Define a custom data provider `curY2` using the built-in external data provider `ExecuteQueryDataProvider`. For further details, see [ExecuteQueryDataProvider](#) on page 565.



- Enter the `queryStatement:select (Opportunity, FirstName, LastName) from (Y2)`



- Enter the parent `link` for the query getting the opportunity record from the process storage: `$.get('curY1Uid')`
4. You need to explicitly initialize this data provider: Add an input argument for the `$nop` action that uses `curY2` as source. You can enter any name as argument name (`curY2Initialize` in this example).
  5. To pass the current participants count (i.e. the number of result rows) to the next step you need to define an output argument writing the result to the process storage variable `curY2Count`: Define `step.get('curY2.$').getNumberOfRows()` as **Value**. This returns the number of rows in the current query result from data provider `curY2`. For more functions, see [ExecuteQueryDataProvider](#) on page 565.
  6. Define the **Dispatching**: This invisible evaluation step always returns to the Process Ask page, see [Step ProcessAskY1](#).

## Process Ask Configuration ProcessAskY1

The Process Ask configuration defines the items to be displayed on the Process Ask page displayed by step `ProcessAskY1`. For further details, see [Step ProcessAskY1](#).

The screenshot shows the configuration for 'ProcessAskY1'. At the top, there's a header with 'Opportunity Process' and 'Rep: Lisa Simpson'. Below this are two buttons: 'Create Opportunity' and 'Add Participants'. The main content area is divided into two sections: 'Current State' and 'Add Participants'. The 'Current State' section displays a table with columns for Opportunity, Opp. Type, Rep ID, Status, Completion Da..., Level, Priority, and Completion. The 'Add Participants' section lists two participants: 'Andrea Gabrieli (Gabrieli) as Contact Person' and 'John Doe (update software AG) as External Consultant'. Below the list, there is a message 'You have 2 participant(s) linked to your opportunity.' and two buttons: 'Add Participant' and 'Show Opportunity'.

1. Create a Process Ask configuration called `ProcessAskY1`.

Process Ask 'ProcessAskY1'

Save Delete

Description

Modify the items that will be displayed on the process ask page:

Type	Event No.	DisplayText	Action	TableCaptionName	CRM Process	Max. rows	Parameters
Caption		Current State			all	Q	Parameters
Expand View					all	Q	{expandName:'Y1.Process'}
Separator					all	Q	
Caption		Add Participants			all	Q	Parameters
Linked Records	0	Display text		Y2.Process	all	Q 5	{linkName:'\$Children[Y2].showGridHeader:fals...
Text		You have {url:Y2Count} par...			all	Q	{image:'CAT_ZF_{url:Y2State}'}
Button	1	Add Participant			all	Q	Parameters
If							{url:Y2Count}>0
Button	2	Show Opportunity			all	Q	Parameters
End If							

- Add the desired elements (Caption, Expand View, Text, Separators, Buttons). For further details, see [Process Ask Configuration](#) on page 582.

You can pass most of the parameters as URL-parameters since they are defined as input arguments of `ProcessAsk` action in step [Step ProcessAskY1](#).

Action ?

Action Template `ProcessAsk`

Asks the user a question and lets him choose from a predefined list of answers.

Pass arguments as an object ?

End process with this action

Input Arguments

Source	Value	Name
Constant Value	ProcessAskY1	Name
Process Storage	\$.get('curY1Uid')	uid
Process Storage	\$.get('curY2Count')	Y2Count
Process Storage	\$.get('curY2Count')==0?0:1	Y2State

- Expand View:** `{expandName: 'Y1.Process'}`. You do not need to specify `uid` here, since it is automatically passed from the `ProcessAsk` action. For further details, see [Expand View](#) on page 585.
- Linked Records:** Specify a **TableCaptionName** to determine from which info area the linked records are read. Specify the link relation to be used (`$Children`) under **Parameters**. For further details, see [Linked Records](#) on page 587.
- Text:**
  - The **DisplayText** uses the `Y2Count` argument: "You have `{url:Y2Count}` participant(s) linked to your opportunity". For further details, see [Text](#) on page 590.
  - The image (specified under **Parameters**) uses the `Y2State` (0 or more participants) argument: `{image: 'CAT_ZF_{url:Y2State}'}`  
(The images used are called `CAT_ZF_0` and `CAT_ZF_1`.)
- Button:** Here you define the buttons triggering the events for the dispatching of step `ProcessAskY1`. For further details, see [Button](#) on page 583.

The second button (**Show Opportunity**) is only displayed, if at least one opportunity participant has been added: `{url: Y2Count }>0`. For further details, see [If...End If](#) on page 586.

## Menu Action ID\_NET\_Process\_NewOpportunityForCompany

1. To call your process (without context) create an application menu action `ID_NET_Process_NewOpportunityForCompany` in `ID_NET_PROCESSES` (i.e. the **Sales** menu). For further details, see [Configuring a Menu](#) on page 292.

**Menu ID\_NET\_Process\_NewOpportunityForCompany**

✓ Save ✗ Delete

Text: Opportunity Process

Configuration: UserConfig\_SUSANNA\_Sub0

CRM Processes: all

Info Area:

QuickLink Context:

Image: <No Image>

Visible

Action: Process

---

**Action for Menu ID\_NET\_Process\_NewOpportunityForCompany**

Action Template: Process

Executes a process.

Pass arguments as an object

Input Arguments:

! name	NewOpportunityForCompany
+ New Argumentname	

Output Arguments:

+ New Argumentname	
--------------------	--

Apply Cancel

2. Select `Process` as **Action Template**.
3. Enter the name of the process as input argument `name`.

## Context Menu Action A\_StartProcessNewOpportunityForCompany

1. To call your process (with context) create a context menu action `A_StartProcess-NewOpportunityForCompany` in `M_Opportunity`.

**Menu A\_StartProcessNewOpportunityForCompany**

✓ Save ✗ Delete

Text: Start Opportunity Process

Configuration: UserConfig\_SUSANNA\_Sub0

CRM Processes: all

Info Area:

QuickLink Context:

Image: <No Image>

Visible

Action: Process

---

**Action for Menu A\_StartProcessNewOpportunityForCompany**

Action Template: Process

Executes a process.

Pass arguments as an object

Input Arguments:

! name	NewOpportunityForCompany
uid	Record
link	Linked

2. Select `Process` as **Action Template**.

3. Enter the name of the process as input argument `name`.
4. Define `uid` as `Record` (i.e. the opportunity record to start the process for). Define `link` as `Linked` (i.e. the parent company of the opportunity).

This writes these two records to the input message of the process.

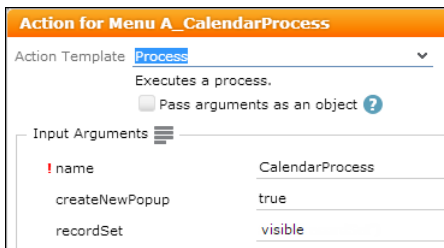
If started from the context menu, the process automatically jumps to the second phase because of the conditions you defined for that phase (`!!$.get('curY1Uid')`). For further details, see [Phase AddY2](#).

## Passing Multiple Records to a Process Ask page

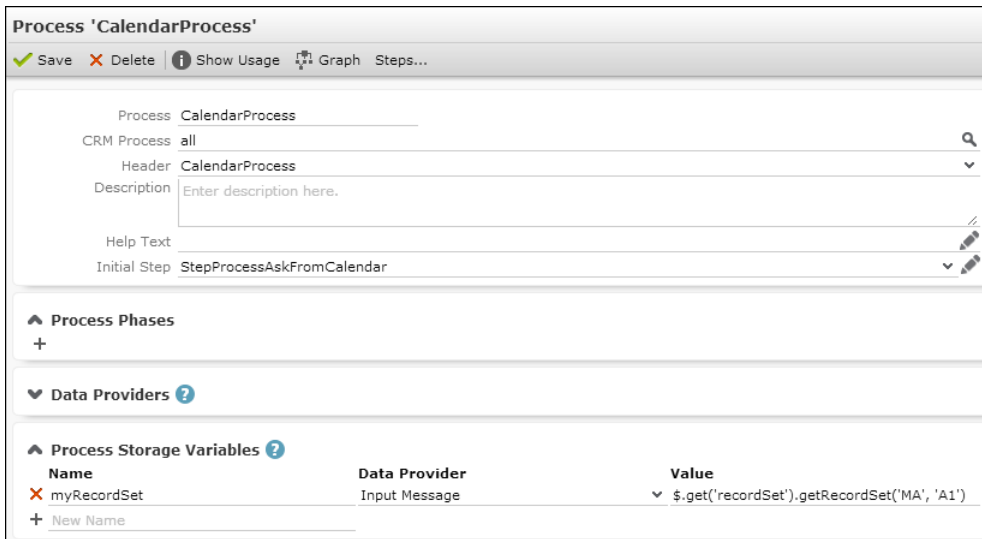
Learn about passing multiple records to a Process Ask page.

A process launched from the calendar's context menu `M_CalNew`.

Action definition:



Process definition (passing the records to a process storage variable and filtering for MA and A1 records):



Step definition (getting the `recordSet` argument from the process storage):

**Process Step 'StepProcessAskFromCalendar'**

Save Delete Usage Graph Processes...

Step: StepProcessAskFromCalendar

Description: Enter description here.

Help Text:

**Data Providers**

**Action**

Action Template: ProcessAsk

Asks the user a question and lets him choose from a predefined list of answers.

Pass arguments as an object

End process with this action

Input Arguments

Source	Value	Name
Constant Value	CalendarProcessAsk	Name
Process Storage	\$.get('myRecordSet')	recordSet

Process Ask configuration with Records item (no parameter specified, i.e. RECORDLISTS):

**Process Ask 'CalendarProcessAsk'**

Save Delete

Description: Enter description here.

Modify the items that will be displayed on the process ask page:

Type	Event No.	DisplayText
Caption		Record List
Records	1	My Appointments

The process is called from the Calendar. All visible records are written to the Input Message as `recordSet`. The process storage variable filters for MA and A1. Since no `uid` or `uid1-3` arguments were specified, only the records from `recordSet` are displayed:

**Calendar Process**

**My Calendar Process**

**Record List**

**My Appointments**

- 14.01.2014 00:00: Visit for Krist OY
- 20.01.2014 00:00: Telephone for Mr. Girolamo Frescobaldi (Merulo S.p.A.)
- Task due 17.01.2014

## Forms

Create custom forms as per your need.

You can create custom forms to be used for various purposes. Example - as start page or within a process.

On the CRM.Designer main page, click **Forms** to display all existing forms and to access the Form Designer.



The following topics provide a short overview of forms and the Form Designer.

For more information, see the article "Form Designer" at <https://support.aurea.com>.

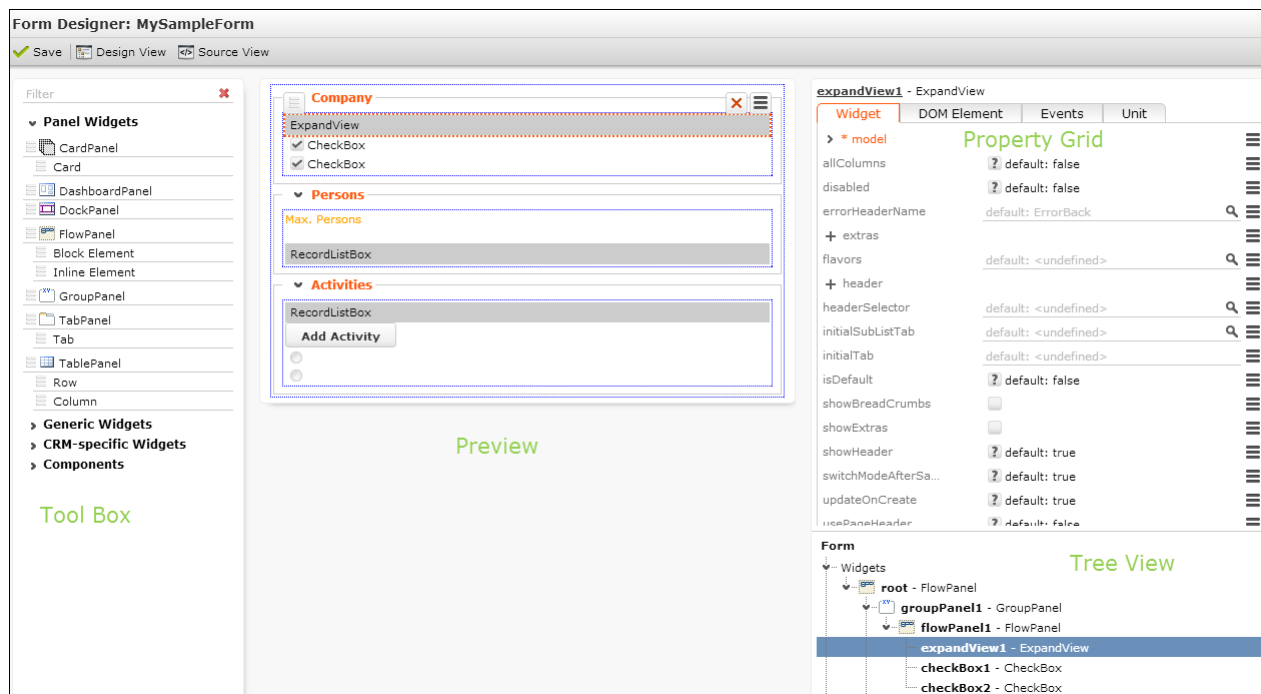
**Important:** Please note that Aurea CRM Legacy Forms are deprecated and no longer supported by Aurea in version 8 and above. Forms like LastQueryResult and related menu items are not supported in versions after U7.

You are requested to move to the new Forms described in this chapter. The Legacy Forms capability in the CRM.Designer is also removed soon.

## Form Designer

An overview of form designer.

To get started designing forms using the Form Designer, read [HOWTO Use the Form Designer](#) to get a quick overview in terms of structure, handling, features and other interesting information.



The Form Designer consists of the following areas:

- **Tool Box:** Lists all available widgets and components. For further details, see [Tool Box](#) on page 623. Enter a filter string in the field above the list to search for an element.
- **Preview:** Displays the current form. Use drag&drop to add elements to the form. For further details, see [Arranging Elements on a Form](#) on page 624.
- **Property Grid:** Lists all supported settings for the selected element. For further details, see [Property Grid](#) on page 625.
- **Tree View:** Displays the elements of the current form hierarchically. Select an element in the Tree View to edit its properties.

You can resize the areas of the Form Designer by dragging the border with the mouse.

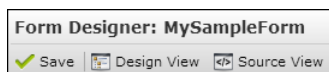
For each element in the Tool Box and the Property Grid you can display an explanatory tooltip on mouse-over. The tooltip also contains links to the respective page in the Aurea CRM.Web SDK Client Reference located at <https://support.aurea.com/sdk/reference/>. In addition, miscellaneous hints and examples can be found here. For further details, see [Hints](#) on page 628.

In the code, elements are referenced by the name - "Button". The elements you add to your form are named automatically as "Button 1", "Button 2", ..... "Button x".



The name of the currently selected element is displayed as a link above the Property Grid. Click on the link to change the name.

Use the **Source View/Design View** buttons to toggle between code and WYSIWYG view.










## Tool Box

Know about the toolbox used to create forms.

The Tool Box contains the following items:

- **Panel Widgets:** Widgets used for structuring a form. The following panel widgets are available:

	CardPanel	Contains a stack of cards where exactly one is visible.
	DashboardPanel	Displays its children in columns or as tabs. For further details, see <a href="#">Dashboards</a> on page 632.

	DockPanel	Lays out its child elements in a dock layout, arranging them either horizontally or vertically. Dock panels are not dynamically resized according to their content, therefore make sure you specify at least the <code>height</code> property.
	FlowPanel	Simply appends one child element after the other in a <code>&lt;DIV&gt;</code> element.
	GroupPanel	Encloses a widget with a group element.
	TabPanel	Displays a tabbed widget, allowing users to switch between the tabs.
	TablePanel	Lays out its child elements in an HTML <code>&lt;table&gt;</code> element.

- **Generic Widgets:** Widgets that can be used independent of records and other CRM-specific data. For Example - field schemata or entries of the CRM.Designer database, `Button`, `TextInput`, `Image`.

Generic widgets are displayed in the preview area exactly as they are displayed in the Aurea CRM.Web user interface.

For a list of available generic widgets, see the article “Form Designer Widgets” at <https://support.aurea.com>.

- **CRM-specific Widgets:** Widgets that are directly related to CRM data. For example, the `ExpandView` widget can only be displayed based on a record.

For CRM-specific widgets no WYSIWYG preview is available.

For a list of available CRM-specific widgets, see the article “Form Designer Widgets” at <https://support.aurea.com>.

- **Components:** Components are not assigned to a specific widget but to the entire form. They are used to access additional data sources.

For a list of available components, see the article “Form Designer Widgets” at <https://support.aurea.com>.

## Arranging Elements on a Form

Here you can learn how to arrange elements on a form.

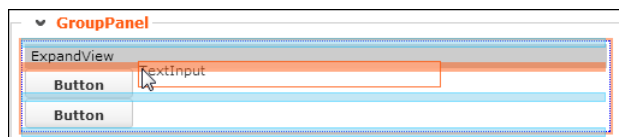
The Preview area displays the current form.



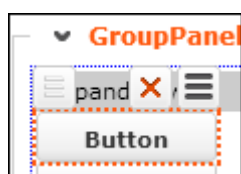
To edit the form:




1. Use drag&drop to add elements from the Tool Box to the form.


When dragging the element over the form, all valid locations for the elements are displayed in light blue. The border of the current target panel is highlighted in orange. The location where the element would be positioned when releasing it, is displayed in orange.



2. Select an element of the form (in the Tree View) to display its controls:



- Click  (**Drag&Drop**) to move the element to another position.
- Click  (**Delete**) to remove the element from the form.
- Click  (**Context Menu**) to display the elements context menu, containing the actions **Cut** and **Copy**.

If a widget has been cut or copied, any empty container area you select in the form displays a  (**Paste**) button, allowing you to paste the copied/cut element at that position.



To create containers to paste elements to, you need to add child panel widgets to your form. For Example - `Block Element` or `Row` and `Column`

3. Define all necessary settings for the widgets on your form. For more details, see [Property Grid](#) on page 625.

## Property Grid

The Property Grid lists all supported parameters for the selected element.

The meaning and possible values of each parameter is described in the tooltip. For CRM-specific widgets you can also refer to the Aurea CRM.Web SDK Client Reference located at <https://support.aurea.com/sdk/reference/> and to the corresponding action templates and their input arguments. For further details, see [Action Templates](#) on page 350.

For text fields of the Property Grid you can click  (**Select text**) and access the multilingual CRM.Designer texts (e.g. for labels). For further details, see [Texts](#) on page 176. In the Property Grid CRM.Designer texts are displayed as `{TXT_<textgroup>_<number>}`. The Preview displays the text in the default language. You can combine CRM.Designer texts and plain text. Example - to add punctuation: `{TXT_<textgroup>_<number>} : {TXT_<textgroup>_<number>}`.

For each parameter the context menu offers options to reset the value (**Reset to default**, **Set to <null>**).

---

**Note:** Opening a property group by clicking on the + resets any default values inside the group to undefined. If you want to undo your changes, make sure you select **Reset to default** (from the context menu) for the properties group. Otherwise the widget may no longer work.

---

**Note:** The same applies to unused bindings: Instead of clearing the value, select **Remove binding** from the context menu.

---

Click **Apply** to see your changes in the Preview pane. The settings of the DOM Element tab are immediately applied to the Preview as well.

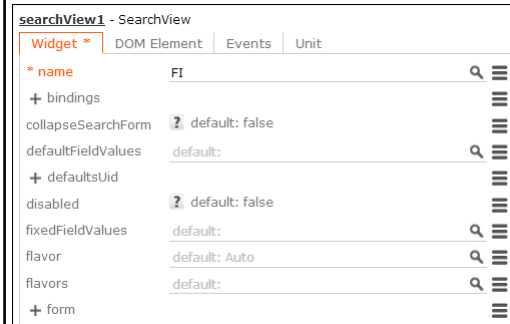
## Widget tab

The **Widget** tab contains all supported parameters controlling the behavior of the widget.

Mandatory parameters are highlighted and tagged with \*.

For each parameter the default value is displayed.

Use the context menu option **Set from binding** to enable data binding. For further details, see [Data Binding](#) on page 628.

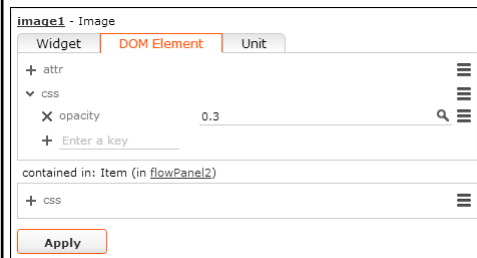


## DOM Element tab

The **DOM Element** tab contains options to change the visualization of the widget. You can define:

- Attributes (**attr**)
- CSS parameters (**css**)

All available DOM Elements are supported.

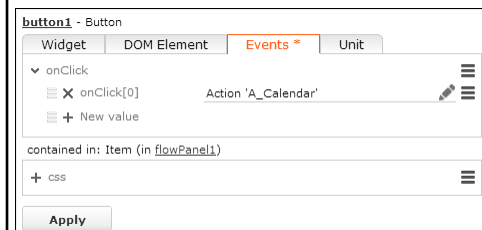


## Events tab

Use the **Events** tab to define actions for predefined events. The list of predefined events is fixed for each widget and cannot be changed (i.e. for the `Button` widget only an `onClick` event is available).

To define an action:

1. Define a new value.
2. Click (**Edit**).
3. You can:
  - **Execute an action:** Enter the name of a (menu) action as defined in CRM.Designer.
  - **Execute a javascript function:** Define your function in the code box.
4. Click **OK**.
5. Use the (**Drag&Drop**) buttons to reorder values.



## Unit tab

Use the **Unit** tab to assign a CRM process to each widget. For further details, see [Assigning CRM Processes](#).

## Data Binding

Data binding is used to define the interaction of widgets in forms.

For a detailed description, see the article “Data Binding in Forms” at <https://support.aurea.com>.

For examples, see the Examples section in the article “Forms” at <https://support.aurea.com>.

## Hints

Here are the list of supported widgets that can be used on a form.

### Calendar Widget

Depending on the context you may want to hide the **New Calendar Entry** button when using the `calendar` widget on a form:

- Set the `header > showNewButton` property to `false`.

To specify another menu to be displayed when clicking the **New Calendar Entry** button:

- Use the `menuItem` property to specify another than the default menu `M_CalNew`.

### Getting a Tree's Root Record in a Form

In order to get the uid of the tree's root record in a form you have to use a binding where `Source = "#rootUid"`. Example in `UPDATE_DEFAULT: RecordHistory` widget.

### Grouping Radio Buttons/Toggle Buttons on a Form

Use the `groupName` and `groupValue` properties to manage the value of a group of radio buttons or toggle buttons. Radio/toggle buttons with the same `groupName` are considered as grouped. This means activating a button from a group deactivates the other buttons in the group. Use the `groupValue` property to pass the value of currently selected button to some other widget.

## Example:

```
"radioButton1": {
  "type": "RadioButton",
  "options": { "groupName": "groupradiol", "value": "0" },
  "binding": [
    {
      "sourcePath": [ "#radiol" ],
      "targetPath": [ "@groupValue" ]
    }
  ]
},
"radioButton2": {
  "type": "RadioButton",
  "options": { "groupName": "groupradiol", "value": "1" },
  "binding": [
    {
      "sourcePath": [ "#radiol" ],
      "targetPath": [ "@groupValue" ]
    }
  ]
}
```

Example in UPDATE\_DEFAULT: The Daily/Weekly radio buttons on the `Recurring` form.

## List Widgets

All properties used to configure the functionality and appearance (header, default action, menu, etc.) of the various list widgets (`SearchView`, `RecordListBox`, `QueryResultPresenter`, etc.) can be found in the `grid` property group.

### ListView Widget

When defining forms with "hierarchical" lists (e.g. to display records from FI, KP and MA) you can hide `ListView` widgets for which no parent record is selected:

1. Check the `hideIfNoLink` property.
2. Also check `autoLoad`. Otherwise the `ListView` widget's header would still be displayed.

`hideIfNoLink` is also available for the `RecordListBox` widget.

### RecordListBox Widget

When using a `RecordListBox` widget in a `DashboardPanel` and wanting to hide the `DashboardPanelItem` in case the `RecordListBox` contains no results:

1. Set the `hideItemsIfEmpty` property of the `DashboardPanel` widget to `true`. For further details, see [Configuring Dashboards](#).
2. Set the `hideIfEmpty` property of the `RecordListBox` to `true`.

### RecordTreeView

To use a `RecordTreeView` widget in a form:

- make sure to specify a root record for the tree by setting a valid `uid`,
- set a height under `css`, otherwise the widget is not displayed.

## SearchView Widget

To hide the **With me** button:

- Set the `form > showOwnItems` property to `false`.

---

**Note:** To set the `includeConditionalRights` parameter, when you are using `SearchView` widget on a **Form**, go to `SearchView -> resultModel` to find and set the parameter.

---

To manage the search history for two `SearchView` widgets which use the same `Search&List` configuration:

- Use the `form > historyNamespace` property. This saves separate search histories for each widget.

You need to define a unique value for `historyNamespace` for each `SearchView` widget if you use two searches with the same `Search&List` configuration but e.g. different fixed filters on one form. Otherwise, the second filter is never applied as the searches share the same cache.

## Defining a Header/Title for a Form

Rules for defining header for a form.

To define headers/titles for your forms:

- Use the `HeaderName` parameter in the `ShowForm` action, to display any special header as the form's header. For further details, see [HeaderName](#).

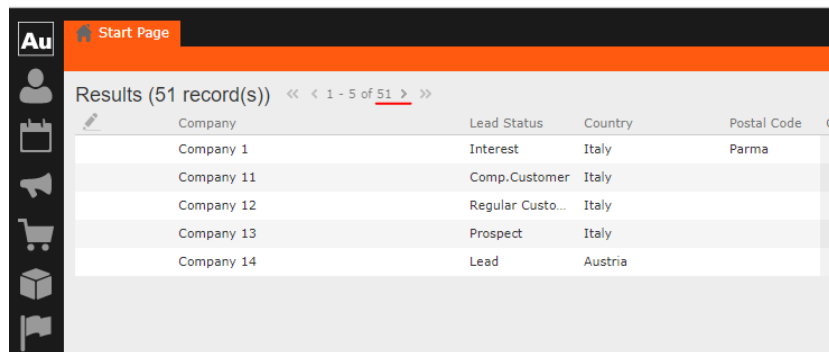
The header is then also used as title if the form is displayed in a tab.

- When displaying forms in a tab you can define a `pageTitle` attribute for the entire form. This attribute is only used if no `HeaderName` is specified in the action.
- Alternatively you can use the `pageTitle` parameter in the `ShowForm` action. `pageTitle` is only used if `HeaderName` is undefined. The `pageTitle` action parameter overrides the `pageTitle` attribute defined in the form.
- When displaying forms in a popup (without using the `ShowForm` action), you can add a `Header` widget to your form and check the `updatePopupTitle` parameter. The header is then displayed in the popup's title bar.
- If no header is found (and `pageTitle` undefined), the label of the button or menu item that called form is used as tab/popup title, see the article “How to Set the Title of a Tab” at <https://support.aurea.com>.

## Displaying the Total Number of Result Records in a List View

ListView widget configuration to display the number of result records in a search result list view.

You can configure the ListView widget to display the number of result records in a search result list view. This header usually contains the text **Results** followed by a count of the number of records available in the search result. It also contains the list browser buttons and allows users to navigate within the result list.



For example, the screenshot above shows the start page with a custom form to display the Search results for company records. The results header shows

« < 1 - 5 of 51 > » . By default it shows the number of records remaining in the N+ format « < 1 - 15 of 50+ > » .

If you wish to show the total number of records available in the search result, you have to set the `countRows` parameter in the ListView widget to `true`. The following sample start page form configured in the Designer produces the result as shown in the screenshot above:

```
"listView9": {
  "type": "ListView",
  "options": {
    "model": { "fieldControlType": "List", "fieldGroupName": "FI",
"fixedFilterName": "",
    "pageSize": 5, "infoAreaId": "FI" },
    "autoLoad": true,
    "showPaging": true,
    "countRows": true
  },
}
```

## Testing Your Forms

Use these steps to test your newly created forms.

Use the **SDK** page to test your forms:

1. Go to **SDK > Forms**.
2. Select **From Configuration**.

3. Enter the **Form name** as defined in CRM.Designer.
4. Click **Create**.

## Displaying Forms in Aurea CRM.Web

Different ways to display forms in Aurea CRM.Web.

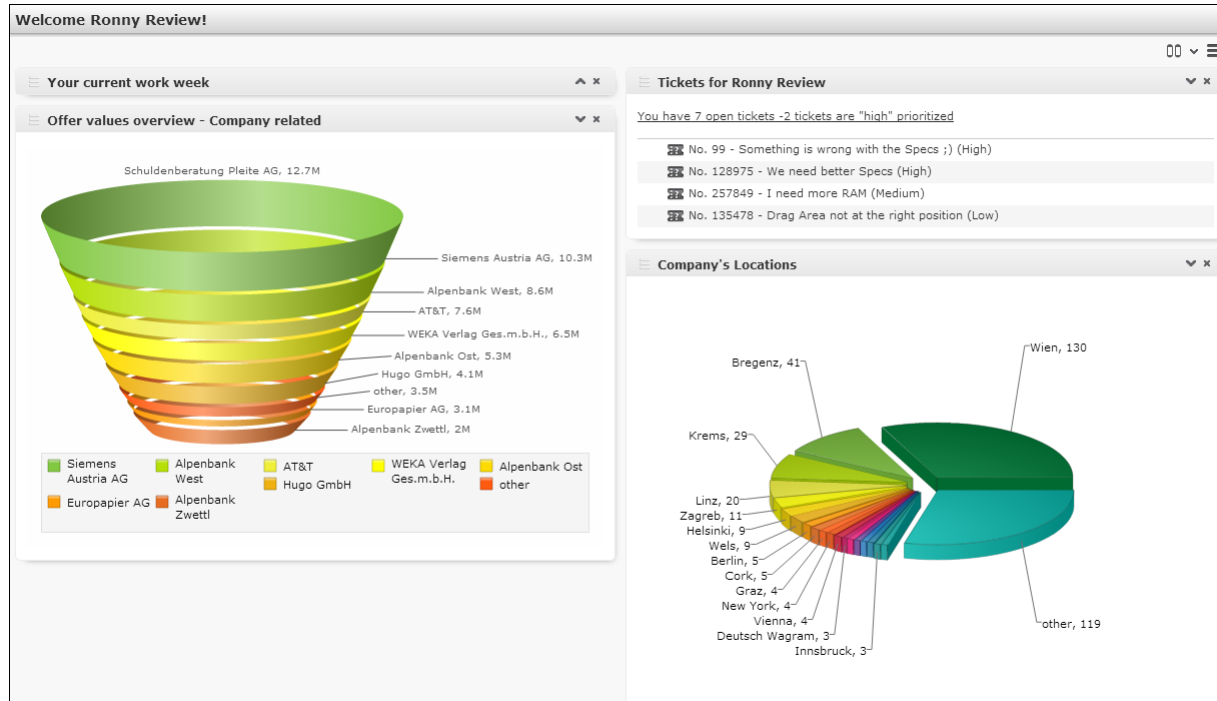
You can display your forms:

- using the `ShowForm` action defined for menu entries, buttons and within processes. For further details, see [ShowForm](#) on page 393.
- in sub-lists. For further details, see [Related Data area \(Header Info Areas\)](#) on page 329.
- in Tree View nodes. For further details, see [Record Settings](#) and [Info Area Settings](#).
- on ProcessAsk pages. For further details, see [FormName](#).

## Dashboards

Dashboards display different kinds of information - appointments, analyses, record lists - on one page.

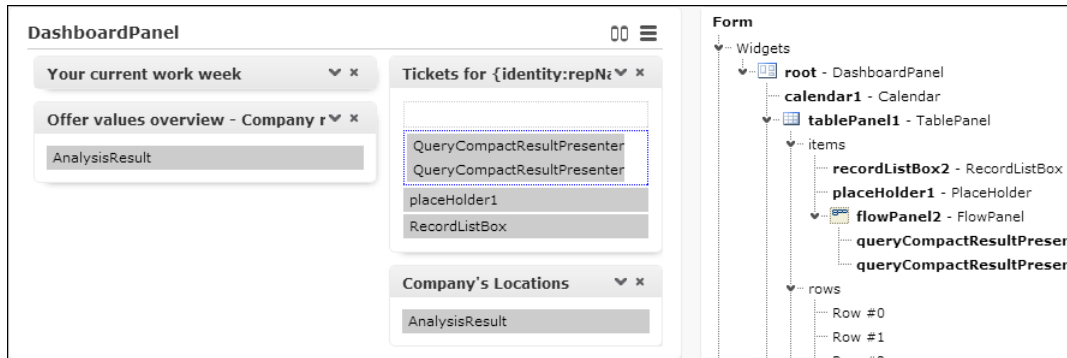
Dashboards can be used as start pages for specific users or user groups.





Users can switch between views (one column, two columns and tabbed), expand/collapse and reorder the areas of the dashboard. The user's settings are saved per client, allowing users to arrange their dashboards differently e.g. on a tablet PC and a notebook. The user's settings are saved separately once for the one column and tab views and once for the two columns view.

Use the `DashboardPanel` widget to configure dashboards. For further details, see [Configuring Dashboards](#).



To configure a dashboard:

1. Add the `DashboardPanel` widget as root widget to your form. The `DashboardPanel` widget can be added to any kind of panel widget as well.

By default, a two-column grid containing placeholders is created. Use the **layout** attribute to set/change this default (1Column | 2Columns | Tab).




The `DashboardPanel` widget uses the special header `DashboardPanel` containing the buttons for switching between views, hiding items and resetting the dashboard.

To configure an additional custom header (e.g. a personalized text like "Welcome {identity:repName}!" on a start page), define a special header and enter it as `HeaderName` in the `ShowForm` action. For further details, see [ShowForm](#) on page 393 and [Placeholders in Texts](#) on page 182.

2. Enter a `settingsName`: This name is used as storage key for the user customizations rearrange panels, selected view etc.
3. Add the desired items to the grid. For further details, see [Arranging Elements on a Form](#) on page 624. You can use any kind of widget: CRM widgets, panel widgets, generic widgets like buttons etc.

Each item is created inside a `DashboardPanelItem`, i.e. a specific area of the dashboard. Use panel widgets to add multiple items to one `DashboardPanelItem`.

- To reorder items, select the widget contained in the `DashboardPanelItem` and use the  (**Drag&Drop**) button.
- To hide items that contain no data (e.g. due to rights restrictions or an assigned CRM process), set the `hideItemsIfEmpty` property to `true`.

**Note:** Hiding empty items may impact performance because it requires pre-loading all widgets.

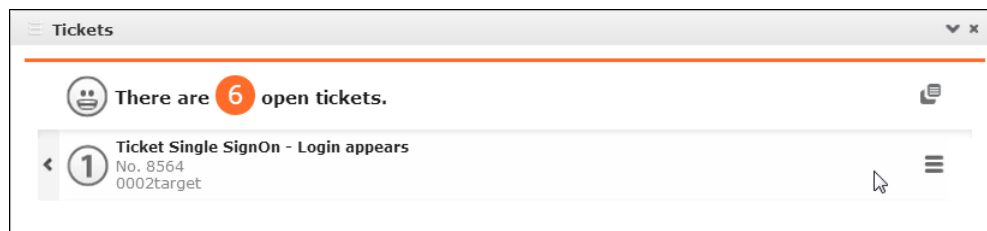
4. To edit the `DashboardPanelItem`'s settings, select one of it's items.




5. Define a **label** to be displayed as the caption of the respective area of the dashboard.

## LiveTile Widget

The LiveTile Widget is a slide show of various kinds of items which has a caption with placeholders and image based on the amount of displayed items.



Use the `LiveTile` widget on a start page, to display important information as well as an, optional, slide show of table captions. The records are displayed based on a query or query statement `queryName` or `queryStatement` property.

The header contains a **QuickView**  button, allowing the user to open the query result in a popup, configurable via property `action`. The same action is executed when clicking on the icon displaying the amount of records.

If the query returns no results and the `hideIfEmpty` property is set to `true`, the widget is not displayed. If the query returns more than `maxResults`, a + sign is added to the amount icon.

Clicking on an item executes the default action, e.g. opens an Expand view of the ticket record. The info area's default context menu (i.e. the context menu defined as **Info Area Default Menu** in the info area's default Expand configuration) is displayed for the items.

For details on configuring `LiveTile` widgets, see the article “How to Use the Live Tile Widget” at <https://support.aurea.com>.

For information on how to display images in the table captions depending on a field value. For further details, see [Configuring a Table Caption](#) on page 266.

Use the **SDK** page to test your `LiveTile` widgets: **SDK > LiveTile**.

## Showing/Hiding the Default Text of DateInput Widget

A `DateInput` Widget is a Generic Widget for selecting a date.

Generic Widgets can be used independently of records and other CRM-specific data (e.g. field schema or entries in the designer database). The visualization of these widgets in the designer preview is identical to the visualization on the web user interface, which makes them easy to configure and customize.

The following example describes how to show or hide the default text of a `DateInput` Widget, based on the web configuration parameter `DateInput.showDefaultText`, Designer Form settings, and Field Group settings. The default text for the `DateInput` Widget is displayed in the format **(date format string)**.

When the web configuration parameter `DateInput.showDefaultText` is set to `true`:

- Using Designer Form settings:
  - If the `DateInput` widget in the designer form has the property `showDefaultText` is set to `true`, then the default text for the widget is displayed. If the property is set to `false` then the property is not displayed.
  - If the designer form does not have the property `showDefaultText`, then the default text for the widget is displayed.
- Using Field Group settings:
  - If the field group has the extended option `showDefaultText` set to `true`, then the default text for the widget is displayed, else if the extended option is set to `false` the default text is not displayed.
  - If the field group does not have the extended option `showDefaultText`, then the default text for the widget is displayed.

When the web configuration parameter `DateInput.showDefaultText` is set to `false`:

- Using Designer Form settings:
  - If the `DateInput` widget in the designer form has the property `showDefaultText` set to `true`, then the default text for the widget is displayed. If the property is set to `false` then the property is not displayed.
  - If the designer form does not have the property `showDefaultText`, then the default text for the widget is not displayed.
- Using Field Group settings:
  - If the field group has the extended option `showDefaultText` set to `true`, then the default text for the widget is displayed, else if the extended option is set to `false` the default text is not displayed.
  - If the field group does not have the extended option `showDefaultText`, then the default text for the widget is displayed.

## Enterprise Features

The Enterprise license of CRM offers special features and add-ons to enhance CRM capabilities.

Following are the special features and add-ons offered by Aurea CRM to Enterprise users:

## Aurea CRM Add-in for Microsoft Word

Learn about the Microsoft Word Add-in for Aurea CRM.

Aurea CRM add-in for Microsoft Word allows CRM users to create documents based on templates that meet the corporate CI guidelines and enrich the documents with content from the CRM.

---

**Note:** You need to have the Enterprise License setup on Aurea CRM to work with the Word add-in.

---

CRM administrator and users can create template documents that can be used for various recurring tasks. Some of the tasks for different user roles include the following:

- A Sales representative could create and maintain template documents for offer letters, order sheets, email etc. Use the different information in the CRM system to enrich the document.
- Marketing personnel could create new templates to meet the latest customer interaction guidelines. Create templates for marketing campaigns sent by letters or flyers using information from the CRM.
- Service representatives can create solution documents or other documents for different services and use the CRM information in those documents.
- Field service representatives can create service report documents at the customer site along with the current ticket.
- All users can ensure that customers receive information in their preferred language.

---

**Note:** To send an email in a language other than the default languages, you should use UTF-16 encoding for the template document. For example, to send an email in Cyrillic, encode the template document in UTF-16. You can use any text editor that supports UTF-16, to encode the template document.

---

The Aurea CRM word add-in supports the following features:

- Allows you to use the same set of methods used by Aurea CRM.Web.
- Template documents can use fields that are used to store information in Aurea CRM.
- Serial and single letter fields can be easily integrated into the templates using drag and drop features.

## Installing Aurea CRM Add-in for Microsoft Word

Aurea CRM Add-in for MS Word is developed as a VSTO Add-in. Visual Studio Tools for Office (VSTO) Add-ins use the ClickOnce deployment solution which automatically updates your users' installation of CRM Add-in for Word on their machines. The recommended way to install Aurea CRM Word Add-in is to use a network shared location containing the latest installation package. As a CRM administrator you should set up a central shared location for the CRM Add-in for Word installation package. Using this process ensures that the latest version of the CRM Add-in for Word is automatically upgraded to the latest version. CRM users only have to install it once using the shared central location.

**Important:** ClickOnce installation checks this location for all subsequent updates. Ensure that you update this location with the latest package to ensure automatic updates.

If your organization cannot provide a shared location, the Aurea CRM add-in for Word installation package can be found on the Aurea CRM customer support site.

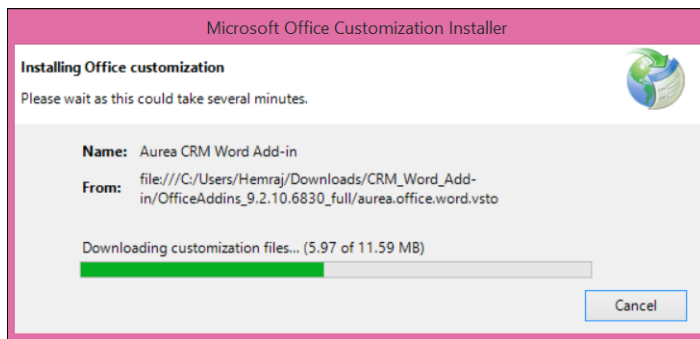
To install the Aurea CRM add-in for Word:

1. Browse to the network shared central location you have set up.

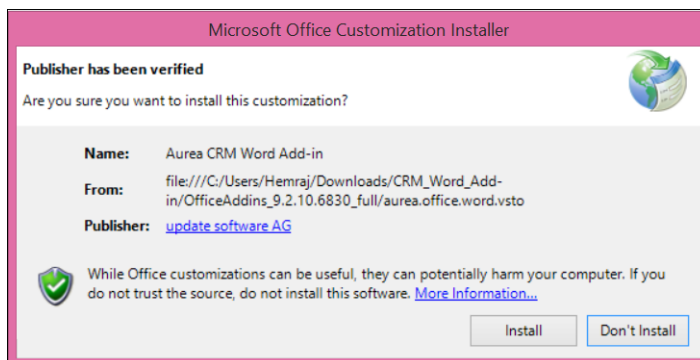
Or

Unzip the package `OfficeAddins_<versionNo>_full.zip`. The `<versionNo>` is the latest version available on the customer support site.

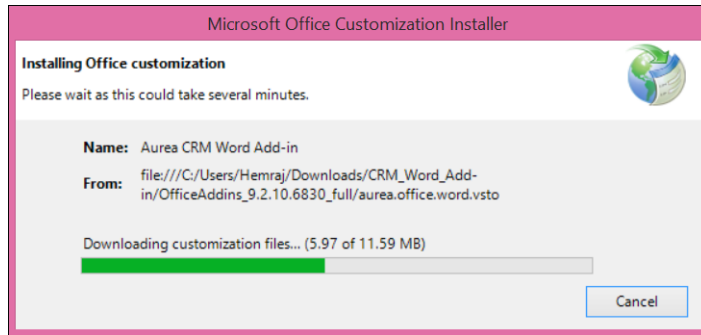
2. Double click `aurea.office.word.vsto` file in the shared location or the unzipped folder. The Microsoft Office Customization Installer window opens.



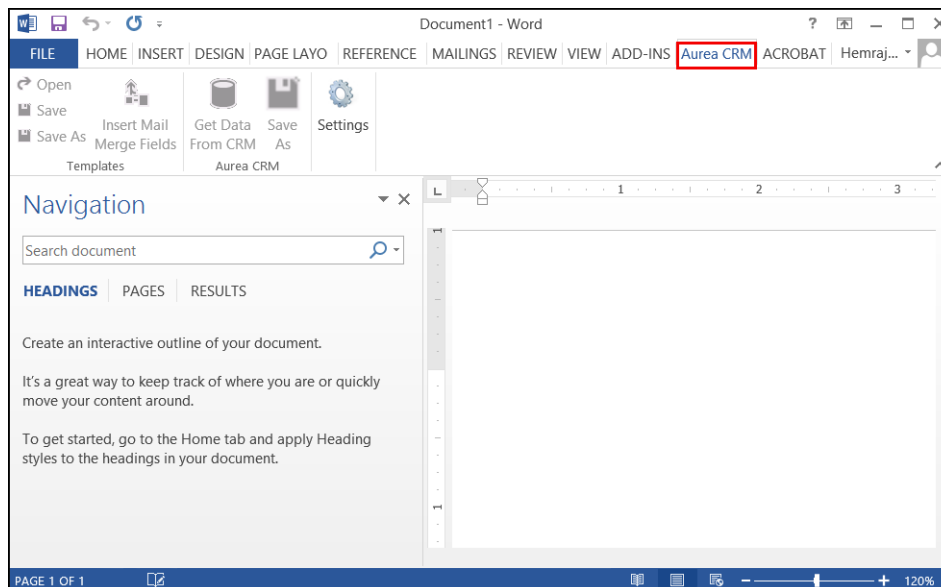
The installer verifies the publisher, in this case Aurea CRM, and then displays the following window.



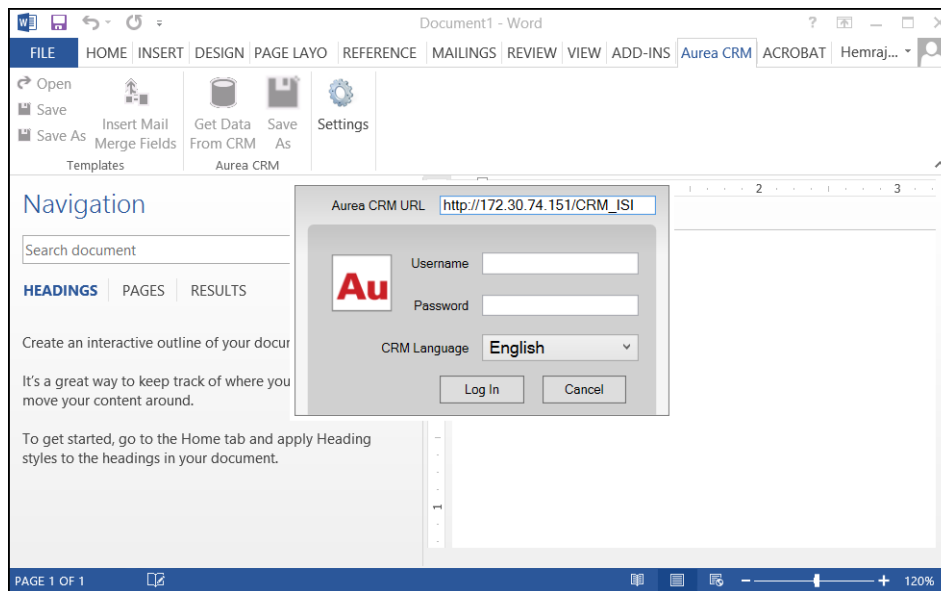
3. Click **Install**. The Installing Office Customization message displays.



4. The window automatically closes on successful installation.
5. Start Microsoft Word and open a blank document, the **Aurea CRM** menu item should be visible.
6. Click the **Aurea CRM** menu.



7. To work with the plugin you have to be connected to Aurea CRM. Click **Settings** to open the connection dialog.



8. Enter the following details in the connection dialog:
  - a. Enter the URL of the Aurea CRM server.
  - b. Enter your Username and Password.
  - c. Select the language you prefer to work on Aurea CRM.
9. Click **Log In**.
10. Now you can use the CRM Word Add-in to create templates for your documents.

### Setting-up Upgrade location for the Aurea CRM Add-in for Microsoft Word

Aurea CRM Add-in for MS Word is developed as a VSTO Add-in. Visual Studio Tools for Office (VSTO) Add-ins use the ClickOnce deployment solution which automatically updates your CRM Add-in for Word. This requires that you publish Aurea CRM Add-in to a network shared central location. Users can install the latest version from this location once, all subsequent upgrades then happens automatically. Publishing a new version to this location automatically updates all CRM Add-in for Word installed from this location.

---

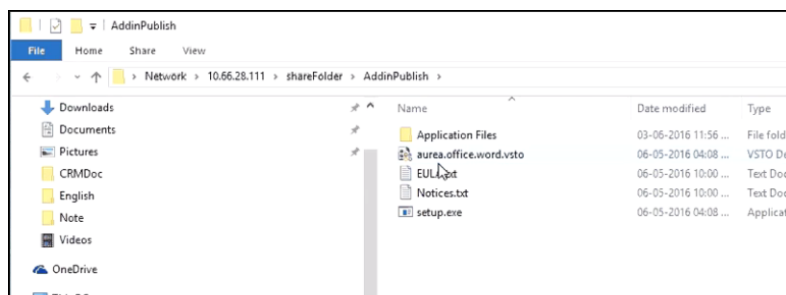
**Note:** If there are multiple users on a machine, then each user has to install his own version of CRM Add-in for MS Word. This is a limitation of the ClickOnce solution.

---

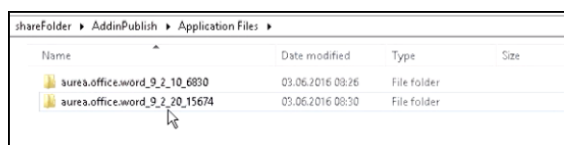
The following steps illustrate how to publish the latest version of word Add-in to the central location:

1. Unzip the package `OfficeAddins_<version>_release.zip`.
2. Create a network shared central location that is accessible to your users and copy the files in the unzipped folder to this location.





3. Instruct your users to access this folder and install CRM Add-in for Word using the `aurea.office.word.vsto` file. Installing from this folder is required only once, after this the updates happens automatically.
4. When a new version of CRM Add-in for word is released, unzip the new `officeAddins_<version>_release.zip` package and copy the files in the unzipped folder to the same network shared central location.
5. Select **Replace the files in the destination** when you prompted to overwrite the files.
6. The **Application Files** folder contains the new version as well as the older versions.



7. When the user starts up Microsoft Word the next time,

---

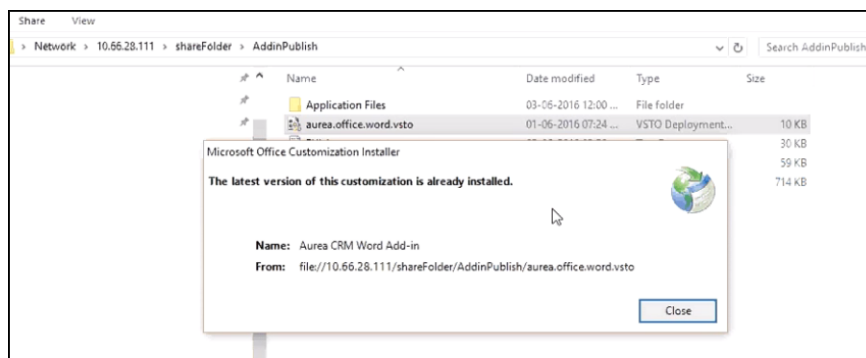
**Note:** ClickOnce installer checks for updates once in every seven days.

---

**Note:** The network shared folder should be accessible to the MS Word user for the automatic updates to work.

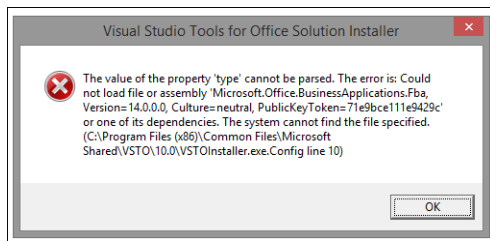
---

8. If the user wants a new version before the seven days check, he can go to the shared folder and update his current installation by using the `aurea.office.word.vsto` file.



## Troubleshooting the Aurea CRM Add-in for Microsoft Word

Sometimes the Aurea CRM add-in for Word installation may fail with an error message as shown in the following screenshot:



To resolve the issue, try the following solutions:

- Solution 1:
  1. Unzip the package `OfficeAddins_8.4.0.0_release.zip`.
  2. Right click `setup.exe` and select **Run as Administrator**.
  3. Continue with the installation from step 3, as described in the section above (Installing Aurea CRM Add-in for Microsoft Word).
- Solution 2:
  1. Rename file `C:\Program Files (x86)\Common Files\Microsoft\Shared\VSTO\10.0 \VSTOInstaller.exe.Config` to `Old-VSTOInstaller.exe.Config`.
  2. Continue with the installation from step 3, as described in the section above (Installing Aurea CRM Add-in for Microsoft Word).
- Solution 3:
  1. Rename file `C:\Program Files (x86)\Common Files\Microsoft\Shared\VSTO\10.0 \VSTOInstaller.exe.Config` to `Old-VSTOInstaller.exe.Config`.
  2. Unzip the package `OfficeAddins_8.4.0.0_release.zip`.
  3. Right click `setup.exe` and select **Run as Administrator**.
  4. Continue with the installation from step 3, as described in the section above (Installing Aurea CRM Add-in for Microsoft Word).

## Cloud Backup Service

Learn about the Cloud Backup Service in Aurea CRM.

Aurea CRM cloud backup service allows you to perform a Full backup and restore of the Aurea CRM data. The backup data includes the data for the following components:

- CRM.Win
- CRM.Web
- CRM Designer data

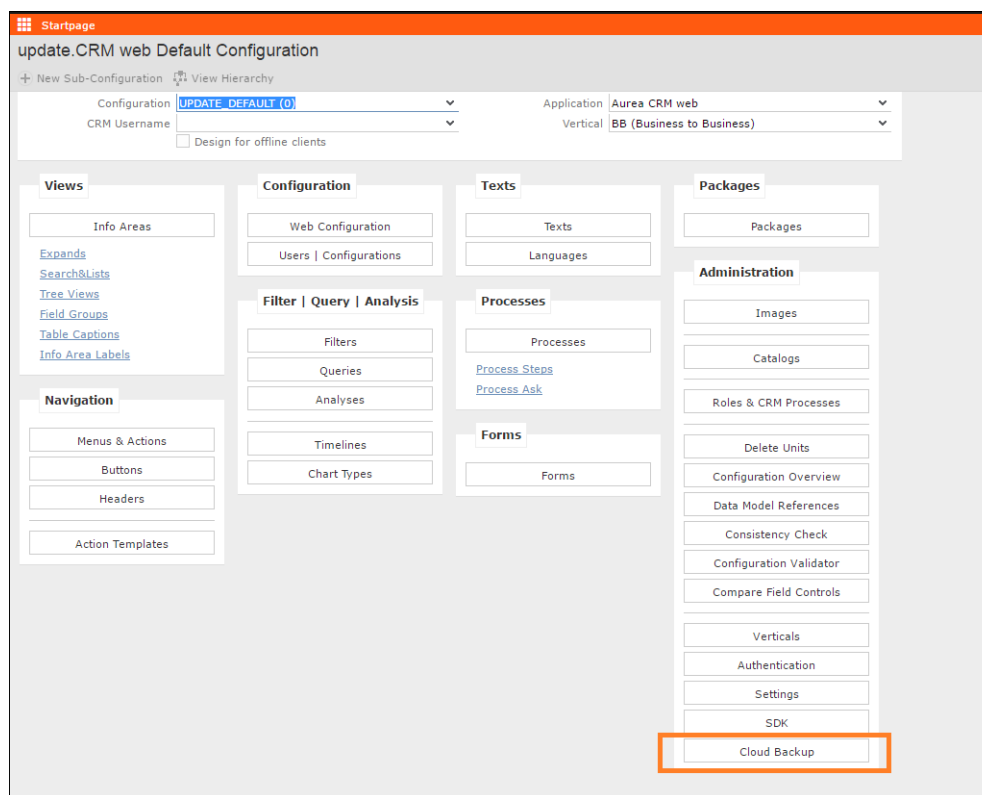
- CRM database
- CRM Designer database

The backup service provides password protection and proper encryption of the data. It allows you to schedule the data backup on a daily, weekly, and monthly basis. You can view and selectively restore data from the backup instances.

## Working with Cloud Backup Service

Learn to configure and manage Cloud Backup Service.

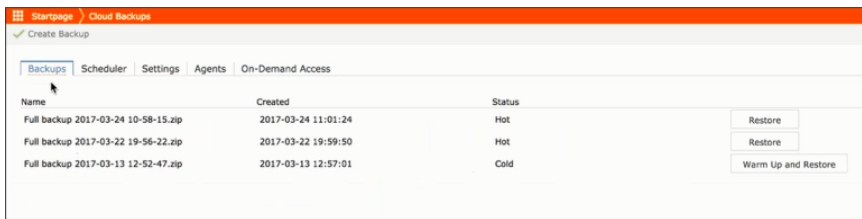
This Enterprise only feature allows you to configure the backup and restore services from the CRM.Designer interface. The following screenshot shows the **Cloud Backup** option in the designer.



The **Cloud Backup** page (see the screenshot below) provides the following tabs:

- **Backups** – This tab shows all the backups performed on the CRM and allows you to trigger a restore operation from a listed backup. It also shows the status of current operations in progress.
- **Scheduler** – This tab allows you to schedule a backup or restore operation. You can configure the following schedules:
  - **Once** – This operation is triggered only once at the set time and date.
  - **Daily** – This operation is triggered daily at the set time.

- **Weekly** – This operation is triggered weekly for the set days and time.
- **Monthly** – This operation is triggered monthly for the set months, dates, and times.
- **Settings** – This tab allows you to select the storage provider you want to use for the backup and restore operations.
- **Agents** – This tab allows you to provide the URL, port number and other details of the Backup Agent that is responsible for collecting packages, encrypting, packaging, and uploading backups to the cloud backup service.
- **On-Demand Access** – Provides access to cloud-based development and test environments for Aurea CRM and is available for SAAS customers. Customers can use these environments in the following scenarios:
  - Disaster recovery
  - Trying new versions of Aurea CRM before performing an actual upgrade.
  - Easily share their environments with Aurea Global Support for troubleshooting.

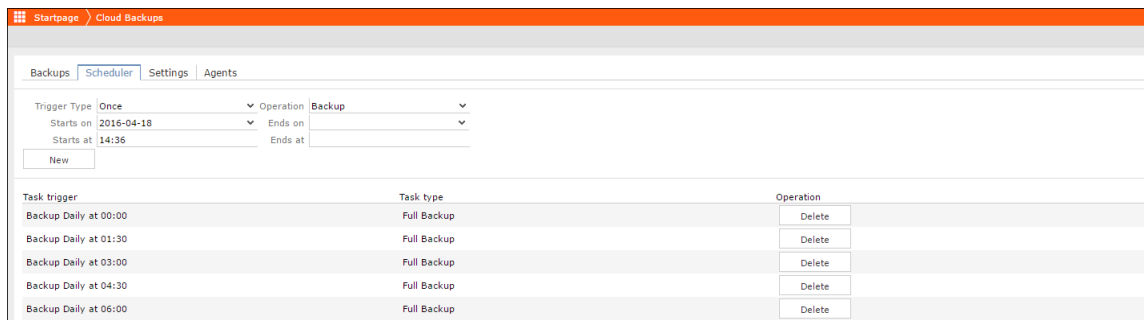


## Scheduling a Backup and Restore operation

Learn to schedule a Backup and Restore operation.

To schedule a backup or restore operation perform the following:

1. Click on the Cloud Backup option on the start page of the designer. The Cloud Backup page opens.
2. Click on the Scheduler tab. The Scheduler tab opens.



3. Select the operation type from the Operation drop-down list. Available operations are **Backup** and **Restore**.
4. Select the operation schedule using the Trigger Type drop-down list.

---

To set up a once only backup operation, perform the following steps:

- a. Select **Once** from the Trigger Type field.
- b. Select the scheduled date from the **Starts on** date field.
- c. Enter the time to trigger the operation in the **Starts at** text field. The format should be in `<hours>:<minutes>` form. For example, in the image above to trigger the operation at 4.36 PM, enter the value 14:26.
- d. Click **New**, to add a new entry in the list of scheduled items. The Task Trigger column should have the entry in the form `<operation type> Once at <date> <time>`. For above example, it has the entry “Backup Once at 2016-04-18 16:36”.

To schedule a daily backup operation, perform the following steps:

- a. Select **Daily** from the Trigger Type field.
- b. Select the scheduled start date from the **Starts on** date field.
- c. Select the scheduled end date from the **Ends on** date field.
- d. Enter the time to trigger the daily operation in the **Starts at** text field. The format should be in `<hours>:<minutes>` form. For example, in the image above to trigger the operation daily at 4.36 PM, enter the value 14:26.
- e. Click **New**, to add a new entry in the list of scheduled items.

The Task Trigger column should have the entry in the form `<operation type> Daily at <time> until <End Date>`. For example, “Backup Daily at 16:36 until 2016-04-23 00:00”.

To schedule a weekly backup operation, perform the following steps:

- a. Select **Weekly** from the Trigger Type field.
- b. Select the scheduled start date from the **Starts on** date field.
- c. Select the scheduled end date from the **Ends on** date field.
- d. Enter the time to trigger the daily operation in the **Starts at** text field. The format should be in `<hours>:<minutes>` form. For example, in the image above to trigger the operation daily at 4.36 PM, enter the value 14:26.
- e. From the **Week Days** field, select the days of the week you want to trigger the operation.
- f. Click **New**, to add a new entry in the list of scheduled items.

The Task Trigger column should have the entry in the form `<operation type> weekly at <time> on days <weekday1>, <weekday2> etc until <End Date> 00:00`. For example, “Backup weekly at 16:36 on days Monday, Friday until 2016-04-23 00:00”.

To schedule a monthly backup operation, perform the following steps:

- a. Select **Monthly** from the Trigger Type field.
- b. Select the scheduled start date from the **Starts on** date field.
- c. Select the scheduled end date from the **Ends on** date field.
- d. Enter the time to trigger the daily operation in the **Starts at** text field. The format should be in `<hours>:<minutes>` form. For example, in the image above to trigger the operation daily at 4.36 PM, enter the value 14:26.
- e. From the **Months** field, select the months of the year you want to trigger the operation.
- f. From the Month days field select the days of each selected month you want to trigger the operation.
- g. Click **New**, to add a new entry in the list of scheduled items.

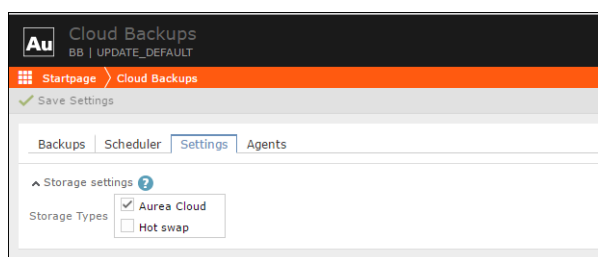
The Task Trigger column should have the entry in the form `<operation type> monthly at <time> on a day <date1>, <date2>, <date3> etc of <month1>, <month2>, <month3> etc until <end date> 00:00`. For example, “Backup Monthly at 16:37 on a day 5, 12, 19, 26, 30 of May, June, July, August until 2016-08-31 00:00”.

## Configuring the Storage Settings

Learn to set-up the storage settings.

To configure the storage settings, perform the following tasks:

1. Click on the Cloud Backup option on the start page of the designer. The Cloud Backup page opens.
2. Click on the **Settings** tab. The Settings tab opens.



3. Select the **Storage Type** from the list:
  - **Aurea Cloud** – Uploads backups to the Aurea CloudService, which efficiently stores backups in the cloud. Aurea CloudService can both upload and restore the backups saved earlier.
  - **Hot swap** - Allows you to send backups to a third party backup agent service, which restores the backup remotely, such that another instance can serve only as backup destination, not the source. Use this option in failover scenarios, to have a hot instance in standby.

## Adding or Deleting a Backup Agent

Learn to manage a Backup Agent.

To add a backup agent, perform the following tasks:

1. Click on the **Cloud Backup** option on the start page of the designer. The **Cloud Backup** page opens.
2. Click on the **Agents** tab. The **Agents** tab opens.

The screenshot shows the 'Cloud Backups' page with the 'Agents' tab selected. The interface includes a navigation bar with 'Startpage' and 'Cloud Backups'. Below the navigation bar are tabs for 'Backups', 'Scheduler', 'Settings', and 'Agents'. The 'Agents' tab is active, displaying a form with the following fields: 'Server address' (empty), 'Server port' (14567), and 'Application name' (backups). A 'New' button is located below the form. At the bottom of the page, there is a list of backup agents with the entry 'localhost:14567/backups' and a 'Delete' button next to it.

3. Enter the URL of the backup agent server in the **Server Address** field.
4. Enter the active port number of the backup agent server in the **Server Port** field.
5. Enter the name of the backup agent application in the **Application Name** field.
6. Click **New** to create the new backup agent.
7. The new backup agent is listed in the page below in the format <Server address>:<Port>/ <Application Name>.

To delete a backup agent, perform the following tasks:

1. Click on the **Cloud Backup** option on the start page of the designer. The **Cloud Backup** page opens.
2. Click on the **Agents** tab. The **Agents** tab opens.

This screenshot is identical to the previous one, showing the 'Cloud Backups' page with the 'Agents' tab selected. The form fields and 'New' button are the same. The list of backup agents at the bottom shows 'localhost:14567/backups' with a 'Delete' button next to it, indicating the step to delete the agent.

3. Click the **Delete** button to delete the backup agent listed.

## Restoring a Backup Instance

Learn how to restore a Backup Instance.

All the backup instances are listed in the Backups tab. The backup list provides the following information about the backup instances:

- **Name** – The name of the backup instance. This is in the format: `Full backup <date> <start time>.zip`. For example, `Full backup 2016-04-18 13-31-17.zip`.
- **Created** – The date and time the backup was completed.
- **Status** – The backup can be in one of two states:
  - **Hot** – This means that the backup package is ready for restoration. All recent packages less than 7 days old are in a hot state and ready for restoration.
  - **Cold** – This means that the backup package is more than 7 days old and it needs to be downloaded from the long term cold storage server (for example, Glacier) and made ready for restoration. This process is called warmup. It takes at least 4 hours for a backup package to be downloaded at

To restore a backup data, perform the following tasks:

1. Click on the **Cloud Backup** option on the start page of the designer. The **Cloud Backup** page opens
2. Click on the **Backups** tab. The **Backups** tab opens.
3. To restore a backup data, perform the following:
  - If the listed backup is in a hot state then click **Restore** to restore the backup data.
  - If the listed backup is in a cold state then click **Warmup and Restore** to restore the backup data.

## On-Demand Access

Aurea CRM provides on demand access to cloud based development and test environments.

You can use the test environments in the following scenarios:

- Disaster recovery
- Trying new versions of Aurea CRM before performing an actual upgrade.
- Easily share their environments with Aurea Global Support for troubleshooting.

The following features are provided by Aurea CRM On-Demand access:

- CRM.Designer allows you to create, start, stop, and delete these environments for all the CRM verticals.
- You can access these environments using predefined links and all communication with these environments are secured using SSL (Secured Socket Layer).

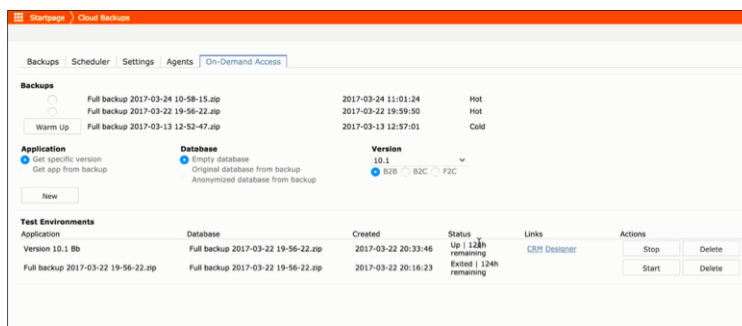


- You can define the backup using the cloud backup feature provided in the CRM.Designer.
- You can use these environments with or without the backup data. You can use the environment using a backup and specify the version number you want to use.
- You can only get the CRM.Web and CRM.Designer functionality using these environments.

The following security measures are taken when creating a test environment:

- All users except for SU and WWW are deactivated in the test environments.
- You can create anonymized databases from the backups when creating a test environment. Data can be masked at the infoarea or field level.

The following screenshot shows a sample On-Demand Access page on the CRM.Designer.



The On-Demand Access page consists of three sections: **Backups**; **Application**; **Test Environment**. A brief description of each is provided below.

## Backups

This section of the page displays the list of backups available and created previously. You have to select a backup for use in creating the test environment. For more information on creating and working with backups, see [Scheduling a Backup and Restore operation](#) on page 644.

## Application

This section allows you to create the test environment. You can configure your test environment by specifying different properties like using the application data from the backup, create an empty database or use the database present in the backup, the version number of the CRM that is used as well as the vertical type of the applications created in the test environment.

- You can only create one test environment with a specific configuration. If you try to create more than one test environment using the same configuration, error message is received.
- If you are creating a test environment that uses the applications and the database from the backup then the test environment created is restricted to the version of the CRM used to create the backup.
- Test environments have a limited life span and expires after a certain number of hours.
- All the test environments are deployed to the cloud infrastructure maintained by Aurea Software.

### Test Environment

This section lists the test environments created. You can see configuration details of the test environment. It also provides you with controls to start, stop and delete the test environments and the generated links to the CRM.Web and CRM.Designer created in the test environment.

### Creating a Test Environment

Learn how to create a Test Environment.

To create a new test environment, perform the following steps:

1. Click the checkbox besides the backup listed in the Backup section.

---

**Note:** The backup source should be in a hot state and ready for deployment. For more information, see [Restoring a Backup Instance](#) on page 648.

---

---

**Note:** You have to select a backup to create a test environment.

---

2. Select the Application to be created from the following options:
  - **Get Specific Version** – Use this option if you want to create a test environment without any of the customizations created in the backup.
  - **Get app from backup** – Use this options if you want to import the customizations created in the CRM server from which the backup was created.

---

**Note:** You cannot use the database from the backup if you want to use a specific version of the CRM server in the test environment.

---

- Select the **Database** option that you want to specify from the following options:
  - Empty Database** – select this option to create a test environment with empty database.
  - Original Database from backup** – select this option if you want to import the data from the backup to the test environment database.

---

**Note:** If you are importing the customizations in the backup then you cannot use a specific version of the CRM to create the test environment.

---

- Anonymized database from backup** – use this option if you want to import the database from the backup but you do not want certain data to be visible. For more information on anonymizing the data, see [Anonymizing the Backup Database](#) on page 652.
- Select the **Version** of the CRM servers with which you want to create the test environment from the drop-down list.
  - Select the required Industry Vertical configuration for the CRM servers in the test environment. You can select from the following options: **B2B**; **B2C**; and **F2C**.

---

**Note:** The above two options is disabled if you have opted to import the customizations and the database or anonymized database from the backup.

---

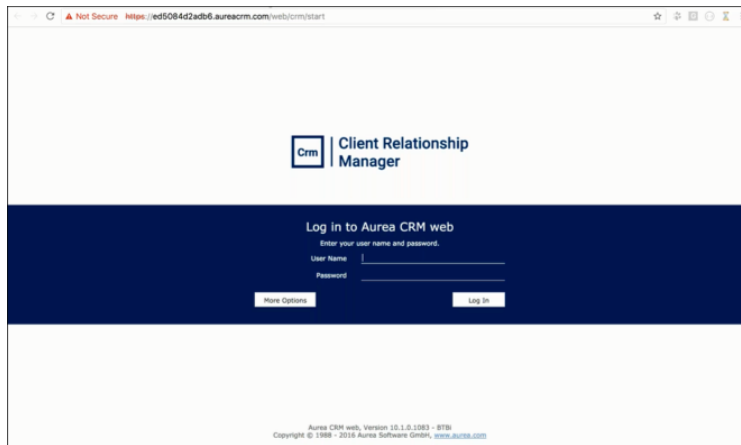
- Click **New**. The new test environment creation is initialized and it is listed in the **Test Environment** section of this page. The Status column shows Initialization in the screenshot below. And the Actions column has the **Delete** button activated. Click **Delete**, if you wish to stop creating the test environment.

Test Environments					
Application	Database	Created	Status	Links	Actions
Version 10.1 Bb	Version 10.1 Bb	2017-03-24 16:07:54	Initializing		Delete
Version 10.1 Bb	Full backup 2017-03-22 19-56-22.zip	2017-03-22 20:33:46	Exited   124h remaining		Start Delete
Full backup 2017-03-22 19-56-22.zip	Full backup 2017-03-22 19-56-22.zip	2017-03-22 20:16:23	Exited   124h remaining		Start Delete

Once the initialization is complete, the status of the test environment created changes to Up. It also shows the lifespan of the test environment (in hours) before it is terminated. The Links column contains links to the CRM.Web and CRM.Designer created in the test environment. The Actions column nows has the Stop and Delete buttons. See the screenshot below.

Test Environments					
Application	Database	Created	Status	Links	Actions
Version 10.1 Bb	Version 10.1 Bb	2017-03-24 16:07:54	Up   167h remaining	CRM Designer	Stop Delete
Version 10.1 Bb	Full backup 2017-03-22 19-56-22.zip	2017-03-22 20:33:46	Exited   124h remaining		Start Delete
Full backup 2017-03-22 19-56-22.zip	Full backup 2017-03-22 19-56-22.zip	2017-03-22 20:16:23	Exited   124h remaining		Start Delete

- Click on the CRM or Designer link to open the server in a new tab. The following screenshot shows the CRM.web server login window in a new tab.



Notice the URL, it uses the SSL to establish the connection to the CRM.Web in the test environment. It is of the format `https://<unique token>.<domain name>/web/crm/start`. The unique token ensures that you always connect to the correct instance of the test environment. The domain name is specified by the `<CrMWebUrl>` configuration element. For more information on configuration, see [Configuring the On-Demand Access settings](#) on page 654.

8. Click the **Stop** button to stop the test environment. The **Start** button gets activated.
9. Click the **Delete** button to delete the test environment. The Status of the test environment changes to **Deleted**.

### Anonymizing the Backup Database

Learn to make your Backup Database Anonymous.

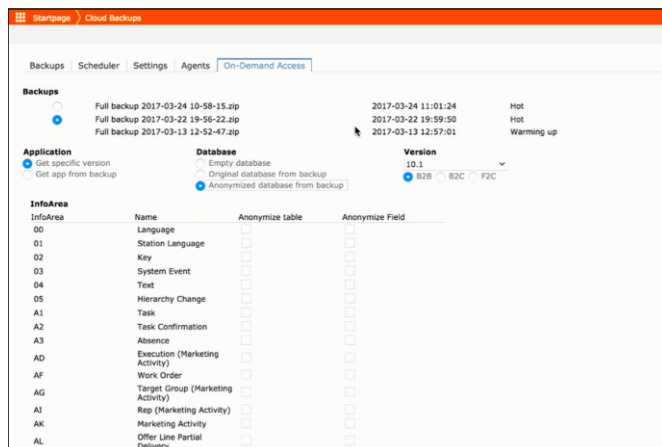
Anonymizing the database allows you to create a test environment and load the database from your backup, while restricting sensitive data from being visible to the user. This is very useful when you want allow Aurea support teams to troubleshoot the issues on your CRM environment, while preserving the integrity of your sensitive data.

You can anonymize data in the database at two levels of data hierarchy:

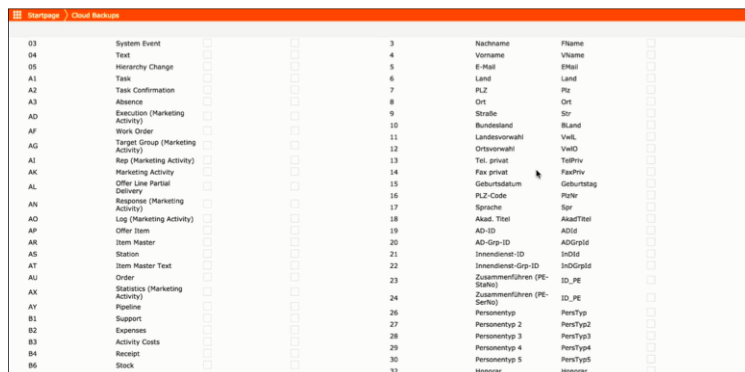
- Anonymize the infoarea by anonymizing the table for the infoarea.
- Anonymize specific fields in the infoarea by selecting the required fields in the table.

To anonymize data in your database, perform the following steps:

1. Select the **Anonymized database from backup** option in the Database column. The database table is displayed at the bottom of the screen. See the screenshot below.



2. You can select the infoarea that you want to anonymize by selecting the checkbox under the Anonymize Table column for the info area. In this example, we select the Company infoarea as we do not want to share the Company details of our customers.
3. You can also selectively anonymize the fields of an infoarea by selecting the checkbox under the Anonymize Field column. The list of all available fields in the infoarea are displayed to the right of the Anonymize table. In the following screenshot all the fields of the Person infoarea is displayed.



4. Select the fields you want to anonymize by click the checkbox to the right of the field. The selected fields is not anonymized and is not visible in the test environment.

## Configuring the On-Demand Access settings

Learn to set-up Aurea CRM for On-Demand Access settings.

You have to configure the `<update.configuration>` element in the `settings.xml` file for the CRM.Designer in the CRM server installation location. The following elements need to be set:

- `<CrmWebUrl>` - The URL for the Aurea CRM.Web application.
- `<CrmWebUserName>` - The CRM.Web super user username that is used to connect to the test environment server.
- `<CrmWebPassword>` - The password for the CRM.Web user.

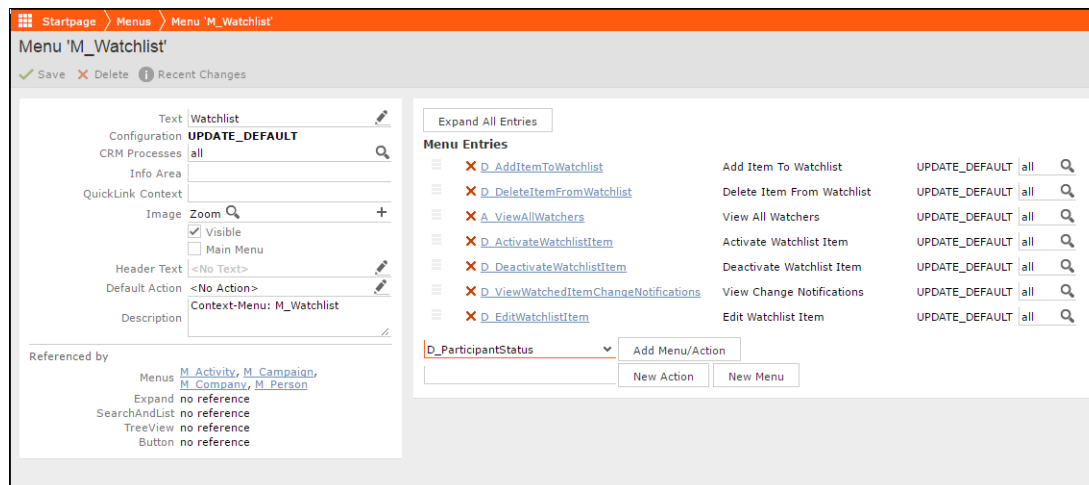
## Change Notifications Service

Aurea CRM change notification service allows a user to be notified about changes to records and events of interest.

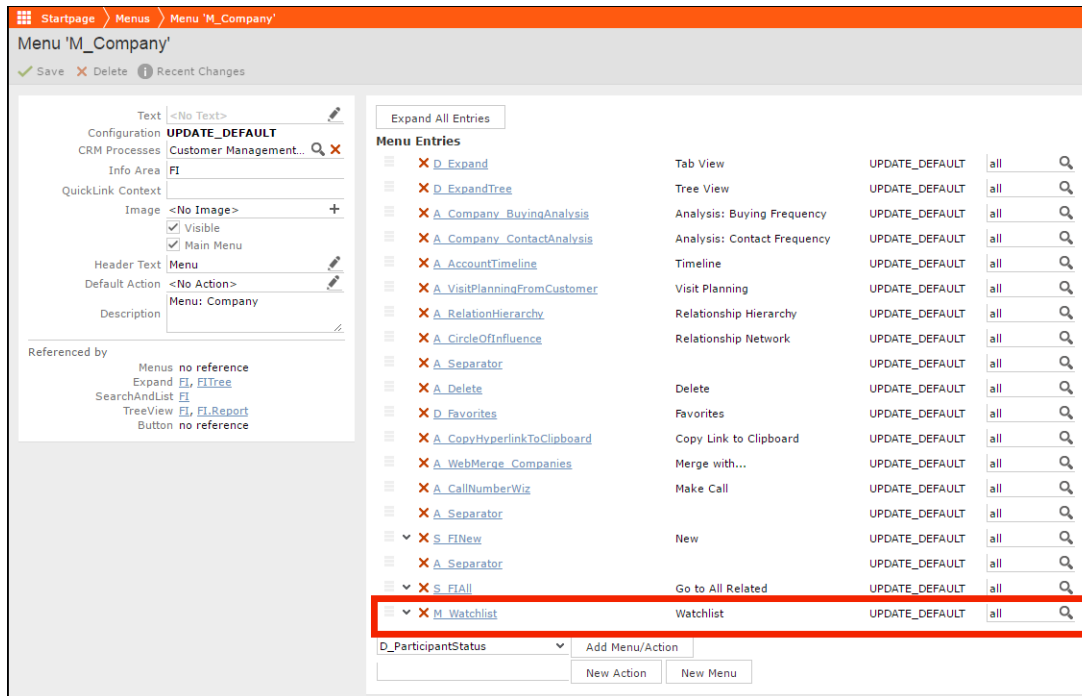
The user can track changes on his dashboard, as well as use other notification types like Email, Popup, and Record. The user can add items of interest to his personal watchlist.

In order to enable change notifications on a record, you should add the **M\_Watchlist** menu to the corresponding infoarea's menu entries. This ensures that the Watchlist menu is added to the context menu of the record's infoarea.

The following screenshot shows the M\_Watchlist menu, which is part of the UPDATE\_DEFAULT configuration. The **Referenced by** section of the window shows the Infoarea menu that point to this watchlist menu. In the screenshot below, the M\_Company menu for the Company infoarea contains the M\_Watchlist menu.



To add the watchlist to an infoarea menu, open the infoarea's menu entry and add **M\_watchlist** to its **Menu Entries**. The following screenshot of **M\_Company** menu shows the **M\_Watchlist** menu added to the list of its **Menu entries**.



After adding the Watchlist menu to an infoarea menu, you have to add the infoarea to the list subscribed infoareas to the web configuration parameter `Watchlist.AvailableInfoAreas`. This list is made available in the infoarea combo box in the search pages for watchlist subscriptions and change notifications.

The change notification service is based on a change notification engine. The change notification engine uses a database that is independent of the designer database. When a notification item is created for the first time, the notification engine creates the required database automatically. The databases supported out of the box are Oracle and MySQL databases.

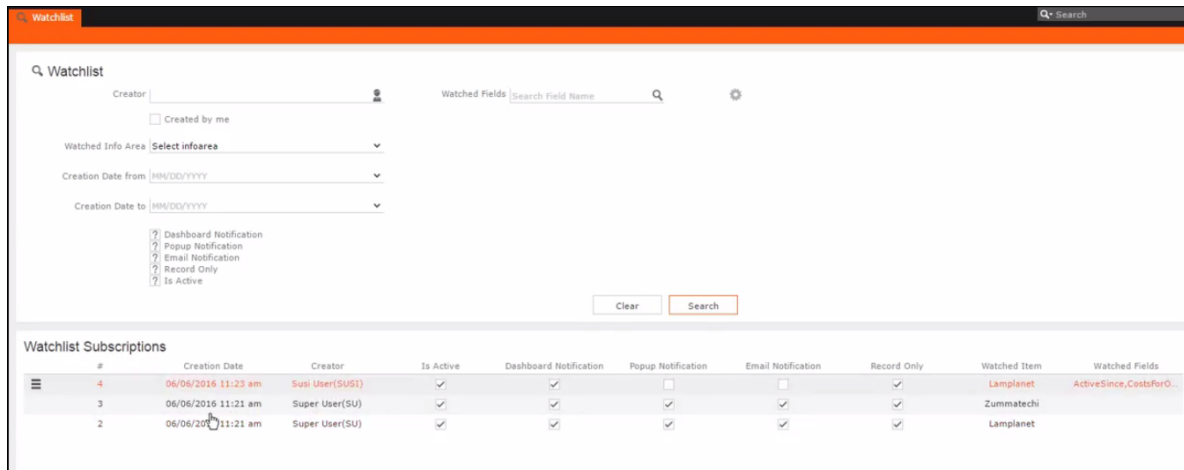
Aurea CRM server uses ASP.NET SignalR library to push change notifications to the connected web clients as soon as a change notification is generated. This ensures that change notifications are delivered in real time.

## Watchlist and Notifications

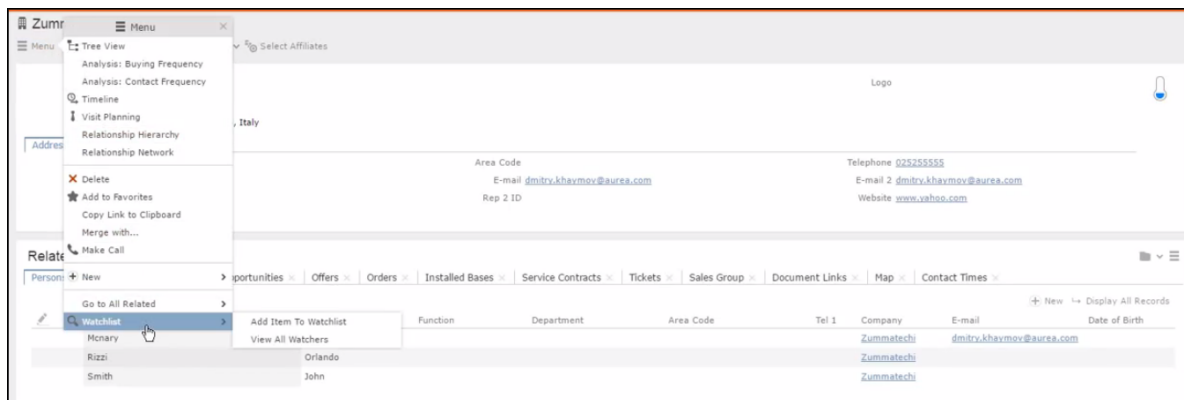
Aurea CRM users can set up their personal watchlist of items to be notified about changes.

For example, an accounts manager might be interested in knowing when a change occurs with his key accounts and he can add the accounts to his watchlist and set the notification types.

The following screenshot shows the list of watchlist items subscribed by all users:



For a given record or infoarea the Watchlist context menu initially allows you to add a watchlist item. For example, the screenshot below shows the **Add Item to Watchlist** and **View all Watchers** sub-menu item for the Company infoarea.

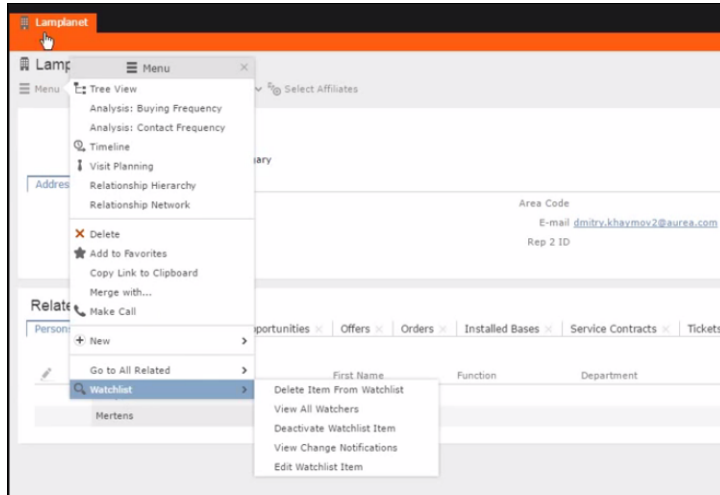


After adding an item to the watchlist the Watchlist context menu allows you to perform the following tasks:

- Delete items from the watchlist.
- View all watchers.
- Deactivate watchlist items.
- View change notifications.
- Edit watchlist item.



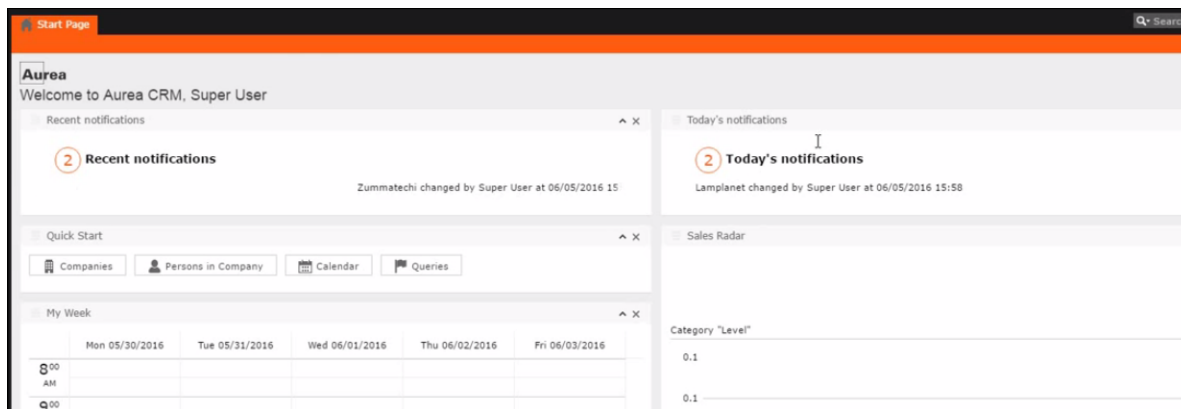
- See the screenshot below:



A notification is generated when one of the following condition is fulfilled:

- A notification request is set for any change to the record of interest and a change is made to the record.
- A notification request is set for changes made to specific fields of a record and one or more of the specified fields are changed.

The following screenshot shows the change notifications visible on the Dashboard:

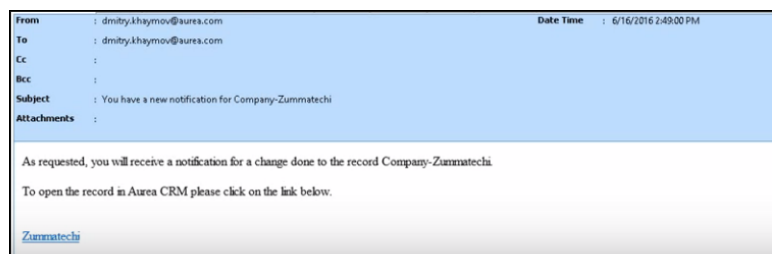


The **Recent Notifications** widget on the dashboard displays the recent change notifications generated for a certain number of days excluding the notifications generated today. The number of days is set in the web configuration parameter `Watchlist.RecentNotifications.PeriodInDays`.

The following screenshot shows a popup change notification:



The following screenshot shows a sample email change notification for a company record.



The subject, body, from, and number of retries for an email notification can be configured using the web configuration parameters: `Email.ChangeNotification.Body`; `Email.ChangeNotification.From`; `Email.ChangeNotification.Subject`; `Email.ChangeNotification.MaximumEmailSendAttempts`. For more information, see [Change Notification Parameters](#).

## Google Analytics Integration

Aurea CRM integration with Google Analytics enables you to get an insight into the usage patterns of Aurea CRM.Web.

Using Google analytics you can track events by user's roles and understand the usage of the CRM.Web application by your users. This can provide you with valuable insights into your business as well as measure the results of your marketing campaigns, behavioral flow patterns of CRM.Web users, help develop KPIs by collecting data ratios and counts.

Some of the key information based on user roles that can be extracted by the Google Analytics integration are listed below:

- Usage frequency metrics
- Usage frequency of the components of the CRM.Web application.
- The info areas used most, for example KP, FI etc.
- Usage frequency of collateral.
- Frequency of tasks like adding new data, editing data, or viewing data. Including the frequency of access of a specific data.
- Interface usage flow metrics

- Behavior flows that show how users move to their target screens.
- Real time information on events triggered by users.
- Usage metrics
- Understand users with different roles and where they spend their time.
- Average time on a specific action (creating a quota, updating forecast, etc.).
- Time spent on data entry?
- Time spent on a certain process (preparing a search, campaign design, creating or updating records)?
- Time spent on searching for items - can we see if the searches are repetitive or common?
- Some examples of frequency of action and events for a given user role:
- How often do sales reps search for items?
- How often do sales reps create records and activities?
- How often do marketing managers access reports? And which reports?

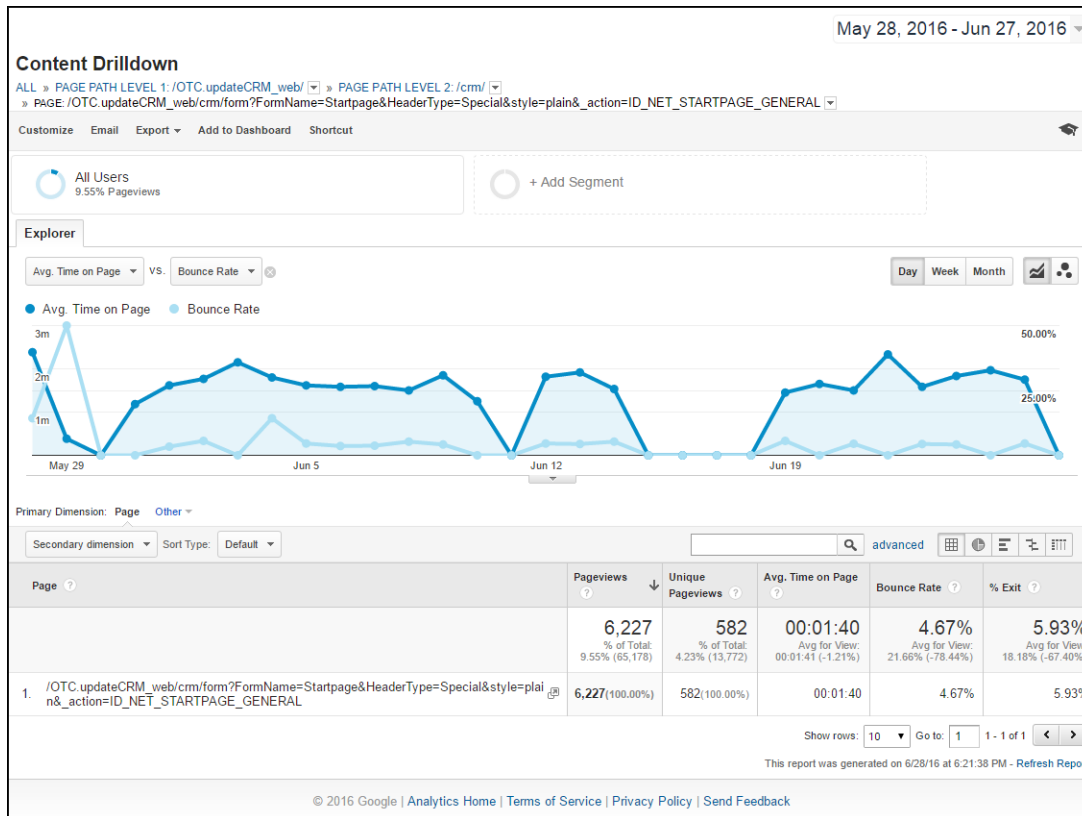
---

**Note:** Enterprise users can use their own Google Analytics tracker and can further customize the tracked components. Standard users of Aurea CRM can use Google Analytics tracker provided by Aurea and have the option to turn it off.

---

The following screenshots show the metrics captured on screen events, user sessions, and screen behavior flows of the CRM.Web users by Google analytics.

The following screenshot shows a **Behavior** report on the **Average Time on Page** versus the **Bounce Rate**. This report allows you to drill down to specific pages. The drill down path **ALL » PAGE PATH LEVEL 1: /OTC.updateCRM\_web/ » PAGE PATH LEVEL 2: /crm/ »** is visible at the top of the page. You can also select other metrics to compare or view user behavior on a single metric. As well as other capabilities.



## Configuring the Google Analytics Tracker

Learn to set-up the Google Analytics Tracker.

To track the CRM.Web application using Google Analytics, you can:

1. Use the main tracker provided by Aurea CRM. This tracks all aspects of CRM.Web application. Use the `GoogleAnalytics.MainTrackerEnabled` parameter to enable tracking by Google Analytics tracker provided by Aurea. Tracking for CRM.Web application is disabled by default.
2. Enterprise customers can use their own Google Analytics tracker and they can specify if the user ids should or should not be sent to the tracker.

To configure your own instance of Google Analytics tracker, perform the following steps:

**Note:** These instructions are a minimal subset of configurations possible for Google Analytics. Refer to the Google Analytics Help Center for detailed instructions and possibilities.

1. Disable tracking by Google Analytics tracker provided by Aurea by setting the parameter `GoogleAnalytics.MainTrackerEnabled` to `false`.
2. Open the Google Analytics home page configured for your organization.
3. Select the **Admin** tab. The **Administrator** page displays.

## Create a new property

1. Under the **PROPERTY** column, select **Create New Property** from the drop down list. The **New Property** page displays.

The screenshot shows the 'New Property' page in Google Analytics. The breadcrumb trail is 'Administration > New Property'. The page title is 'Aurea Global'. The main heading is 'New Property'. Below the heading, there is a sub-heading 'What would you like to track?' with two buttons: 'Website' and 'Mobile app'. The 'Website' button is selected. Below this, there is a 'Tracking Method' section with a dropdown menu set to 'Universal Analytics'. The 'Setting up your property' section includes a 'Website Name' field with the value 'My New Website', a 'Website URL' field with the value 'http://www.mywebsite.com', an 'Industry Category' dropdown set to 'Select One', and a 'Reporting Time Zone' dropdown set to 'United States'. At the bottom, there is a 'Get Tracking ID' button and a 'Cancel' button.

This page allows you to create the tracking id for your Aurea CRM.Web application. The tracking allows google analytics to track the data on your Aurea CRM site. Click the **Website** button and provide the following details:

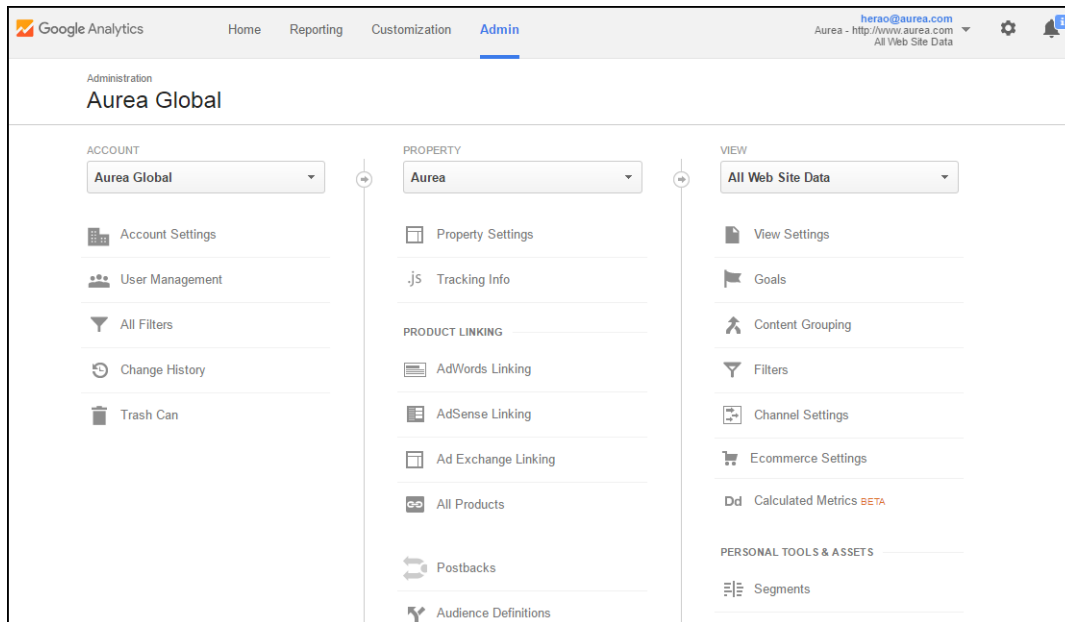
- Website Name: The name of the website configuration you are creating for Aurea CRM.
- Website URL: The URL of your Aurea CRM.Web application.
- Industry Category: Select the category of your industry.
- Reporting Time Zone: Select your country to set the time zone.

---

**Note:** In the following sample screenshots we use the website name **Aurea** to configure the remaining properties.

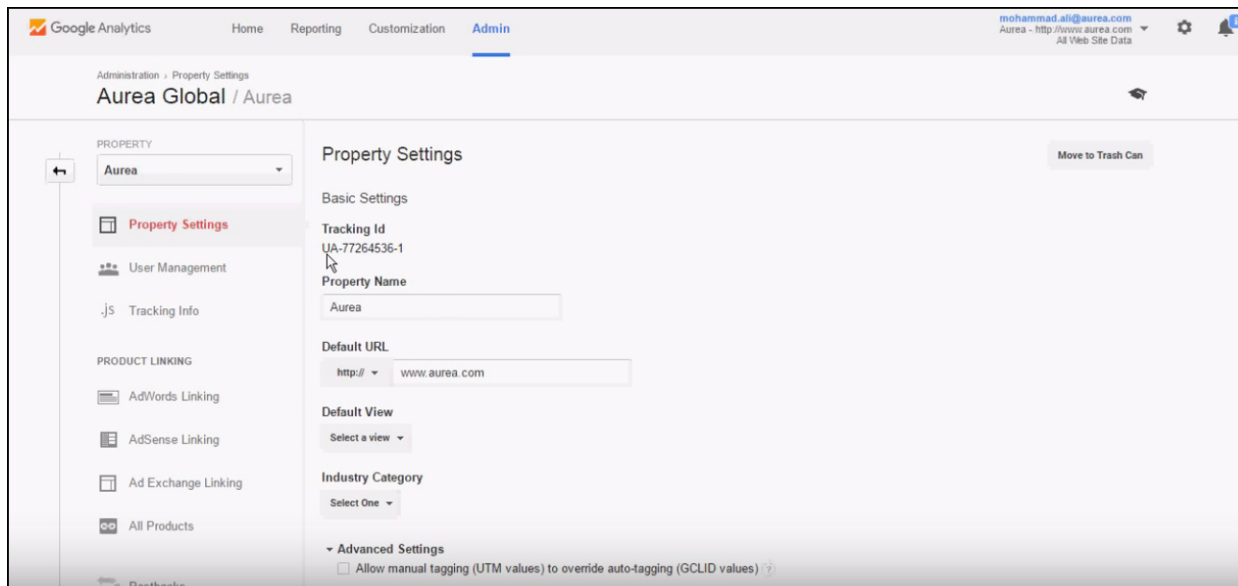
---

2. Click **Get Tracking ID**. The new tracking id for the new property is generated. The new property is now visible under the drop down list under the Property column in the admin page.



### Edit the property settings for the new property

1. Select the new property name from the drop down list in the PROPERTY Column.
2. Click **Property Settings**. The **Property Settings** page displays.

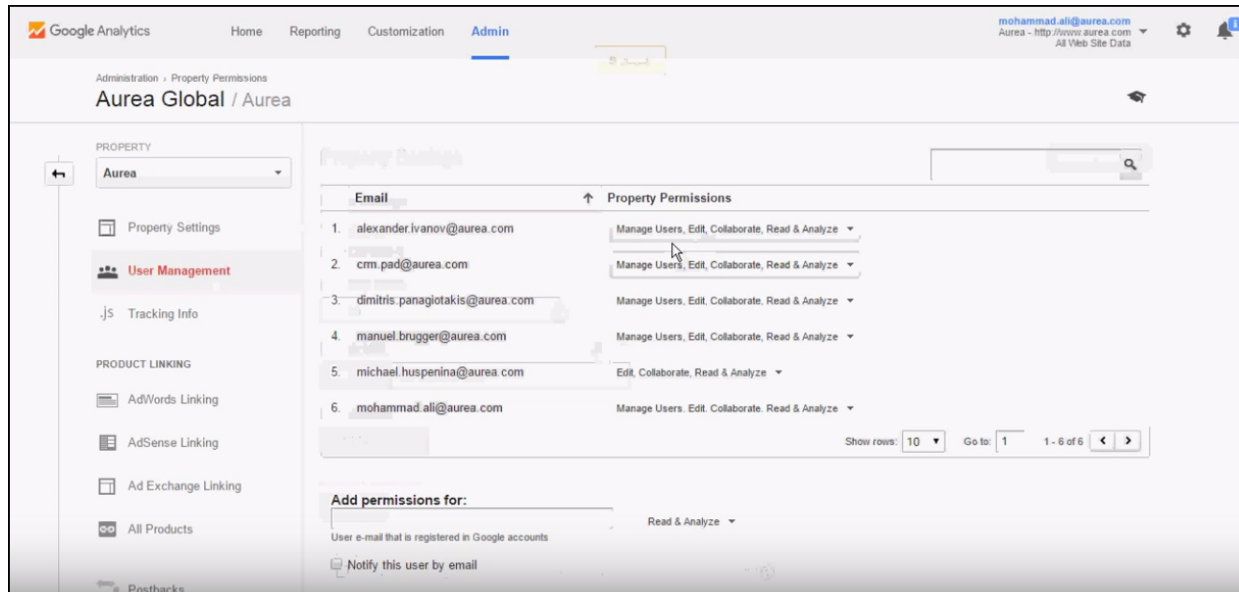


Configure the basic and advanced properties listed on this page as required.

- Click **Save**. The Administrator page opens again.

## User Management settings

- Click User Management node in the PROPERTY column. You can add users and assign them access permissions from this page. Click Save.



## Configure and retrieve tracking information

1. Expand the Tracking info node in the PROPERTY column. Click Tracking Code. This page provides the following:
  - a. You can retrieve the **Tracking ID** for the Aurea CRM tracking instance. You have to configure this value in the Aurea CRM Designer.
  - b. You can get the Tracking code that you have to embed in the Aurea CRM Web pages that you want to track.

---

**Note:** Refer to the Google Analytics Help Center for more information.

---

## Configuring Custom Dimensions

Learn to configure Custom Dimensions in Google Analytics.

A Dimension in Google Analytics is an artifact that captures data accessed on your web application. Google Analytics provides you with default dimensions to capture and analyze most data. But it does not capture Aurea CRM specific data. In order to capture CRM data, for example Infoarea of the data retrieved by your users, you have to provide your own custom dimensions. Aurea provides a few custom dimensions to capture some of the Aurea CRM data. The following custom dimensions are provided by Aurea:

- Infoarea
- View
- CRMVersion
- InstanceID
- UserID

To add more custom dimensions specific to your requirements, please contact Aurea support ([email:support@aura.com](mailto:support@aura.com)).

To use the custom dimensions provided by Aurea CRM:

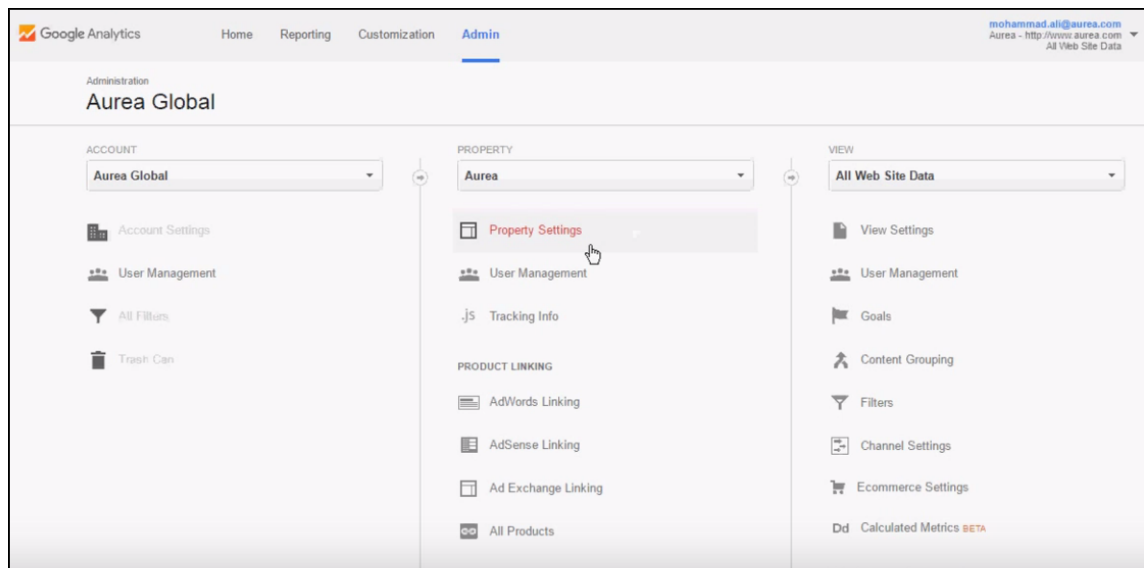
1. Create the custom dimensions in the tracker.
2. Configure the custom dimensions in the CRM Designer to allow Google Analytics to track the data for the custom dimensions.

### Creating Custom Dimensions in the Tracker

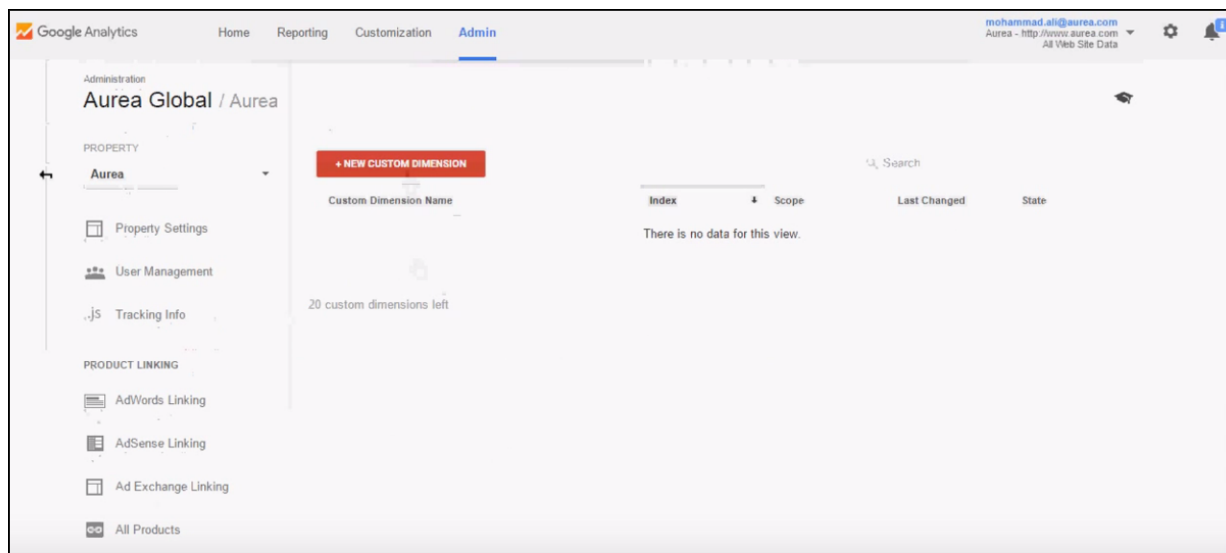
To configure a custom dimension follow these steps:

1. Open the Google Analytics tracker page for your organization.
2. Select the **Admin** tab. The Administrator page displays.

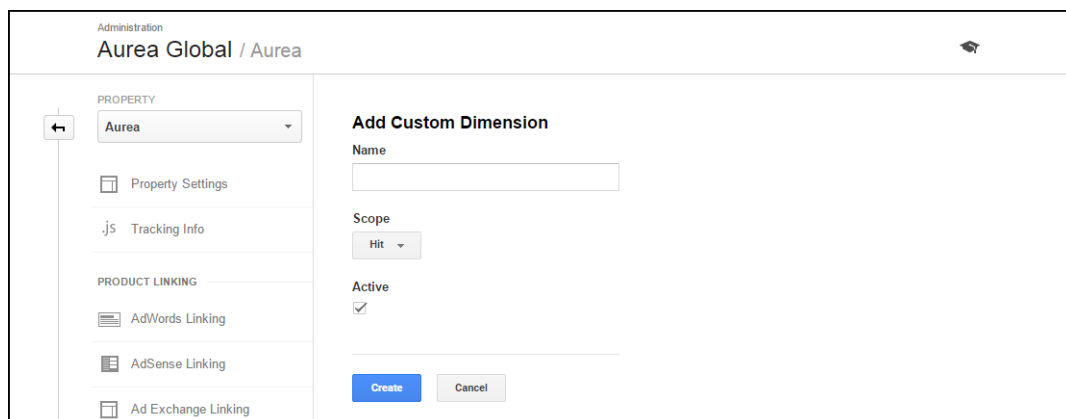




3. Expand the **Custom Definitions** node in PROPERTY column.
4. Click **Custom Dimensions**. The **Custom Dimensions** page displays.



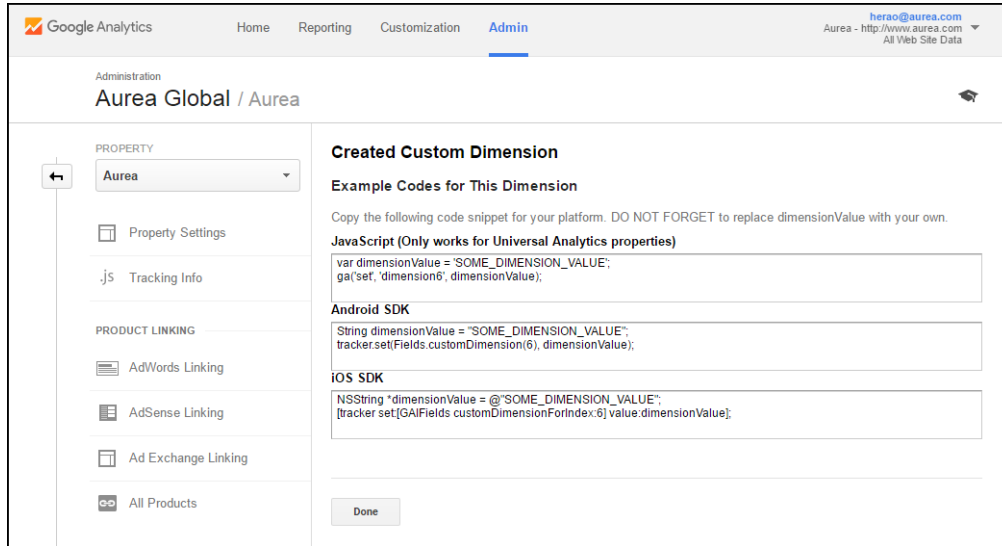
5. Click **New Custom Dimension**. The **Add New Custom Dimension** page displays.



Enter the following information:

- Name: Enter a name for the custom dimension.
- Scope: Specify the level, when you want to start tracking.
- Active: To activate and start collecting data.

6. Click **Create**. The Created Custom Dimensions page displays.

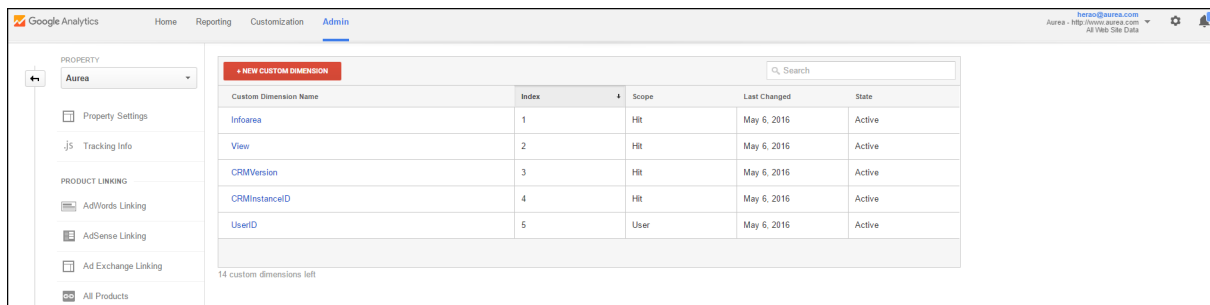


Code snippets to be used in the CRM code to capture the required CRM data is provided.

7. Click **Done**. The Custom dimension is created.

### Configuring Custom Dimensions in the CRM Designer

After configuring all the Aurea supplied custom dimensions in the Google Analytics tracker in your organization’s account, you have to set them up for tracking in the CRM Designer. The following screenshot shows the custom dimensions page in the Google tracker.



Each of the custom dimension configured has an index value associated, shown in the table above. To configure the custom dimension in CRM Designer, you have to set the value of the corresponding web configuration parameter for the custom dimension in the format `dimension<index>`. The web configuration parameters for a custom dimension is of the format `GoogleAnalytics.CustomDimension.<custom dimension name>`

For example, in the screenshot above, the custom dimension Infoarea has the index value of 1. The corresponding web configuration parameter for custom dimension, Infoarea is `GoogleAnalytics.CustomDimension.Infoarea` and its value is set to `dimension1`.

The following screenshot shows the web configuration parameter configured for the custom dimensions shown in the Google Tracker screenshot above. For more information, see [Google Analytics Parameters](#) on page 118

Parameter Name	Value
GoogleAnalytics.CRMInstanceID	123456789
GoogleAnalytics.CustomDimension.CRMVersion	dimension2
GoogleAnalytics.CustomDimension.Infoarea	dimension1
GoogleAnalytics.CustomDimension.InstanceID	dimension4
GoogleAnalytics.CustomDimension.UserID	dimension5
GoogleAnalytics.CustomDimension.View	dimension2
GoogleAnalytics.CustomerTrackerID	UA-79872830-1

## Creating a new Custom Report using Custom Dimensions

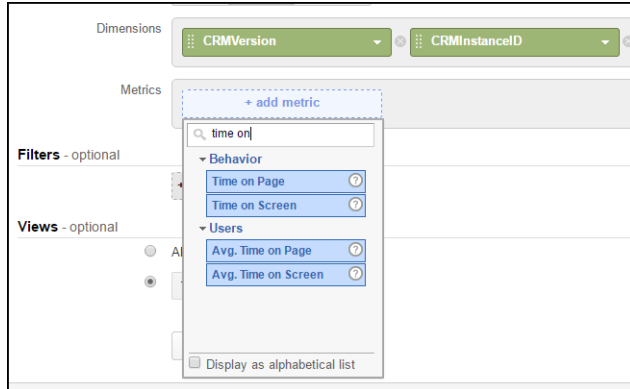
Google Analytics allows you to create custom reports. You can use the custom dimensions in these reports.

To create a custom report, perform the following steps:

1. Open the Google Analytics tracker page for your organization.
2. Select the **Customization** tab. The **Customization** page displays.
3. Click **New Custom Report**. The Create Custom Report page displays.

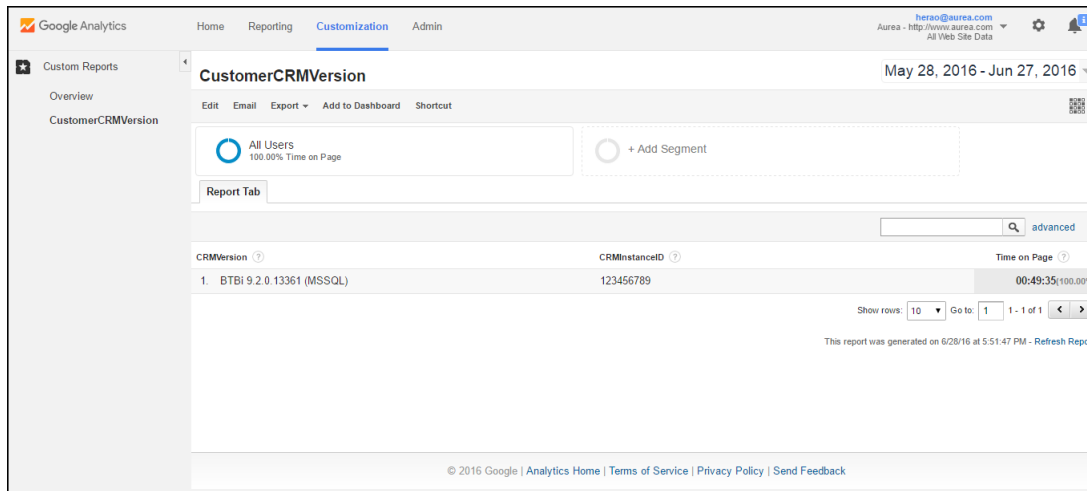
4. Click the **Flat Table** button.
5. Click **+ add dimensions**. The drop down list opens.
6. Expand the **Custom Dimension** node and select **CRMInstanceID**.
7. Repeat steps 5 and 6 and add the custom dimension **CRMVersion**.

8. Click **+ add metric**. The drop down list opens.

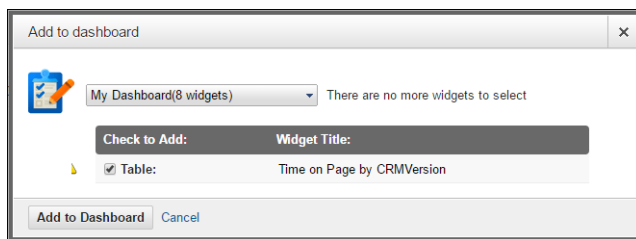


9. Enter search for and select the **Time on Page** metric. The metric is added to the page.

10. Click **Save**. The custom report is created.

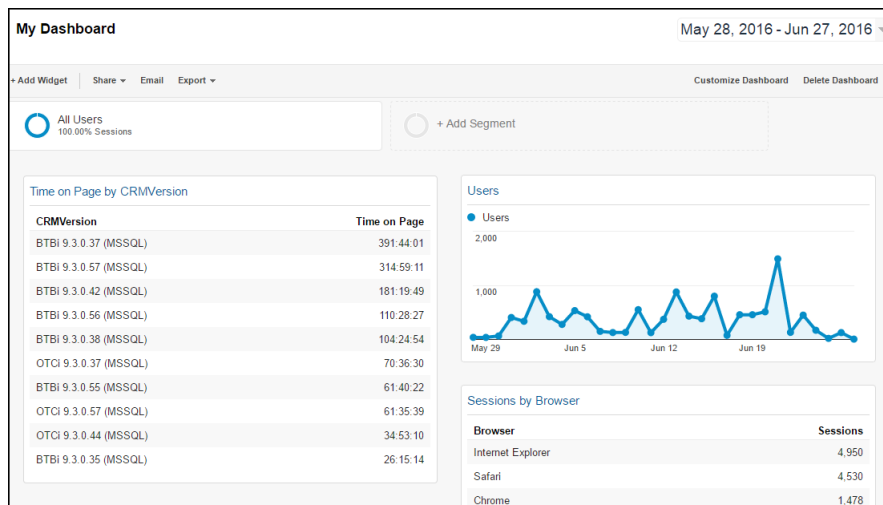


11. Click **Add to Dashboard**. The **Add to Dashboard** window pops up.



12. Select the check box on the table row.

13. Click **Add to Dashboard**. The report is added to the specified dashboard in table format. See the dashboard screenshot below. The total time spent on all the pages, by all users of each of the different CRM server versions are displayed in the table.



## Aurea Campaign Manager (formerly Lyris HQ) Integration

Learn to integrate Aurea CRM with Aurea Campaign Manager.

As part of a campaign, the marketing activities can be carried out using multiple channels, like letters, fax, telephone, and email. Generally the email channel is the most preferred channel for marketing activities. While Aurea CRM provides its own email channel, it does not provide all the capabilities of a dedicated Email marketing solution. To provide a sophisticated email marketing channel, Aurea CRM is integrated with Aurea Campaign Manager (ACM) (formerly Lyris HQ), a full-fledged Email marketing solution. Aurea CRM Enterprise edition users can use ACM to handle their email marketing activities and take full advantage of its sophisticated email marketing capabilities.

---

**Note:** Aurea CRM Enterprise license enables you to access ACM features. In order to use ACM features you require a separate license for ACM.

---

Aurea Campaign Manager (ACM) notifies Aurea CRM of the actions it performs by using the event flow mechanism to send event notifications. Events are generated by ACM, when it performs some action on Contacts, Mailing lists, Segments, and Messages. Events are also generated by user actions, like opening an email message or clicking a response link provided within the email message. The event notifications generated by the Campaign Manager is placed in event queues in the ACM Sync Middleware and received by the CRM. Server listening for event notifications. Multiple instances of the event queues can be configured to monitor each of the event queues in the Sync Middleware. The Sync Middleware supports eight queues and each of these queues can be configured to notify specific event types.

**Note:** To enable ACM you have to set the `Lyris.Enabled` parameter to `true`. This parameter should be set in the configuration that corresponds to the CRM.Server user, and should be set in the `users.xml` file. The CRM Server Plugin for SyncMiddleware uses the above credentials to access the SyncMiddleware credentials configured in the CRM.Designer database. For more information on setting the user details, see [Configuring CRM Server to listen to the SyncMiddleWare Message Queues](#) on page 672.

## ACM Sync Middleware Overview

CRM.Server has a plugin for the ACM Sync Middleware that allows it to listen to event notifications posted on the event queues hosted by the ACM Sync Middleware.

The Sync Middleware URI and the authentication details are configured in the following web configuration parameters in the designer: `Lyris.SyncMiddlewareUri`; `Lyris.SyncMiddlewareUsername`; and `Lyris.SyncMiddlewarePassword`. For more information on these parameters, see [Lyris Parameters](#) on page 127.

The ACM Sync Middleware uses the Java Messaging System to publish event messages. You can configure multiple instances of CRM.Server to listen to one or more event queues. There are 8 event queues maintained by the ACM Sync Middleware. The event queues handled by the CRM.Server instance is configured using the `<update.PlugIns.Lyris.SyncMiddleware.Consumer>` element in the `settings.server.xml` file for the CRM.Server instance. The `<consume>` child node contains the name of the event queue the CRM.Server instance listens to.

The event queue name should have the following format:

```
<ORGANISATION UID>.VirtualTopic.<EVENT NAME>
```

For example, the following screenshot shows the configured event queues:

```
<!-- Lyris Sync Middleware queues consumers -->
<update.PlugIns.Lyris.SyncMiddleware.Consumer>
  <consume>265895cf- ..... 3779a2.VirtualTopic.Contact.Updated</consume>
  <consume>265895cf- ..... 3779a2.VirtualTopic.Contact.Upload.Status</consume>
  <consume>265895cf- ..... 3779a2.VirtualTopic.MailingList.Updated</consume>
  <consume>265895cf- ..... 3779a2.VirtualTopic.Message.Status</consume>
  <consume>265895cf- ..... 3779a2.VirtualTopic.Message.Deleted</consume>
  <consume>265895cf- ..... 3779a2.VirtualTopic.Message.Updated</consume>
  <consume>265895cf- ..... 3779a2.VirtualTopic.Segment.Generated</consume>
</update.PlugIns.Lyris.SyncMiddleware.Consumer>
```

Where the `Organization UID` has the value `26589cf-... -377pa2`. To get the `<Organization UID>` for your organization, see [Ordering Access to the ACM SyncMiddleware](#).

```
<consume>vm0000a88a.VirtualTopic.Contact.Updated</consume>
```

The following event names are supported:

- **Contact.Updated**—for example, `Consumer.Lyris_EndPoint.VirtualTopic.Contact.Updated`.
- **Contact.Upload.Status**—for example, `Consumer.Lyris_EndPoint.VirtualTopic.Contact.Upload.Status`.
- **MailingList.Updated**—for example, `Consumer.Lyris_EndPoint.VirtualTopic.MailingList.Updated`.
- **Message.Status**—for example, `Consumer.Lyris_EndPoint.VirtualTopic.Message.Status`.
- **Message.Deleted**—for example, `Consumer.Lyris_EndPoint.VirtualTopic.Message.Deleted`.
- **Message.Updated**—for example, `Consumer.Lyris_EndPoint.VirtualTopic.Message.Updated`.
- **Segment.Generated**—for example, `Consumer.Lyris_EndPoint.VirtualTopic.Segment.Generated`.

**Important:** Always ensure that a queue name is unique for each CRM.Server instance. If more than one instance of the CRM.Server is configured to monitor the same queue, then a message retrieved by one instance is not available to the other instance and results in inconsistent results, missed events etc.

The following configuration settings shows the CRM.Server instance listening to different message queues (in the red box).

```

182
183 <!-- settings of update.server -->
184 <update.server>
185 <!-- A list of all plugins that will be loaded by update.server -->
186 <Plugins>
187 <PlugIn type="update.Crm.Base.PlugIn,update.Crm.Base" />
188 <PlugIn type="update.Crm.Server.PlugIn,update.Crm.Server">
189 <!--Maximum number of jobs to be executed at the same time,if blank then 1.5 * Logical CPU-->
190 <MaxConcurrentJobs></MaxConcurrentJobs>
191 </PlugIn>
192 <PlugIn type="update.Crm.Services.PlugIn,update.Crm.Services" />
193 <PlugIn type="update.Configuration.PlugIn,update.Configuration" />
194 <PlugIn type="update.Crm.PlugIn,update.Crm" />
195 <PlugIn type="update.PlugIns.Lyris.SyncMiddleware.Consumer.PlugIn,update.PlugIns.Lyris.SyncMiddleware.Consumer"/>
196 </Plugins>
197 </update.server>
198
199 <!-- Lyris Sync Middleware queues consumers -->
200 <update.PlugIns.Lyris.SyncMiddleware.Consumer>
201 <consume>265895cf- ..... 3779a2.VirtualTopic.Contact.Updated</consume>
202 <consume>265895cf- ..... 3779a2.VirtualTopic.Contact.Upload.Status</consume>
203 <consume>265895cf- ..... 3779a2.VirtualTopic.MailingList.Updated</consume>
204 <consume>265895cf- ..... 3779a2.VirtualTopic.Message.Status</consume>
205 <consume>265895cf- ..... 3779a2.VirtualTopic.Message.Deleted</consume>
206 <consume>265895cf- ..... 3779a2.VirtualTopic.Message.Updated</consume>
207 <consume>265895cf- ..... 3779a2.VirtualTopic.Segment.Generated</consume>
208 </update.PlugIns.Lyris.SyncMiddleware.Consumer>
209
210 <!-- settings of update.configuration-->
211 <update.configuration>
212 <OleDbDsn>Provider=SQLOLEDB;Data Source=MGRABKA-PC\SQLEXPRESS;Initial Catalog=crm-lyris-final-demo.bb.10.designer;UID=mm; PWD=mm</OleDbDsn>
213 <Vertical>BB</Vertical>
214 </update.configuration>
215 </update>

```

**Note:** The CRM.Server plugin configuration in the red ellipse, shows the plugin type settings required for the ACM Sync Middleware.

## Ordering Access to the ACM SyncMiddleware

You can place your request for the ACM SyncMiddleWare details from <https://support.aurea.com/>. Ensure that you provide the Aurea support group with the following information:

- Your preferred Lyris Organization UID or proper Lyris HQ Organization name.

You should receive the following information from support on the ACM SyncMiddleWare and Lyris API:

- `Organization UID`
- `Lyris.SyncMiddlewareUri`
- `Lyris.SyncMiddlewareUsername`
- `Lyris.SyncMiddlewarePassword`
- `Lyris.Host`
- `Lyris.APIPassword`

## Configuring Bi-directional Communication between ACRM and ACM

Learn to set up bi-directional Communication between ACRM and ACM.

To setup bi-directional communication between Aurea Campaign Manager and Aurea CRM, you have to configure the following:

1. Configure Aurea CRM with the ACM SyncMiddleware information, to enable the CRM Server to listen to message events from ACM, see [Ordering Access to the ACM SyncMiddleware](#).
2. Configure the ACM Web configuration parameters in the CRM designer to enable ACRM to communicate with Lyris using API calls. Lyris Parameter should be correctly configured in the CRM Sub-configuration created in the CRM.Designer, and are pointing to the correct ACM. For more information on configuring Lyris Parameters in the CRM.Designer, see [Configuring Lyris HQ](#) on page 676.

## Configuring CRM Server to listen to the SyncMiddleWare Message Queues

Learn to set-up CRM Server to listen to the SyncMiddleWare Message Queues.

Before you perform the setup, ensure the following conditions are met:

1. ACRM Enterprise version should be 10.2.0 or higher.
2. CRM Server should be up and running.
3. Ensure that you have all the ACM SyncMiddleware details with you. For more information on ordering the access to the SyncMiddleWare, see [Ordering Access to the ACM SyncMiddleware](#).



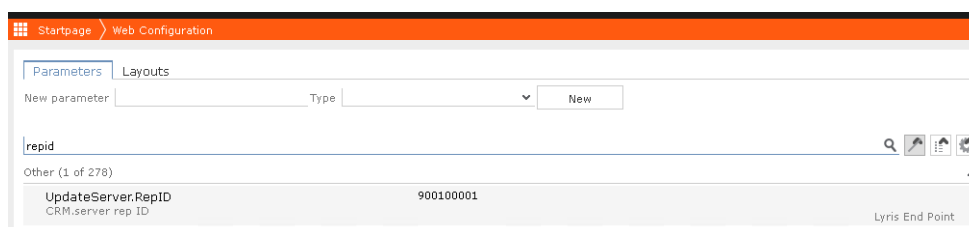
Perform the following steps:

1. Configure the CRM Server user in `user.xml` located at `<CRM.Web Vertical Root>\web\system \settings` directory. The CRM Server user should have access to the CRM.server module and the Configuration module. The user should also be defined as CRM.server Rep ID in web configuration to ensure that Todos are assigned to the user. See the screenshot below, where `type="Server"`:

```
<?xml version="1.0"?>
<Users>
  <!--
  |
  | This file is usually not edited manually!
  | Use "update.users.exe" to add/edit the
  | users.
  | For security reasons it is recommended
  | to encrypt this file.
  -->
  <User type="Default">
    <Name>WWW</Name>
    <Password></Password>
  </User>
  <User type="Special">
    <Name>SU</Name>
    <Password></Password>
  </User>
  <!-- enter your CRM.server user here -->
  <User type="Server">
    <Name>SU</Name>
    <Password></Password>
  </User>
</Users>
```

**Note:** It is not mandatory to use SU (super user). Any user having access to CRM.server module account to the Configuration module can be added here. The user should also be defined as a CRM.server Rep ID in CRM.Web configuration.

2. Set the CRM Server users Rep ID in web configuration parameter `UpdateServer.RepID`. See the screenshot below:



For more information, see [UpdateServer.RepID](#).

3. Update the `<appSettings>` element in the `update.Server.exe.config` file located at `<CRM.Web Vertical Root>\web\system\settings` directory to point to the correct path of file `settings.server.xml`.

```
<?xml version="1.0" ?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.0"/>
  </startup>


  <appSettings>
    <!-- the hosting environment will look for its settings in this file: -->
    <add key="settingsFile" value="..\system\settings\settings.server.xml" />
  </appSettings>
</configuration>
```

4. The file `settings.server.xml` should have the following settings defined.

**Note:** These settings are predefined in a fresh installation of 10.2.20 and higher, but needs to be defined if you are upgrading from versions prior to 10.2.20.


a. The database connection string should be properly specified.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE us [
  <ENTITY vertical "bb">
  <ENTITY VERTICAL "BB">
  <ENTITY Vertical "Bb">
  <ENTITY logfile "server.log">
  <ENTITY dataPath "..\..\web.data">
  <ENTITY designerDsn "Provider=SQL Server Native Client 11.0;Data Source=LOCALHOST;Initial Catalog=CRMdesigner;UID=CRM; PWD=Update1!">
]>
```



b. The `update.server` element should include the `SyncMiddleware` plugin.

```
<!-- settings of update.server -->
<update.server>
  <!-- A list of all plugins that will be loaded by update.server -->
  <PlugIns>
    <Plugin type="update.Crm.Base.Plugin,update.Crm.Base" />
    <Plugin type="update.Crm.Server.Plugin,update.Crm.Server">
      <!--Maximum number of jobs to be executed at the same time,if blank then 1.5 * Logical CPU-->
      <MaxConcurrentJobs></MaxConcurrentJobs>
    </Plugin>
    <Plugin type="update.Crm.Services.Plugin,update.Crm.Services" />
    <Plugin type="update.Configuration.Plugin,update.Configuration" />
    <Plugin type="update.Crm.Plugin,update.Crm" />
    <Plugin type="update.PlugIns.Lyris.SyncMiddleware.Consumer.Plugin,update.PlugIns.Lyris.SyncMiddleware.Consumer"/>
  </PlugIns>
</update.server>
```



c. The `< update.PlugIns.Lyris.SyncMiddleware.Consumer >` should be defined with the list of consumer queues which are created in ActiveMQ (SyncMiddleware). The name of queues should be unique to every installation. In the following example, the Organization UID value is `26589cf- ...377pa2`.

```
<!-- Lyris Sync Middleware queues consumers -->
<update.PlugIns.Lyris.SyncMiddleware.Consumer>
  <consume>265895cf- ..... 3779a2.VirtualTopic.Contact.Updated</consume>
  <consume>265895cf- ..... 3779a2.VirtualTopic.Contact.Upload.Status</consume>
  <consume>265895cf- ..... 3779a2.VirtualTopic.MailingList.Updated</consume>
  <consume>265895cf- ..... 3779a2.VirtualTopic.Message.Status</consume>
  <consume>265895cf- ..... 3779a2.VirtualTopic.Message.Deleted</consume>
  <consume>265895cf- ..... 3779a2.VirtualTopic.Message.Updated</consume>
  <consume>265895cf- ..... 3779a2.VirtualTopic.Segment.Generated</consume>
</update.PlugIns.Lyris.SyncMiddleware.Consumer>
```

d. Enter the settings for the `< update.configuration >` element:

```

<!-- settings of update.configuration-->
<update.configuration>
  <OleDbDsn>&designerDsn;</OleDbDsn>
  <Vertical>&VERTICAL;</Vertical>
</update.configuration>

```

e. Set up update.server as a windows service.

a. Open a command prompt with administrative privileges and change directory to <CRM.Web Vertical Root>\web\bin.

b. Run the following command:

```
update.server.exe --install --name "update.server"
```

c. The service named update.server should appear in the list of windows services.

d. Start the service. If all the settings are configured properly the Status should change to 'Running'.

Themes	Provides us...	Running	Automatic	Local System...
Thread Ordering Server	Provides or...	Manual	Manual	Local Service
Time Broker	Coordinates...	Running	Manual (Trig...	Local Service
Touch Keyboard and Hand...	Enables Tou...	Running	Manual (Trig...	Local System...
update.server		Running	Automatic	Local System...
UPnP Device Host	Allows UPn...	Disabled	Disabled	Local Service
User Access Logging Service	This service ...	Running	Automatic (D...	Local System...
User Profile Service	This service ...	Running	Automatic	Local System...

f. To configure the server error log files perform the following steps:

a. To troubleshoot errors in the server setup, set the `loglevel` attribute to `Debug` in the `<Logging>` element in the `settings.server.xml` files. See the screenshot below:

```

<!-- settings of update.lib -->
<update.lib>
  <!--
  Logging:
  Define the ...
  (-) The file name(s) of the log files(s)
  (-) The verbosity (logLevel)
  (-) Filters
  (-) how the log values are formatted
  -->
  <Logging>
    <!-- file -->
    <Channel
      logLevel="Debug"
      writerName="DefaultLog"
      formatterName="Default" />
    <Writer
      filePath="%dataPath%\log\%logFile;" name="DefaultLog"
      maxSizeKB="8192"
      closeOnIdleTimeout="5"
      type="update.Lib.Logging.Writers.FileLogWriter,update.Lib" />

```

b. The following log entry in the log file is a confirmation that the value set in the web configuration parameter `Lyris.SyncMiddleUri` is working.

```

2017.09.14;05:42:53; 515;update.PlugIns.Lyris.SyncMiddleWare.Consumer;Debug ; 104038400; 3220; ;Lyri
tcp://52.204.45.159:61616.

```

**Note:** The URI value is read directly from the **Lyriss.SyncMiddleUri** parameter and if it is changed, restart the Update Server to use the new Sync middleware URI value.

## Configuring Lyriss HQ

Learn to set up Lyriss HQ for Aurea CRM.

Aurea CRM communicates with Lyriss using the Lyriss API. And CRM users can use the Lyriss Web interface to view the mailing lists, messages, and contacts (target group members). This requires that you set up the Lyriss URL templates and the authentication parameters required by Lyriss API. You can set up language in which the demographic information of target group members is created in Lyriss.

Some of the parameter and their use to communicate with Lyriss are described in this topic. You can find all the web configuration parameters for Lyriss and their descriptions in [Lyriss Parameters](#) on page 127.

## Getting Lyriss Web Configuration Parameter Values from Lyriss HQ

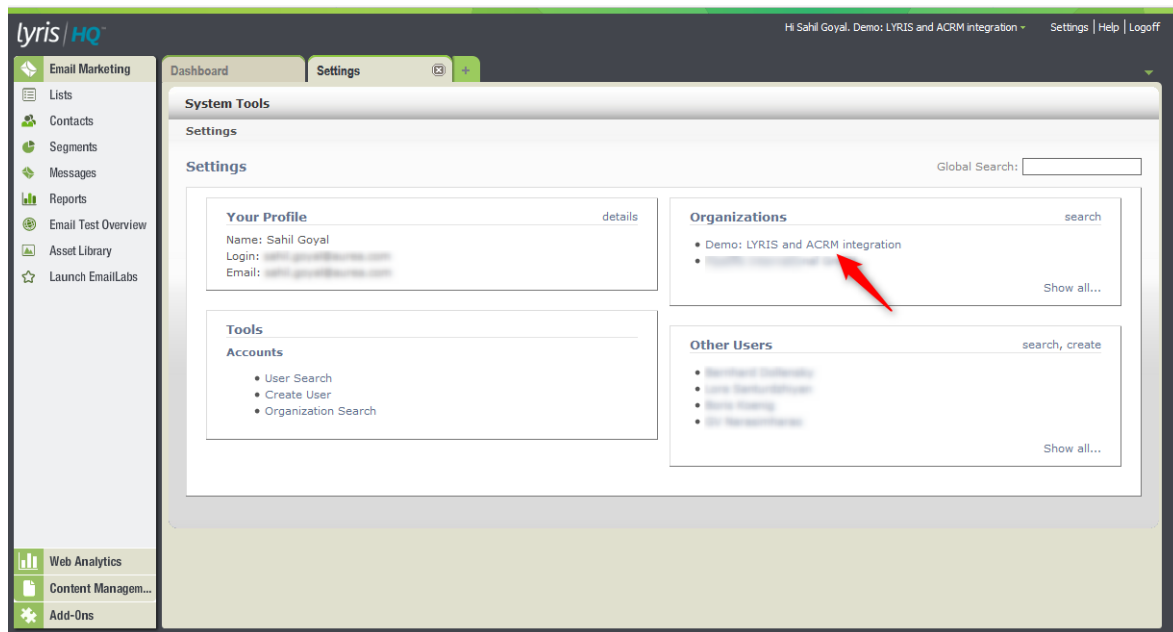
Contact Aurea support (support.aurea.com) for setting up your Organizational account with the Email ID for the owner of the account and Email IDs of additional users to be added to your organization.

Before you configure the ACRM.Web configuration parameters for Lyriss HQ, you need to retrieve the retrieve your organizations information configured on Lyriss HQ. To retrieve them perform the following steps:

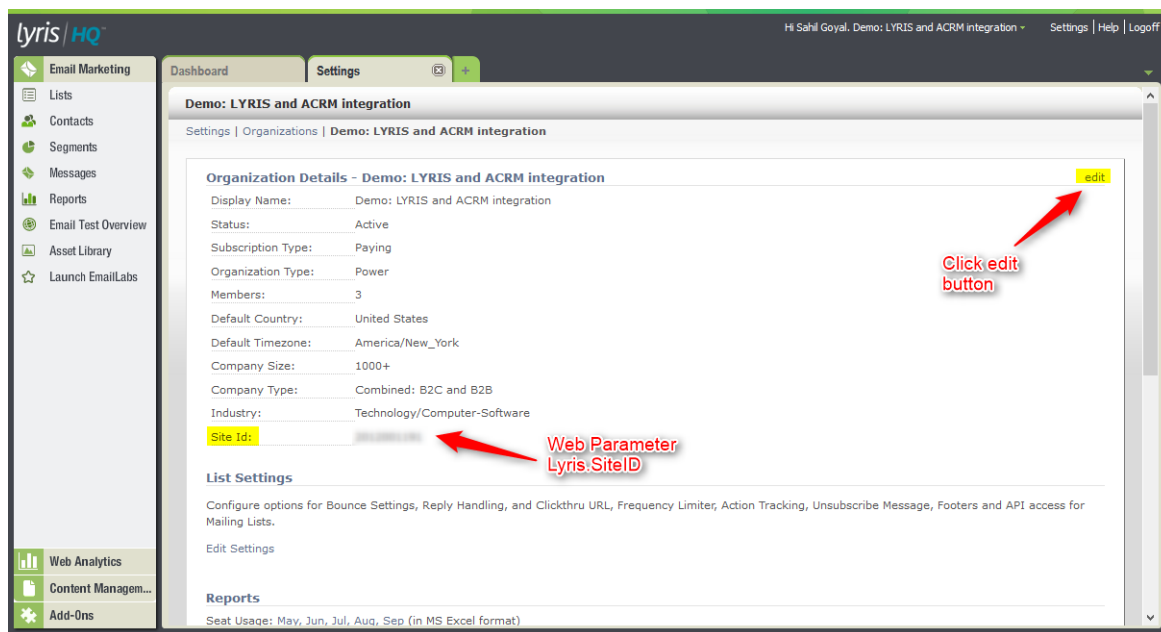
1. Login to your Lyriss HQ Account at [www.lyrisshq.net](http://www.lyrisshq.net).
2. Use the **Settings** option as shown in the screenshot below to view the organization's settings.

The screenshot shows the Lyriss HQ dashboard. At the top right, there is a user profile 'Hi Sahil Goyal, Demo: LYRIS and ACRM integration' and a 'Settings' link. A red arrow points to the 'Organisation' dropdown menu in the top center. Another red arrow points to the 'Settings' link in the top right. The dashboard includes a left sidebar with navigation options like 'Email Marketing', 'Lists', 'Contacts', 'Segments', 'Messages', 'Reports', 'Email Test Overview', 'Asset Library', and 'Launch Email Labs'. The main content area has a 'Dashboard' section with 'Reports' (Most Recently Sent Messages) and a calendar for 'September 2017'. On the right, there are 'My Stats' (Website Statistics, Mailing List Statistics) and a 'Social Media Panel'.

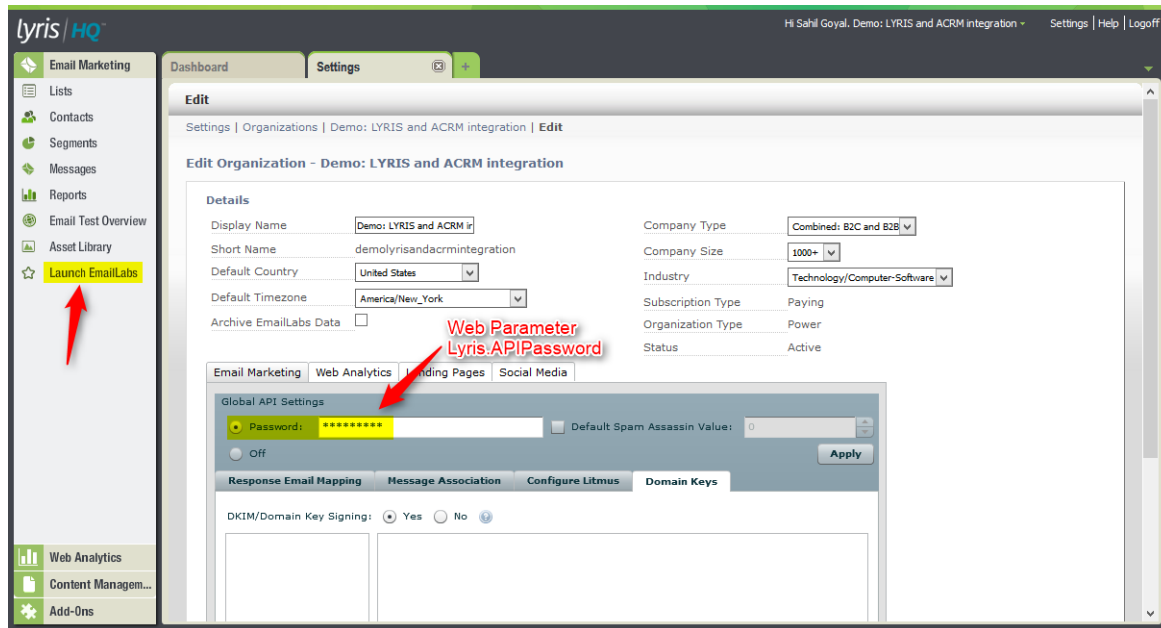
3. Select your organization from the list in the **Organizations** pane.



4. The `Lyris.SiteID` configuration parameter value can be found in the **Site ID** field in the **Organizational Details** windows.



5. Click **Edit** on the top right corner of the **Organizational Details** window. You can set the API Password for the organization in the **Edit Organization** pane. This value is provided in the web configuration parameter `Lyris.APIPassword`.



6. Click **Launch EmailLabs** in the **Email Marketing** pane to the left of the page. The **EmailLabs** page displays. The domain name displayed in the URL for the page is the value that is required to set the web configuration parameter **Lyris.Host**.

In the screenshot below, the domain value is `elabs12.com`.

Web Parameter  
Lyris.Host = elabs12.com

Account Home

Mailing Lists Archived Lists Recent Messages

New List

Search

Find Name beginning with Search

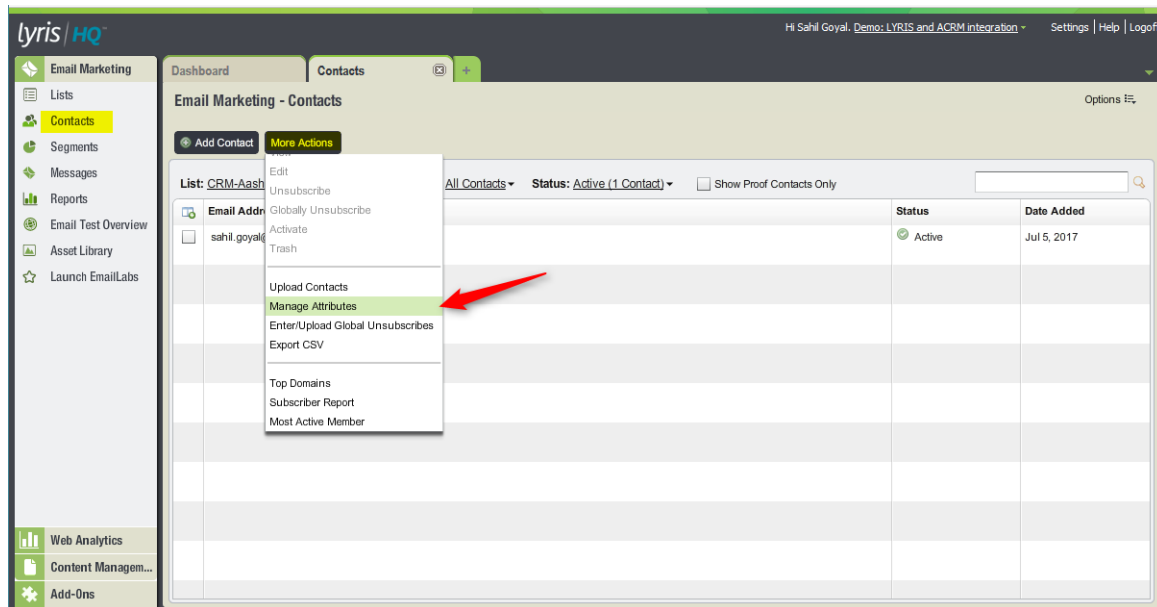
Active 1 to 10 of 20

Name	Members	Last Message	Action
CRM-Pongal - Survey	2	Jul 7, 2017	Delete
CRM-Pongal - Christmas Card	2	Jul 6, 2017	Delete
CRM-Pongal - Quick Campaign	2	Jul 6, 2017	Delete
CRM-Pongal - Product Information (2)	2	Jul 6, 2017	Delete
CRM-Demo Test - email	3	Jul 6, 2017	Delete
CRM-Pongal - Product Information	2	Jul 5, 2017	Delete
CRM-Aashadam Sale - Product Information	1	Jul 5, 2017	Delete
CRM-Lyris Test - Event	2	Jul 2, 2017	Delete
NR	1	Jun 23, 2017	Delete
CRM-Template: Event - Invitation 007	1		Delete

**Note:** The value of the Lyris parameter `Lyris.Host` is used to configure the values of the following Lyris web configuration parameters:

- `Lyris.MailingListUrlTemplate`
- `Lyris.MessageUrlTemplate`
- `Lyris.SegmentUrlTemplate`

7. Set the value of the web configuration parameter, `Lyris.ActivityExecutionIdDemographicId`. Click **Contacts** in the left pane. Then select **More Actions > Manage Attributes** in the resulting **Email Marketing – Contacts** page.



8. In the resulting **Attribute Management** dialog, create a new attribute with the name `ActivityExecutionIdDemographicId` of type `Multiple Select List`. The ID generated for this attribute should be added to the `Lyris.ActivityExecutionIdDemographicId` web configuration parameter.
9. Edit the attribute `Lyris.TargetGroupTransferMaxChunkSize`. It determines the maximum number of contacts that can be transferred to a mailing list created in Lyris. Set it to a large number if the target group number is more.

### Configuring Lyris, Lyris Host, and Lyris API details

In order to use Lyris as an email marketing tool it has to be first enabled using the `Lyris.Enabled` web configuration parameter. Setting this to true is the first configuration step to start using Lyris from Aurea CRM.

All communication with Lyris is done using the Lyris API. The host address used by Aurea CRM to communicate with Lyris API is provided using the `Lyris.Host` parameter. All API calls are made against the URL format. `Lyris.Host` should be either an IP address or a fully qualified name of the machine hosting Lyris HQ API.

Authentication password for a Lyris API connection is provided in the parameter `Lyris.APIPassword`.

All API requests passed by Aurea CRM to the Lyris host needs to carry a site id. This is a Lyris requirement and is configured in the parameter `Lyris.SiteID`.

### Configuring Mailing list name prefix

To transfer a marketing activity to Lyris, a mailing list and a message is created by Lyris for that marketing activity. The `Lyris.SuggestedMailingListNamePrefix` parameter takes a prefix value that is used by Aurea CRM to suggest a name for the mailing list. It is displayed when the user chooses to create a new mailing list. Custom configurations for each user of CRM system can also be used to provide specific prefix to differentiate between CRM users. The user can overwrite the suggested name along with the prefix during the creation of a new mailing list.



## Configuring Lyris URL Templates

When a user transfers a marketing activity to Lyris, a mailing list and a message template are created on Lyris. Different marketing activities can use the same mailing lists, to differentiate between them, Lyris generates a segment for each marketing activity. This helps it to associate the target groups associated with the same mailing list used by different marketing activities.

On successful creation of these entities on Lyris, Aurea CRM creates the URLs to access these instances on Lyris. The user can click on these links to navigate to the instances on Lyris. To access the mailing lists, messages and segments, Lyris expects these URL to have a certain format.

The parameters `Lyris.MailingListURLTemplate`, `Lyris.MessageURLTemplate`, and `Lyris.SegmentUrlTemplate` contain the URL templates used by Aurea CRM to generate the URLs to access the mailing list, message, and segment instances on Lyris.

The following is a sample of the mailing list URL.

```
https://ip-172-30-90-193.ec2.internal/v3/mailing_list.html?
NEXTPAGE=mailing_list_view&setmlcookie=on&mlid={LyrisMailingListId}
```

The id generated by Lyris when it creates the mailing list is returned to Aurea.CRM and inserted in the `{LyrisMailingListId}` placeholder.

Similarly, the following is a sample of the message URL,

```
https://ip-172-30-90-193.ec2.internal/v3/mailing_list_message_view.html?
setmlcookie=on&mlid={LyrisMailingListId}&mid={LyrisMessageId}
```

The message id as returned by Lyris API on successful creation of a message during marketing activity transfer to Lyris, is inserted in the `{LyrisMessageId}` placeholder.

## Configuring target group creation and scheduling

When the target group for a marketing activity is transferred to Lyris, the demographics created for certain catalog values like country, date format, and greetings are created according to the demographics language specified in the `Lyris.DemographicOptionsLanguage` parameter. Valid values are: ger, eng, fre, spa, por, nld, dan, ita, cze, hun, slk, pol, gre, ukr, sln, rus, swe, fin, nor, tur, cro, ser, ron, jpn, chn, kor, bul, tha, and alb. For example, if you set the value to "ger" the demographic created for the CRM field Country in Lyris has **Deutschland**.

When CRM requests Lyris to create the required demographics for the CRM fields in Lyris, it has to provide the demographics ID to the Lyris API, this value should be provided in the parameter **Lyris.ActivityExecutionDemographicId**.

To ensure that the transfer of target groups to the Lyris server happens successfully, CRM schedules the transfer of a certain number of items in the target group at a time. The value of the chunk to be transferred at a time can be specified in

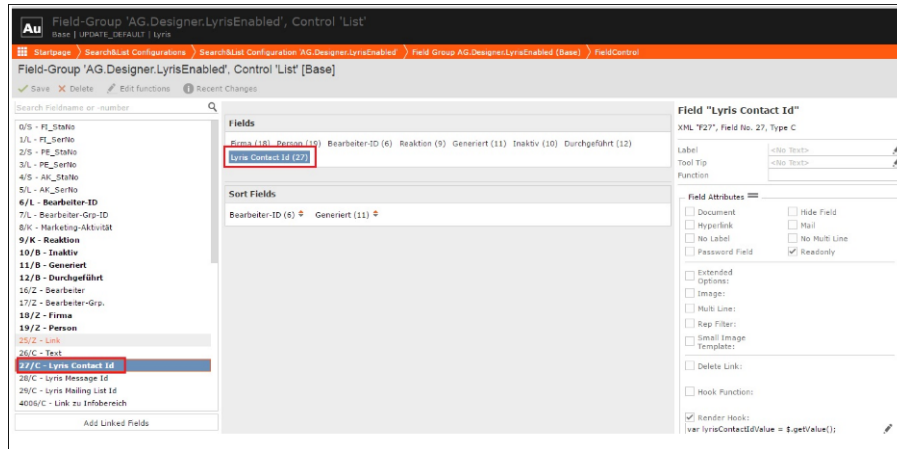
```
Lyris.TargetGroupTransferMaxChunkSize.
```

## Configuring Standard Editor to display Lyris Icon for Target Group Members

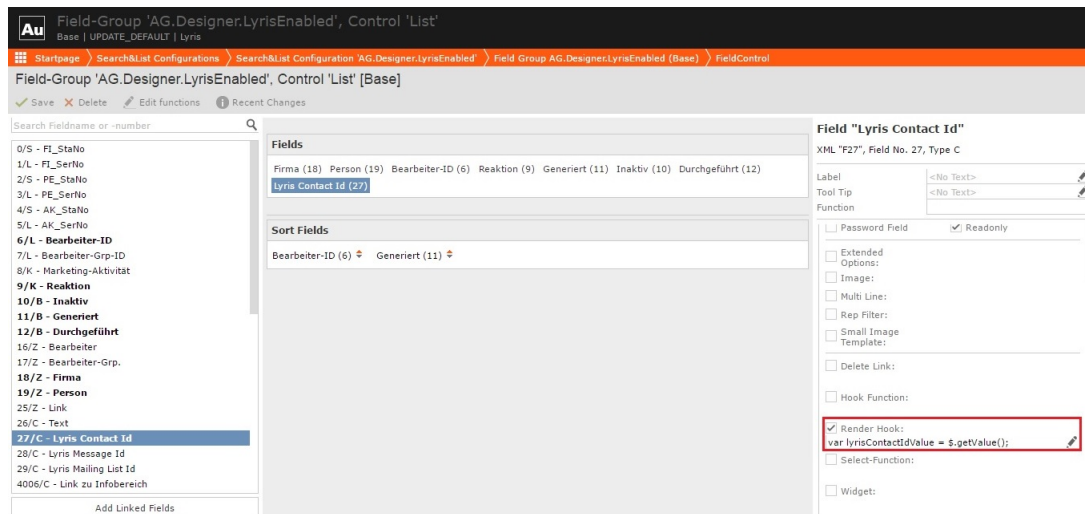
Lyris features available in Aurea CRM are accessible from the Campaign Designer, as well as the standard editors for marketing activities. However, the state of target group entities in terms of Lyris is not shown by default in the target group search list for standard editors. In order to enable an extra (custom) column that is capable of showing the Lyris icon in those target group entities already transferred to Lyris, perform the following steps in a List control of an AG field group:

To display the Lyris icon for target group members transferred to Lyris, perform the following steps:

1. Add new column for Lyris Contact Id field.

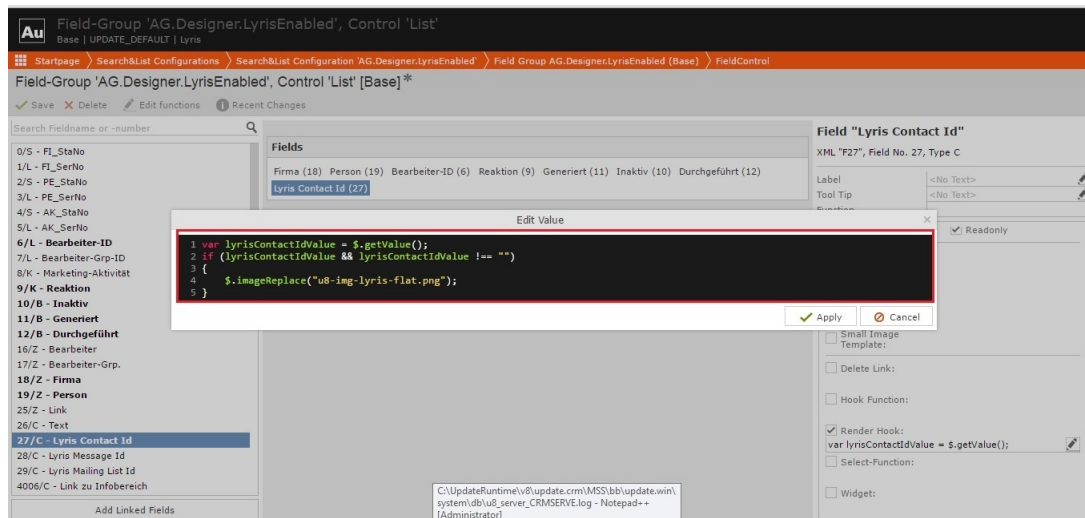


2. Click `Render Hook` field attribute in the **Field Definition** area.



3. Click **Edit** and enter the following javascript code:

```
var lyrisContactIdValue = $.getValue();
if (lyrisContactIdValue && lyrisContactIdValue !== "")
{
    $.imageReplace("u8-img-lyris-flat.png");
}
```



## ACM Event notification

Learn about the ACM event notification.

ACM sends the following event notifications when the Contact status of target group member changes:

- **Trashed** - A target group member is set to `Trashed` state in ACM.
- **Bounced** – A target group member is set to `Bounced` state in ACM.
- **Unsubscribed** – A target group member unsubscribes in ACM.
- **Active** - A target group member is set to `Active` state in ACM.

ACM sends the following event notifications on the status of the Email messages for a marketing activity:

- **Email Sent** – When ACM sends the marketing email message to a target group member.
- **Email Delivered** – When ACM delivers marketing activity email message to the inbox of a target group member.
- **Email Opened** – When the target group member opens the email message for the marketing activity.
- **Email Bounced** – When the email for a target group member could not be delivered for some reason.

Other event notifications from ACM:

- **URL Click Event** – When a user clicks on a response URL link in a marketing activity email message.
- **Update Boolean Field** – When the value of a property (demographic) of type `Boolean` (`checkbox`), of a target group member in ACM is updated.

- **Update Date Field** – When the value of a property (demographic), of type `Date`, of a target group member in ACM is updated. For example, the Date of Birth value of a target group member in ACM is updated.
- **Update Text Field** – When the value of a property (demographic), of type `text` or `text area`, of a target group member in ACM is updated. For example, the First Name value of a contact in ACM is updated.
- **Update Catalog field** – When the value of a property (demographic), of type `select List`, of a target group member in ACM is updated. For example, the Country field of a target group member in ACM is updated.

Some of the log messages generated by Aurea CRM after processing the event notification are shown below:

### Contact Trashed event

When a target group member’s status is set to Trashed in Lyris, a notification is generated for the responsible Rep of the marketing activity. The target group member in CRM is set to the inactive status. The notification message contains information about the CRM target group member that was set to inactive status. Aurea CRM on processing the contact trashed event, logs the following:

```

19719 2017.04.20;16:58:27; 2710;update.Crm.Services ;Debug ; 107704320; 6164;21,33212;lyris event message with NMS message Id ID:KONSTANTINOS-PC-9338-1492650344818-2;2;11:130 has started being parsed
19725 2017.04.20;16:58:27; 0;update.Crm.Services ;Debug ; 107705220; 6164;100;lyris event message with NMS message Id ID:KONSTANTINOS-PC-9338-1492650344818-2;2;11:130 and type ContactUpdate was su
19831 2017.04.20;16:58:34; 15;update.Crm.Services ;Debug ; 108642784; 6164;10;91111;trigger OnlyLyrisContactBounced was executed for Lyris contact set to 'Bounced' state on activity target group A0;
19838 2017.04.20;16:59:00; 20;update.Crm.Services ;Debug ; 107958272; 6164;16,47648;lyris event message with NMS message Id ID:KONSTANTINOS-PC-9338-1492650344818-2;2;11:131 has started being parsed
19844 2017.04.20;16:59:01; 47;update.Crm.Services ;Debug ; 107958272; 6164;100;lyris event message with NMS message Id ID:KONSTANTINOS-PC-9338-1492650344818-2;2;11:131 and type ContactUpdate was su
19950 2017.04.20;16:59:01; 32;update.Crm.Services ;Debug ; 108091152; 6164;52,65482;trigger OnlyLyrisContactUnsubscribed was executed for Lyris contact set to 'Unsubscribed' state on activity target
20167 2017.04.20;17:35:43; 3844;update.Crm.Services ;Debug ; 68857856; 6164;75,93649;lyris event message with NMS message Id ID:KONSTANTINOS-PC-9338-1492650344818-2;3;11:11 has started being parsed.
20279 2017.04.20;17:35:43; 79;update.Crm.Services ;Debug ; 69340480; 6164;100;lyris event message with NMS message Id ID:KONSTANTINOS-PC-9338-1492650344818-2;3;11:11 and type ContactUpdate was suc
20282 2017.04.20;17:35:49; 79;update.Crm.Services ;Debug ; 70546048; 6164;100;activity target group A0;XXXXXXXXXXXXXXXXXXXX was set to Inactive, because respective Lyris contact was trashed
    
```

### Contact Active event

When an inactive (or trashed) target group member is set to Active in ACM, a notification is sent to the responsible rep of the marketing activity. The target group member in CRM is set to the active status. The notification message contains information about the CRM target group member that is set to the active status. Aurea CRM on processing the contact active event, logs the following:

```

20187 2017.04.20;17:35:43; 3844;update.Crm.Services ;Debug ; 68857856; 6164;75,93649;lyris event message with NMS message Id ID:KONSTANTINOS-PC-9338-1492650344818-2;3;11:11 has started being parsed. Message content: <
20173 2017.04.20;17:35:43; 79;update.Crm.Services ;Debug ; 69340480; 6164;100;lyris event message with NMS message Id ID:KONSTANTINOS-PC-9338-1492650344818-2;3;11:11 and type ContactUpdate was successfully parsed.
20252 2017.04.20;17:35:49; 79;update.Crm.Services ;Debug ; 70546048; 6164;100;activity target group A0;XXXXXXXXXXXXXXXXXXXX was set to Inactive, because respective Lyris contact was trashed
20285 2017.04.20;17:39:53; 4938;update.Crm.Services ;Debug ; 75792384; 6164;72,29481;lyris event message with NMS message Id ID:KONSTANTINOS-PC-9338-1492650344818-2;3;11:12 has started being parsed. Message content: <
20291 2017.04.20;17:39:53; 47;update.Crm.Services ;Debug ; 75862016; 6164;100;lyris event message with NMS message Id ID:KONSTANTINOS-PC-9338-1492650344818-2;3;11:12 and type ContactUpdate was successfully parsed.
20298 2017.04.20;17:39:55; 31;update.Crm.Services ;Debug ; 74463552; 6164;86,16375;activity target group A0;XXXXXXXXXXXXXXXXXXXX was set to active, because respective Lyris contact was set to active status
<
Upgrade GamutLogViewer@today - Click Here!
    
```

### Contact Bounced event

When ACM changes the status of a contact to bounced, the event is received by Aurea CRM and it executes a predefined named trigger. You can define the trigger logic based on your business requirement. The name of the trigger can be specified in the Web configuration parameter `Lyris.TriggerOnLyrisContactBounced` parameter. If the parameter is empty then no trigger is executed by CRM and the event is ignored. For a description of the parameter, see [Lyris Parameters](#) on page 127.

Aurea CRM on receiving this event, executes the configured trigger and logs the

```

20291 2017.04.20;17:39:53; 47;update.Crm.Services ;Debug ; 75862016; 6164;100;lyris event message with NMS message Id ID:KONSTANTINOS-PC-9338-1492650344818-2;3;11:12 and type ContactUpdate was successfully parsed.
20298 2017.04.20;17:39:55; 31;update.Crm.Services ;Debug ; 74463552; 6164;86,16375;activity target group A0;XXXXXXXXXXXXXXXXXXXX was set to active, because respective Lyris contact was set to active status
20409 2017.04.20;17:43:37; 32453;update.Crm.Services ;Debug ; 76492800; 6164;72,07745;lyris event message with NMS message Id ID:KONSTANTINOS-PC-9338-1492650344818-2;3;11:13 has started being parsed. Message content: <
20409 2017.04.20;17:43:37; 47;update.Crm.Services ;Debug ; 76492800; 6164;100;lyris event message with NMS message Id ID:KONSTANTINOS-PC-9338-1492650344818-2;3;11:13 and type ContactUpdate was successfully parsed.
20215 2017.04.20;17:43:45; 10;update.Crm.Services ;Debug ; 80288864; 6164;89,31165;trigger OnlyLyrisContactBounced was executed for Lyris contact set to 'Bounced' state on activity target group A0;XXXXXXXXXXXXXXXXXXXX
<
Upgrade GamutLogViewer@today - Click Here!
    
```

following:

## Contact Unsubscribed Event

When Lyris changes the status of a contact to Unsubscribed, the event is received by Aurea CRM and it executes a predefined named trigger. You can define the trigger logic based on your business requirements. The name of the trigger can be specified in the Web configuration parameter `Lyris.TriggerOnLyrisContactUnsubscribed` parameter. If the parameter is empty then no trigger is executed by CRM and the event is ignored. For a description of the parameter, see [Lyris Parameters](#) on page 127.

## URL Click Events

Aurea CRM marketing activity email messages generated by ACM can contain response URL links to capture response from the target group members. When a recipient clicks on an URL, ACM generates an Email click event. On receiving the Email click event from the ACM Sync Middleware, Aurea CRM records the response in the Mailings (BR) info area record for the target group member, for each marketing activity.

The following screenshot shows the log generated in the CRM.Server log file on receiving the Email click event notification:

The screenshot displays a log entry with the following details:

```

28876 2017.04.27;17:18:29; 16;update.Crm.Services ;Debug ; 76845056; 15636;0 ;;Lyris url clicked: "www.lyris.com?response=NotMapped" is not mapped to any marketing activity res
29050 2017.04.27;17:46:11; 8828;update.Crm.Services ;Debug ; 68870144; 15636;75,19791;;Lyris event message with NMS message id ID:KONSTANTINOS-PC-21152-1493294229412-2:2:1:1:4 has
29053 2017.04.27;17:46:11; 47;update.Crm.Services ;Debug ; 69152768; 15636;100;;Lyris event message with NMS message id ID:KONSTANTINOS-PC-21152-1493294229412-2:2:1:1:4 and type
29115 2017.04.27;17:46:20; 156;update.Crm.Services ;Debug ; 75317248; 15636;100;;Mailing record BR;x0000006400000150 activity response 1 was set due to Lyris url click: "www.lyri
  
```

The log entry is expanded to show the following details:

```

Column 1 : 2017.04.27;17:46:20; 156;update.Crm.Services ;Debug ; 75317248; 15636;100;;Mailing record BR;x0000006400000150 activity response 1 was set due to Lyris url click:
"www.lyris.com?response=Accepted"
  
```

The message shows that the Mailing record was updated to record the first response from the recipient, a click action on the response URL: `www.lyris.com?response=accepted`.

An email message recipient can send multiple responses configured for a marketing activity. If the user sends the same response multiple times, they are ignored. The following screenshot shows the server log, for the third response from this recipient being a duplicate, as the response is already recorded. Hence, the Mailing record is not updated.

The screenshot displays a log entry with the following details:

```

29210 2017.04.27;17:49:09; 4609;update.Crm.Services ;Debug ; 77275136; 15636;99,40716;;Lyris event message with NMS message id ID:KONSTANTINOS-PC-21152-1493294229412-2:2:1:1:6 has
29213 2017.04.27;17:49:09; 0;update.Crm.Services ;Debug ; 77275136; 15636;100;;Lyris event message with NMS message id ID:KONSTANTINOS-PC-21152-1493294229412-2:2:1:1:6 and type
29271 2017.04.27;17:49:10; 31;update.Crm.Services ;Debug ; 79020032; 15636;100;;Response 3 corresponding to activity response AN;x000000640000060c0 already set in mailing
record BR;x0000006400000150
  
```

The log entry is expanded to show the following details:

```

Column 1 : 2017.04.27;17:49:10; 31;update.Crm.Services ;Debug ; 79020032; 15636;100;;Response 3 corresponding to activity response AN;x000000640000060c0 already set in mailing
record BR;x0000006400000150
  
```

If there are more than five responses configured for a marketing activity, and if the user clicks on all the response URLs then response event after the 5th response are ignored. Similarly, click events generated when recipients click on URLs in a message that are not mapped to a response are ignored. The appropriate log entries are created in the server log.

### Events with faulty data

When Aurea CRM receives an event with faulty data and is unable to process it further, it creates a task for the rep associated with the marketing activity for which the event was generated. The task provides information about the faulty data and enables the Rep to take appropriate actions.

Events generated with invalid data also generate appropriate log messages that is useful in understanding the type of invalid data and provide meaningful clues about the fault. Some of these invalid data scenarios are listed below:

- Invalid Message ID
- Invalid Mailing list ID
- Invalid Site ID
- Invalid Boolean field value
- Invalid catalog field value
- Click event notifications generated for recipients who are not members of Aurea CRM.

The following screenshot shows a log generated on receiving a faulty data. It indicates that a contact's demographic value for the country field was updated to Greece, but the corresponding Aurea CRM field could not be updated because Greece is not a valid value, as it is not part of the country catalog in CRM. This raises a task for the Rep responsible for the marketing activity. The Rep can then manually add Greece to the country catalog and update the target group member's country field.

```

2017.05.11;17:37:08; 94;configuration.cache;Info ; 13612;Cache Hit on path '/0/EntityCache/update.Configuration
).Entities.WebConfigParameter>_0_1'.
2017.05.11;17:37:08; 16;update.Crm.Database;Info ; 13612;WriteCursor(A1,Values=[2=,4=10000002,20=Failed to upd
ite field Land of PE record with value Greece communicated by Lyris Sync middleware.,46=3,58=Automatic update of PE reco
rd due to Lyris contact update failed.,4006=PE,4007=100,4008=7196])
2017.05.11;17:37:08; 219;update.Lib;Info ; 13612;Disposing CrmBaseSession.
2017.05.11;17:37:08; 0;update.Lib;Info ; 13612;Disposing ConfigurationSession(Configuration=10000003
2 as Dummy->0 as UPDATE_DEFAULT).
2017.05.11;17:37:08; 0;update.Lib;Info ; 13612;ConfigurationSession(Configuration=100000032 as Dummy
->0 as UPDATE_DEFAULT) has been disposed.
2017.05.11;17:37:08; 15;update.Lib;Info ; 13612;CrmBaseSession has been disposed.
2017.05.11;17:37:08; 16;update.Crm.Services;Warning; 13612;Could not propagate Lyris demographic update on CRM f
or Lyris demographic with id: 31647 on target group AG:00000000000043d2 due to the fact that Lyris demographic is mappe
d to the CRM catalog field with field number 6 of info area PE and not a valid value for this field was provided (Greece

```

The following screenshot shows that Aurea CRM could no find the target group entity to update the target group member's demographic details, because the mailing list ID for the target group member could not be found.

3630	2017.04.13;17:41:39;	0;update.Crm.Services;Debug; 101158912;	14380;100;;Lyris Events Listener has been started.
3706	2017.04.13;17:44:21;	17938;update.Crm.Services;Debug; 103800832;	14380;69,59026;;Lyris event message with NMS message id ID:KONSTANTINOS-PC-3565-1492076255559-2;3:1:1 h
3713	2017.04.13;17:44:21;	94;update.Crm.Services;Debug; 104071168;	14380;83,97628;;Lyris event message with NMS message id ID:KONSTANTINOS-PC-3565-1492076255559-2;3:1:1 a
3804	2017.04.13;17:44:27;	16;update.Crm.Services;Debug; 105689088;	14380;100;;Could not find target group entity with Lyris contact id bernard_allison@example.com to update

Upgrade GamutLogViewer@ today - Click Here!

Details about the currently selected line in the current log file

Column 1 : 2017.04.13;17:44:27; 16;update.Crm.Services;Debug; 105689088; 14380;100;;Could not find target group entity with Lyris contact id bernard\_allison@example.com to update.

## Launching Fat Clients Using Aurea CRM

Aurea CRM allows you to launch third party application on the user's machine using the CRM.launcher.

Aurea CRM uses the CRM.launcher to launch any third party fat client application from the command shell and also pass parameter values to the application. The fat client application should be registered with Aurea CRM designer and also installed locally on the client machine.

### Configuring a Fat client application in the CRM.Designer

Feature is implemented using action templates. You have to create a button and define the action using the action template DesktopAppLauncher. For more information on creating and configuring buttons, see Buttons.

To configure a fat client launcher in the CRM.Designer, perform the following steps:

1. Click **Buttons** on the CRM.Designer **Start** page.

The **Buttons** page displays.

2. Create a new button and open the button for editing.

The screenshot below, shows the **New Activity** button properties page.

Button 'New Activity'  
BB | UPDATE\_DEFAULT | DocFatClientConfig

Startpage > Buttons > Button 'launchFatClient\_7z'

Button 'launchFatClient\_7z'\*

Save Delete

Label Launch 7z

Image update Au

Action DesktopAppLauncher

Hot-Key Ctrl+Alt+N

Query State Action <No Action>

CRM Processes Marketing Activity (3001)

Help Text <No Text>

Description  
Enter description here.

Hide button in QuickView  
 Hide button text  
 Owner Draw Button  
 Disable button  
 Hide button

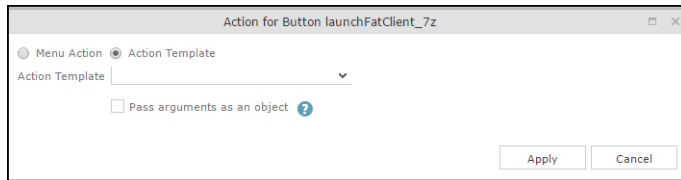
Referenced by  
Header [AK\\_SearchResults](#), [AKTree\\_SearchResults](#)

Recent changes

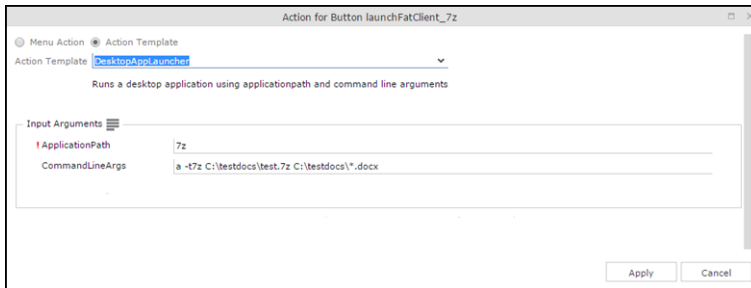
When	Operation	Who	Details
2016-06-24 16:40:36	Update	Default	update
2016-06-24 16:31:36	Copy	Default	From UPDATE_DEFAULT (Base) to DocFatClientConfig

Provide all the required information for the fields. For more information on working with buttons, see the chapter Buttons. The button is configured as part of the marketing activity CRM process and is displayed on the marketing activity search page.

3. Click  in the Action field. The **Action for Button <Name>** page displays.

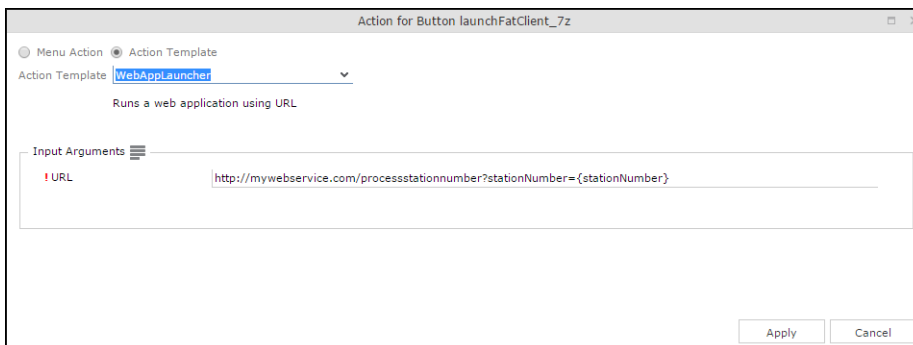


4. Select the **Action Template** radio button.
5. From the dropdown list, select the required fat client action template from the following:
  - DesktopAppLauncher—allows you to configure a fat client application available on the users’s system to be launched.
  - WebAppLauncher—allows you to configure a web application URL. This is opened in the user’s default browser.
6. Configure the action template selected.
  - If you select **DesktopAppLauncher** action template, the **Action for Button <Name>** page is populated as shown below.



Enter the following input arguments for the desktop fat client:

- Application path—specify the path to the application on the desktop.
- CommandLineArgs—specify the parameters and their values that can be passed to the application on the command shell.
- If you select WebAppLauncher action template, the **Action for Button <Name>** page is populated as shown below.





Enter the following input arguments for the web application fat client:

- URL—specify the URL to the web application.

---

**Note:** You can specify the following variables in the commandline arguments or the URL values for the fat client action templates. Their actual values is substituted at runtime when the launcher is opening the web url or launching the desktop application. See below for more information.

---

7. Click **Apply**. The action template is saved.

8. Click **Save**. The button configuration is saved.

The following variables can be used in the action template definitions of a button:

- {crm.repId}—provides the id value of the Rep.
- {crm.repName}—provides the name of the Rep.
- {crm.userName}—provides the name of the current user logged into Aurea CRM.Web.
- {crm.isSuperUser}—sets to true, if the current user is a super user, else sets to false.
- {crm.orgGroupId}—provides the groupid of the user,
- {crm.tenantName}—provides the tenant Name.
- {crm.tenantNo}—provides the tenant Number.
- {crm.stationNumber}—provides the station number of the current user.
- {crm.vertical}—provides the vertical type of the current Aurea CRM installation.
- {crm.applicationName}—provides the name of the application being launched.
- {crm.isOfflineClient}—sets to true if the current user is logged into an offline client, else sets to false.

Some examples of commandline values for the DesktopAppLauncher are provided in table below:

ApplicationPath	CommandLineArgs
C:\Program Files\Wireshark\wireshark	-i Ethernet -k
7z	a -t7z C:\Users\Administrator\Downloads\7ztest\files.7z C:\Users\Administrator\Downloads\7ztest\*.docx
C:\Program Files\Wireshark\wireshark	-o uat:user_dlt:User 0 (DLT=147),http,0,,0,
%TESTAPP-PATH%/testapp.exe	{crm.stationNumber} {crm.userName}

Some examples of URLs for the WebAppLauncher action template are provided below:

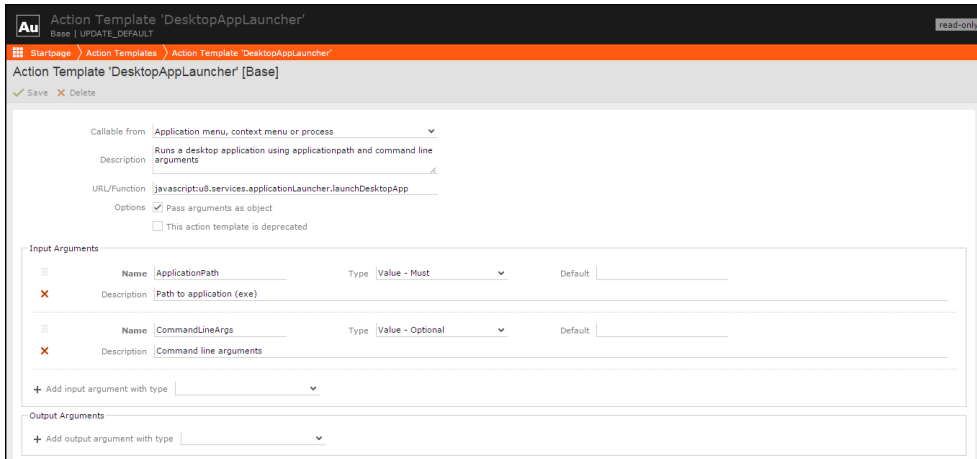
- `http://www.google.com?q=aurea crm`
- `http://localhost/displaycrmattributes?stationnumber={stationNumber}`

---

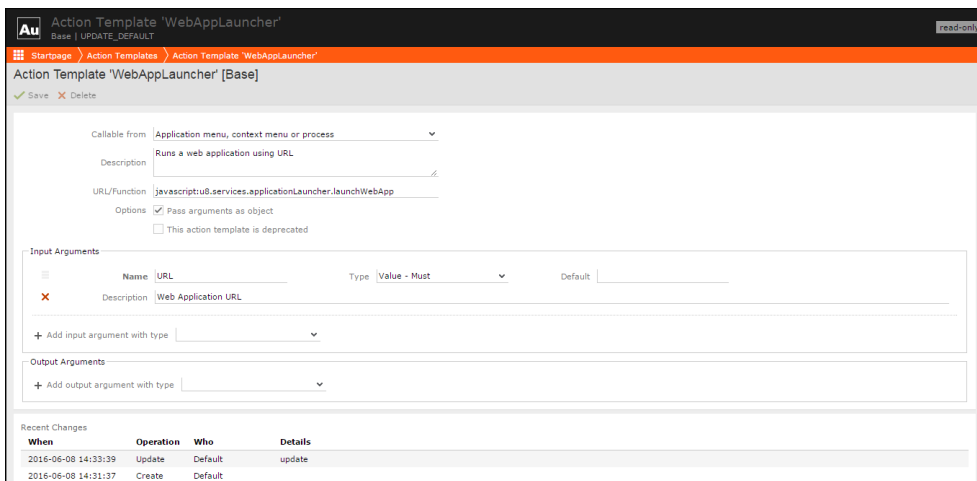
**Note:** Notice the use of the stationNumber variable in the URL.

---

The following screenshots show the base template of the DesktopAppLauncher action template.



The following screenshot shows the base template of the WebAppLauncher action template.



## Channels Output Cache Framework

Learn how to reduce the CRM server workload and improve the response time in high latent networks.

Aurea CRM.Web (Enterprise) provides output caching framework that reduces the CRM server workload and improve the response time in HIGH latent networks, by allowing you to cache the CRM web server output. Output cache framework takes care of server side caching and sets the appropriate client side (response) headers for a data channel decorated with cache output attribute.

Channel output cache framework decorates each channel's methods with `CacheOutputAttribute` attribute to make the method's output cacheable on server and client by design. To override the inbuilt cache output properties, you have to enable output caching first (disabled by default) and then override the `CacheOutputAttribute` properties using a channel's output cache configuration placed in the application configuration file, `web.config` in Aurea CRM application. In the `web.config` file, the cache output configuration is placed under the section `update.web.channelOutputCaching` of type `ChannelOutputCachingConfigurationSection`.

---

**Note:** A Channel refers to the pipeline used to retrieve configuration data, calendar data, data from the database, and similar information required for the various operations defined in the CRM server.

---

The following example shows the channel definition for a channel type with the cache output attribute elements defined in an external file (`channelsOutputCaching.config`).

```
<section name="update.web.channelOutputCaching"
type="update.Web.Base.Contracts.Caching.Configuration.ChannelOutputCaching
ConfigurationSection, update.Web.Base.Contracts"/>
...
<update.web.channelOutputCaching configSource="Configs\channelsOutputCaching.config"
/>
```

---

**Note:** Using `configSource` attribute to reference an external configuration file is kept `web.config` clean and uncluttered.

---

The following example provides the output cache configuration elements for the `FormChannel` and `configurationChannel` channel types in the configuration file.

```
<update.web.channelOutputCaching enabled="true">
  <channels>
    <channel channelType="update.Configuration.Web.Channels.FormChannel,
update.Configuration.Web" methodName="Get" enabled="false" />
    <channel channelType="update.Configuration.Web.Channels.ConfigurationChannel,
update.Configuration.Web" methodName="GetTexts" enabled="true" noCache="true"
clientTimeSpan="3600" serverTimeSpan="3600" mustRevalidate="true"
anonymousOnly="false" private="true" excludeQueryStringFromCacheKey="false"/>
  </channels>
</update.web.channelOutputCaching>
```

The cache output elements should be defined for each of the different data channel types for which you wish to enable client and server side caching for the output from different method calls. The different channel configuration attributes are described below:

- `channelType`—specify the channel type whose output is to be cached. Mandatory element. For the list of channel types that support caching, see the [table below](#).
- `enabled`—enable or disable caching the output of the channel. Enabled by default.
- `methodName`—specify the name of the method whose output is to be cached. Mandatory element. See the table below to get the names of methods that support caching in the list of channel types.
- `anonymousOnly`—cache enabled only for requests when `Thread.CurrentPrincipal` is not set. Default value is `true`.
- `clientTimeSpan`—time (in seconds) for which a response is cached on the client side. This corresponds to the HTTP Cache-Control general-header's cache-request-directive `max-age` (in seconds).
- `serverTimeSpan`—time (in seconds) for which a response is cached on the server side.
- `private`—ensures that a response is cached by the browser on the client side but is not cached by any intermediate caches like a proxy server. This corresponds to the HTTP Cache-Control general-header's cache-response-directive `private`.
- `mustRevalidate`—ensures that the cache entry is not used after it becomes stale, when responding to a subsequent request without first revalidating it with the origin server. This corresponds to the HTTP Cache-Control general-header's cache-response-directive `must-revalidate`.
- `excludeQueryStringFromCacheKey`—ensure that cache does not vary by querystring values. Default value is `true`.

The following table lists the channel types and the method names that support the output caching. The corresponding HTTP method is also pointed out.

Channel Type	Method Name	Implementation	HTTP Method
Administration Channel	GetSystemInfo	<p>Only client caching is supported.</p> <p>Supported HTTP Cache-Control directives:</p> <ul style="list-style-type: none"> <li>• <b>cache-request-directive:</b> <code>max-age</code> (in seconds).</li> <li>• <b>cache-response-directive:</b> <code>private</code> and <code>must-revalidate</code>.</li> </ul>	Get
AnalysisChannel	GetAnalysis	<p>Only client caching is supported.</p> <p>Supported HTTP Cache-Control directives:</p> <ul style="list-style-type: none"> <li>• <b>cache-request-directive:</b> <code>max-age</code> (in seconds).</li> <li>• <b>cache-response-directive:</b> <code>private</code> and <code>must-revalidate</code>.</li> </ul>	Get
AnalysisChannel	GetSettings	<p>Only client caching is supported.</p> <p>Supported HTTP Cache-Control directives:</p> <ul style="list-style-type: none"> <li>• <b>cache-request-directive:</b> <code>max-age</code> (in seconds).</li> <li>• <b>cache-response-directive:</b> <code>private</code> and <code>must-revalidate</code>.</li> </ul>	Get
Application Channel	GetLogin Settings	<p>Only client caching is supported.</p> <p>Supported HTTP Cache-Control directives:</p> <ul style="list-style-type: none"> <li>• <b>cache-request-directive:</b> <code>max-age</code> (in seconds).</li> <li>• <b>cache-response-directive:</b> <code>private</code> and <code>must-revalidate</code>.</li> </ul>	Get

Channel Type	Method Name	Implementation	HTTP Method
Application Channel	GetStartupData	<p>Only client caching is supported.</p> <p>Supported HTTP Cache-Control directives:</p> <ul style="list-style-type: none"> <li>• <b>cache-request-directive:</b> <code>max-age</code> (in seconds).</li> <li>• <b>cache-response-directive:</b> <code>private</code> and <code>must-revalidate</code>.</li> </ul>	Get
CalendarChannel	GetItem	<p>Server side caching (with ETag support) and client caching supported.</p> <p>Supported HTTP Cache-Control directives:</p> <ul style="list-style-type: none"> <li>• <b>cache-request-directive:</b> <code>max-age</code> (in seconds).</li> <li>• <b>cache-response-directive:</b> <code>private</code> and <code>must-revalidate</code>.</li> </ul>	Get
CalendarChannel	GetItems	<p>Server side caching (with ETag support) and client caching supported.</p> <p>Supported HTTP Cache-Control directives:</p> <ul style="list-style-type: none"> <li>• <b>cache-request-directive:</b> <code>max-age</code> (in seconds)</li> <li>• <b>cache-response-directive:</b> <code>private</code> and <code>must-revalidate</code>.</li> </ul>	Get
CalendarChannel	GetStartOf	<p>Server side caching (with ETag support) and client caching supported.</p> <p>Supported HTTP Cache-Control directives:</p> <ul style="list-style-type: none"> <li>• <b>cache-request-directive:</b> <code>max-age</code> (in seconds)</li> <li>• <b>cache-response-directive:</b> <code>private</code> and <code>must-revalidate</code>.</li> </ul>	Get

Channel Type	Method Name	Implementation	HTTP Method
CatalogChannel	GetCatalogs	<p>Server side caching (with ETag support) and client caching supported.</p> <p>Supported HTTP Cache-Control directives:</p> <ul style="list-style-type: none"> <li>• <b>cache-request-directive:</b> <code>max-age</code> (in seconds)</li> <li>• <b>cache-response-directive:</b> <code>private</code> and <code>must-revalidate</code>.</li> </ul>	Get
Collaboration Channel	GetHolidays	<p>Server side caching (with ETag support) and client caching supported.</p> <p>Supported HTTP Cache-Control directives:</p> <ul style="list-style-type: none"> <li>• <b>cache-request-directive:</b> <code>max-age</code> (in seconds)</li> <li>• <b>cache-response-directive:</b> <code>private</code> and <code>must-revalidate</code>.</li> </ul>	Get
Configuration Channel	GetActionItem	<p>Only client side caching supported.</p> <p>Supported HTTP Cache-Control directives:</p> <ul style="list-style-type: none"> <li>• <b>cache-request-directive:</b> <code>max-age</code> (in seconds)</li> <li>• <b>cache-response-directive:</b> <code>private</code> and <code>must-revalidate</code>.</li> </ul>	Get
Configuration Channel	GetChartTypes	<p>Server side caching (with ETag support) and client side caching supported.</p> <p>Supported HTTP Cache-Control directives:</p> <ul style="list-style-type: none"> <li>• <b>cache-request-directive:</b> <code>max-age</code> (in seconds)</li> <li>• <b>cache-response-directive:</b> <code>private</code> and <code>must-revalidate</code>.</li> </ul>	Get

Channel Type	Method Name	Implementation	HTTP Method
Configuration Channel	GetTextKeys	<p>Server side caching (with ETag support) and client caching supported.</p> <p>Supported HTTP Cache-Control directives:</p> <ul style="list-style-type: none"> <li>• <b>cache-request-directive:</b> <code>max-age</code> (in seconds)</li> <li>• <b>cache-response-directive:</b> <code>private</code> and <code>must-revalidate</code>.</li> </ul>	Get
Configuration Channel	GetTexts	<p>Server side caching (with ETag support) and client caching supported.</p> <p>Supported HTTP Cache-Control directives:</p> <ul style="list-style-type: none"> <li>• <b>cache-request-directive:</b> <code>max-age</code> (in seconds)</li> <li>• <b>cache-response-directive:</b> <code>private</code> and <code>must-revalidate</code>.</li> </ul>	Get
Favorites Channel	IsFavorite	<p>Server side caching (with ETag support) and client caching supported.</p> <p>Supported HTTP Cache-Control directives:</p> <ul style="list-style-type: none"> <li>• <b>cache-request-directive:</b> <code>max-age</code> (in seconds)</li> <li>• <b>cache-response-directive:</b> <code>private</code> and <code>must-revalidate</code>.</li> </ul>	Get
FormChannel	Get	<p>Server side caching (with ETag support) and client caching supported.</p> <p>Supported HTTP Cache-Control directives:</p> <ul style="list-style-type: none"> <li>• <b>cache-request-directive:</b> <code>max-age</code> (in seconds)</li> <li>• <b>cache-response-directive:</b> <code>private</code> and <code>must-revalidate</code>.</li> </ul>	Get



Channel Type	Method Name	Implementation	HTTP Method
HeadersChannel	GetHeader	<p>Server side caching (with ETag support) and client caching supported.</p> <p>Supported HTTP Cache-Control directives:</p> <ul style="list-style-type: none"> <li>• <b>cache-request-directive:</b> <code>max-age</code> (in seconds)</li> <li>• <b>cache-response-directive:</b> <code>private</code> and <code>must-revalidate</code>.</li> </ul>	Get
JobsChannel	Poll	No cache	Get
MenusChannel	GetDefaultAction	<p>Server side caching (with ETag support) and client caching supported.</p> <p>Supported HTTP Cache-Control directives:</p> <ul style="list-style-type: none"> <li>• <b>cache-request-directive:</b> <code>max-age</code> (in seconds)</li> <li>• <b>cache-response-directive:</b> <code>private</code> and <code>must-revalidate</code>.</li> </ul>	Get
MenusChannel	GetMenu	<p>Only client side caching.</p> <p>Supported HTTP Cache-Control directives:</p> <ul style="list-style-type: none"> <li>• <b>cache-request-directive:</b> <code>max-age</code> (in seconds)</li> <li>• <b>cache-response-directive:</b> <code>private</code> and <code>must-revalidate</code>.</li> </ul>	Get
MenusChannel	GetQuickSearch	<p>Server side caching (with ETag support) and client caching supported.</p> <p>Supported HTTP Cache-Control directives:</p> <ul style="list-style-type: none"> <li>• <b>cache-request-directive:</b> <code>max-age</code> (in seconds)</li> <li>• <b>cache-response-directive:</b> <code>private</code> and <code>must-revalidate</code>.</li> </ul>	Get

Channel Type	Method Name	Implementation	HTTP Method
QueryChannel	GetExpressionValues	<p>Server side caching (with ETag support) and client caching supported.</p> <p>Supported HTTP Cache-Control directives:</p> <ul style="list-style-type: none"> <li>• <b>cache-request-directive:</b> <code>max-age</code> (in seconds)</li> <li>• <b>cache-response-directive:</b> <code>private</code> and <code>must-revalidate</code>.</li> </ul>	Get
QueryChannel	GetFilter	<p>Server side caching (with ETag support) and client caching supported.</p> <p>Supported HTTP Cache-Control directives:</p> <ul style="list-style-type: none"> <li>• <b>cache-request-directive:</b> <code>max-age</code> (in seconds)</li> <li>• <b>cache-response-directive:</b> <code>private</code> and <code>must-revalidate</code>.</li> </ul>	Get
QueryChannel	GetFilterList	<p>Server side caching (with ETag support) and client caching supported.</p> <p>Supported HTTP Cache-Control directives:</p> <ul style="list-style-type: none"> <li>• <b>cache-request-directive:</b> <code>max-age</code> (in seconds)</li> <li>• <b>cache-response-directive:</b> <code>private</code> and <code>must-revalidate</code>.</li> </ul>	Get
QueryChannel	GetFilters	<p>Server side caching (with ETag support) and client caching supported.</p> <p>Supported HTTP Cache-Control directives:</p> <ul style="list-style-type: none"> <li>• <b>cache-request-directive:</b> <code>max-age</code> (in seconds)</li> <li>• <b>cache-response-directive:</b> <code>private</code> and <code>must-revalidate</code>.</li> </ul>	Get

Channel Type	Method Name	Implementation	HTTP Method
QueryChannel	GetFilterSets	<p>Only client side caching.</p> <p>Supported HTTP Cache-Control directives:</p> <ul style="list-style-type: none"> <li>• <b>cache-request-directive:</b> <code>max-age</code> (in seconds)</li> <li>• <b>cache-response-directive:</b> <code>private</code> and <code>must-revalidate</code>.</li> </ul>	Get
QueryChannel	GetQuery	<p>Server side caching (with ETag support) and client caching supported.</p> <p>Supported HTTP Cache-Control directives:</p> <ul style="list-style-type: none"> <li>• <b>cache-request-directive:</b> <code>max-age</code> (in seconds)</li> <li>• <b>cache-response-directive:</b> <code>private</code> and <code>must-revalidate</code>.</li> </ul>	Get
RepsChannel	GetRepSets	<p>Server side caching (with ETag support) and client caching supported.</p> <p>Supported HTTP Cache-Control directives:</p> <ul style="list-style-type: none"> <li>• <b>cache-request-directive:</b> <code>max-age</code> (in seconds)</li> <li>• <b>cache-response-directive:</b> <code>private</code> and <code>must-revalidate</code>.</li> </ul>	Get
RepsChannel	Read	<p>Server side caching (with ETag support) and client caching supported.</p> <p>Supported HTTP Cache-Control directives:</p> <ul style="list-style-type: none"> <li>• <b>cache-request-directive:</b> <code>max-age</code> (in seconds)</li> <li>• <b>cache-response-directive:</b> <code>private</code> and <code>must-revalidate</code>.</li> </ul>	Get

Channel Type	Method Name	Implementation	HTTP Method
RepsChannel	ReadMany	<p>Server side caching (with ETag support) and client caching supported.</p> <p>Supported HTTP Cache-Control directives:</p> <ul style="list-style-type: none"> <li>• <b>cache-request-directive:</b> <code>max-age</code> (in seconds)</li> <li>• <b>cache-response-directive:</b> <code>private</code> and <code>must-revalidate</code>.</li> </ul>	Get
SchemaChannel	GetInfoArea Schemas	<p>Only Client side caching supported.</p> <p>Supported HTTP Cache-Control directives:</p> <ul style="list-style-type: none"> <li>• <b>cache-request-directive:</b> <code>max-age</code> (in seconds)</li> <li>• <b>cache-response-directive:</b> <code>private</code> and <code>must-revalidate</code>.</li> </ul>	Get
SearchChannel	GetQuick Searches	<p>Only Client side caching supported.</p> <p>Supported HTTP Cache-Control directives:</p> <ul style="list-style-type: none"> <li>• <b>cache-request-directive:</b> <code>max-age</code> (in seconds)</li> <li>• <b>cache-response-directive:</b> <code>private</code> and <code>must-revalidate</code>.</li> </ul>	Get
SettingsChannel	Find	<p>Only Client side caching supported.</p> <p>Supported HTTP Cache-Control directives:</p> <ul style="list-style-type: none"> <li>• <b>cache-request-directive:</b> <code>max-age</code> (in seconds)</li> <li>• <b>cache-response-directive:</b> <code>private</code> and <code>must-revalidate</code>.</li> </ul>	Get

Channel Type	Method Name	Implementation	HTTP Method
SettingsChannel	Load	<p>Only Client side caching supported.</p> <p>Supported HTTP Cache-Control directives:</p> <ul style="list-style-type: none"> <li>• <b>cache-request-directive:</b> <code>max-age</code> (in seconds)</li> <li>• <b>cache-response-directive:</b> <code>private</code> and <code>must-revalidate</code>.</li> </ul>	Get
TableCaptionsChannel	Read	Only server side caching supported.	Post
TableCaptionsChannel	ReadBatch	Only server side caching supported.	Post
TypesChannel	GetTypes	<p>Server side caching (with ETag support) and client caching supported.</p> <p>Supported HTTP Cache-Control directives:</p> <ul style="list-style-type: none"> <li>• <b>cache-request-directive:</b> <code>max-age</code> (in seconds)</li> <li>• <b>cache-response-directive:</b> <code>private</code> and <code>must-revalidate</code>.</li> </ul>	Get

## KPI Definition (Enterprise CRM)

A KPI definition provides a quick measure of your business metrics primarily focusing on key targets and objectives of your business.

For example, you can calculate the current state of your sales for existing and new customers. This would be very useful in monitoring open calls, new customers of interest and other related information without having to be informed by email.

A KPI definition record allows you to specify operations on specific field of an info area and calculate the required metrics. You can specify target values to be achieved and measure the percentage of target achieved and the standard deviation from the target by scheduling the KPI definition to perform the calculation for specified time periods. For example, you could specify that the KPI calculation is performed “per week” and the calculation would consider the required field values for the previous week once the definition is executed.

The following example illustrates how the KPI calculations are performed for different calculation periods (assuming the KPI calculation is scheduled to start on May 10th 2017):

- Per Hour - The KPI value is calculated for the previous 60 minutes from the scheduled time.
- Per Day – The KPI value is calculated for the previous 24 hours from the scheduled date and time.
- Per Week – The KPI value is calculated for the previous week: April 30th to May 7th.
- Per Month - The KPI value is calculated for the previous month: April 1st to April 30th.
- Per Quarter – The KPI value is calculated for the previous quarter: Jan 1st to March 31st.
- Per Year - The KPI value is calculated for the previous year: Jan 1st to December 31st 2016.

You can use the following operations in a KPI definition to generate the required values: Add; Average; Count; Divide; Maximum; Minimum; Multiply; Subtract; and Sum.

As an administrator you can allow the CRM.web users to access one or more of the following menu items by creating appropriate roles and assigning them to the users.

1. Create and schedule a new KPI definition: Administration > KPI > New Schedule.
2. Search for scheduled KPI definitions: Administration > KPI > Search Schedules.
3. Search the results of the KPI runs performed: Administration > KPI > Search Results.

---

**Note:** If you do not find one or more of these in the KPI, contact your CRM.Web administrator to assign the appropriate role.

---

## Creating a New KPI Definition

Learn to create a new KPI definition.

To define and schedule a new KPI definition, perform the following steps:

1. Select the **Administration > KPI > New Schedule** menu. The **New: Definition**

page displays.

2. Provide the KPI definition by entering the following values in the Definition section:

- **Name:** Enter a name for the definition.
- **Target Value:** The expected target value for the KPI calculation.
- **Calculation Period:** The time periods for which the target values are calculated. You can select the following from the drop-down list:
  - **Per Hour**—calculates the KPI value for the previous 60 minutes from the scheduled time.
  - **Per Day**—calculates the KPI value for the previous 24 hours from the scheduled date and time.
  - **Per Week**—calculates the KPI value for the previous full week (Monday – Sunday) before generation.
  - **Per Month**—calculates the KPI value for the previous full month before generation.
  - **Per Quarter**—calculates the KPI value for the previous full quarter before generation.
  - **Per Year**—the calculation is always done for the previous full year before generation.

---

**Note:** The records selected for the calculation are the ones that have been created or updated in the specified Calculation Period. They are determined using the Created On(date), Created at(time), Updated on(date), and Updated at (time) fields of a record.

---

- **Info Area:** Select the Infoarea that contains the field of interest for the KPI calculation.
- **Field Id:** Select the field of interest from the drop-down list.



- **Calculation Method:** Select the type of calculation you want to perform from the drop-down list. The methods available depend on the type of the field you select. The following lists the possible values available: **Add; Average; Count; Divide; Maximum; Minimum; Multiply; Subtract; and Sum.**

---

**Note:** The Count method does not require a Field Id to be specified, as it only counts the number of records.

---

- **Operand Value:** Specify an operand value desired. For example, if you select divide as the calculation method, you have to specify the value of the divisor.

---

**Note:** An operand is only required for the Add, Divide, Multiply, and Subtract calculation methods.

---

3. Scheduling involves setting the start date and time for the definition as basic requirement to start running the KPI definition and calculating the required values. You can also re-schedule a new run of the definition by setting some of the values in this section. The fields in the Scheduling section are described below:

- **Start Date:** Specify the scheduled date to start the KPI calculation.
- **Start Time:** Select the time of the day to start the KPI calculation.
- **Duration:** Specify the duration for which this calculation should be executed. If the calculation exceeds the specified duration then it is aborted.
- **Duration Units:** Specify the units of the duration. Select from the following: Sec, Min, Hour(s), and Day(s).
- **Repeats:** To set a repetition pattern for the rescheduled execution of the definition. You can select from the following options: **Daily**; **Weekly**; **Monthly**; **Quarterly**; **Yearly**.

---

**Note:** If you select **Weekly** then you can optionally specify the weekday in the **Repeats On** field.

---

- **Repeats On:** Specify the optional weekday on which the execution pattern is repeated. You can select from Monday to Sunday.
- **Interval:** An integer value that amplifies the repetition pattern selected in the Repeats field. For example, if you have the Start Date set to 2017.03.31, Repeats to Weekly, Interval to 2 and Repeats On to Thursday, next execution is set to 2017.04.13 which is Thursday 2 weeks from start date.
- **Starts On:** Specify the date on which the repeated executions of the definition starts.
  - o **Ends On:** Specify the date on which the repeated executions of the definition stops.
  - o **Ends After:** Specify the number of times the KPI execution is rescheduled.

---

**Note:** The Ends On date is checked first and if the deadline is crossed then the rescheduled execution is stopped. This takes priority over the Ends After.

---

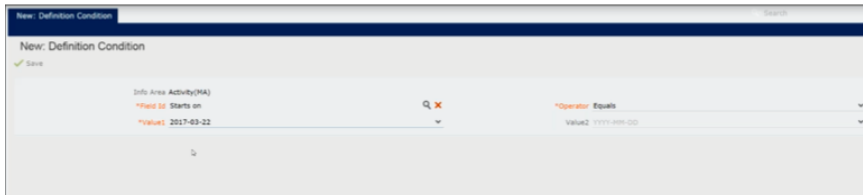
4. The **Next Run** section displays the date and time for when this definition is scheduled to be run again. These values are automatically populated by Aurea CRM. Opening this record at a later time should show when the next run of the definition is scheduled.

5. Click **Save**. The new definition is created and scheduled.

6. You can add additional conditions to the definition. The definition is executed only when all the conditions are satisfied.

To add conditions to the definition, perform the following:

- a. Select the **Definition Condition** tab in the **Related Data** section.
- b. Click **New Condition**. The **New: Definition Condition** page displays.



**Note:** The **Info Area** field is populated with the value you selected for the KPI definition. This is because a KPI definition can be defined for only one infoarea and cannot span multiple infoareas.

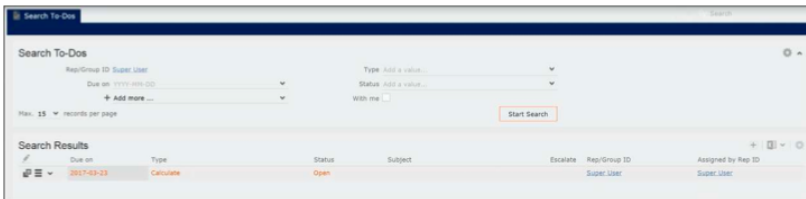
In the sample screenshot, the condition is being set for a definition on the Activity infoarea and we set the condition to ensure that the KPI definition calculation only considers activities that start on a specific date.

- c. Select the field that you want to set the condition on, from the **Field Id** drop-down list.
- d. Select the operator you want to apply on the field, from the **Operator** drop-down list. The type of operators available depends on the type of field selected. In this case we selected a date field and the operators available would: **less than**, **greater than**, **between**, and **equals**.
- e. Select the required date in the **Value1** field. The number of fields to be populated again depends on the type of the operator. For example, if you select the operator **between**, then you have to specify two dates to select the activities that start in this date range.
- f. Click **Save**.

7. You can view the list of definitions scheduled to run in the future, you can search for them in the **To- Dos** search window.

To view the scheduled definitions in the queue, perform the following:

- a. Select the **Calendar > To-Dos** menu.
- b. Select the type of To-Do from the **Type** drop-down list. In this case it is **Calculate**.
- c. Select the status if the To-Do from the **Status** drop-down list. In this case it is **Open**.
- d. Click **Start Search**. The scheduled definitions display in the **Search Results list**.



## Searching for KPI Definitions

Learn how to search for KPI Definitions.

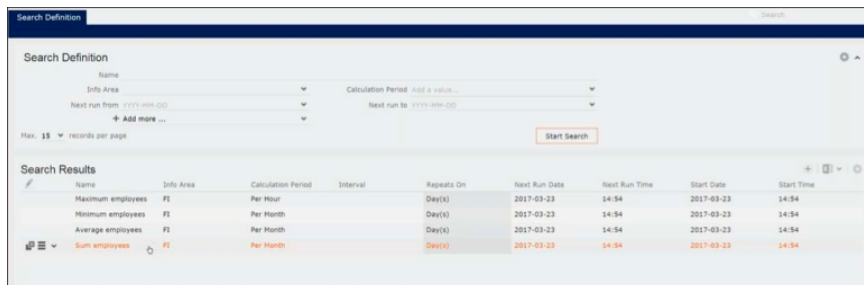
To search for existing KPI definitions, perform the following:

1. Select **Administration > KPI > Search Schedules** .

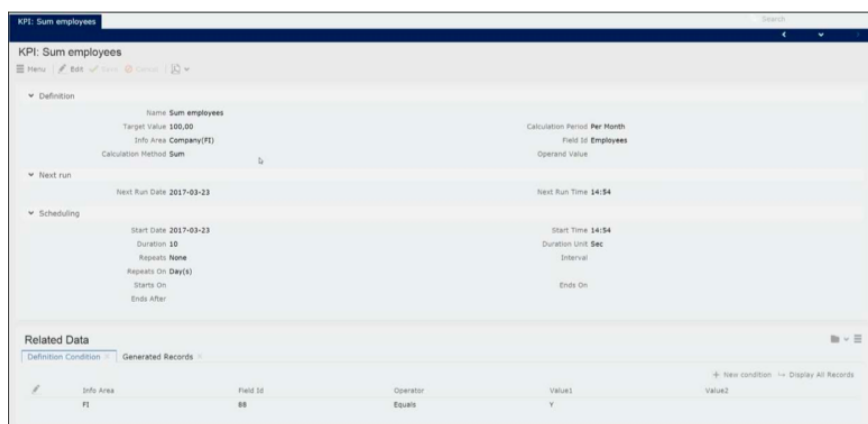
The **Search Definition** page displays.

2. You can filter the search result by providing relevant values in the fields provided. For example, provide the infoarea in the search filter to find all KPI calculations defined for that infoarea.
3. Click **Start Search**.

The search results display.



4. Double click on the required item in the result list to open the **Definition** page.



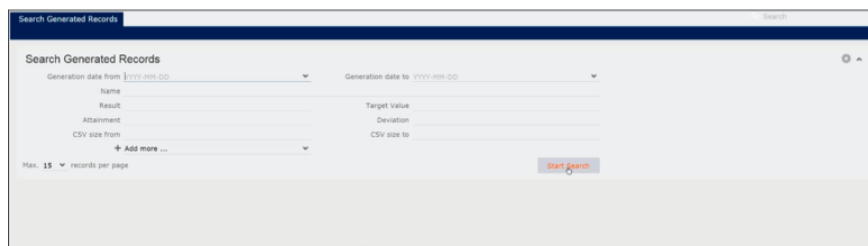
## Viewing the Generated Records

Learn how to view the generated records.

When a scheduled Definition is executed it generates records that contain the KPI values and also the status/values of the KPI calculations for the previous scheduled runs in a `.csv` file. To view the generated results perform the following steps:

1. Select **Administration > KPI > Search Schedules**.

The **Search Generated Records** page displays.



2. Enter the required values to filter the search results.
3. Click **Start Search**.

The search results list displays the list of generated records.

Name	Created on	Created at	Result	Target Value	Attachment	Deviation	Csv size (bytes)
Sum employees	2017-03-23	13:51:06.599	450.00	100.00	450.00%	350.00	116
Sum employees	2017-03-23	14:01:04.838	0.00	100.00	0.00%	-100.00	0
Sum employees	2017-03-23	14:07:43.440	0.00	100.00	0.00%	-100.00	0
Sum employees	2017-03-23	14:20:29.155	0.00	100.00	0.00%	-100.00	0
Sum employees	2017-03-23	14:27:38.061	100.00	100.00	100.00%	0.00	65
Sum employees	2017-03-23	14:35:07.517	200.00	100.00	200.00%	100.00	83
Average employees	2017-03-23	14:40:58.861	200.00	100.00	200.00%	100.00	83
Minimum employees	2017-03-23	14:40:10.787	200.00	50.00	400.00%	150.00	83
Maximum employees	2017-03-23	14:41:12.597	200.00	500.00	40.00%	-300.00	83
Maximum employees	2017-03-23	14:47:48.265	200.00	500.00	40.00%	-300.00	83
Sum employees	2017-03-23	14:55:12.827	250.00	100.00	250.00%	150.00	86
Average employees	2017-03-23	14:55:12.845	125.00	100.00	125.00%	25.00	86
Minimum employees	2017-03-23	14:55:12.861	100.00	50.00	200.00%	50.00	86
Maximum employees	2017-03-23	14:55:12.882	150.00	500.00	30.00%	-350.00	86
Average employees	2017-03-23	15:13:05.796	125.00	100.00	125.00%	25.00	86

## 4. Double click the required record.



The result record for the KPI calculation displays the following details:

- **Name:** The name of the KPI definition.
  - **Result:** The KPI value calculated according to the definition for the selected time period.
  - **Attainment:** The percentage value of the KPI result versus the expected target value.
  - **Target Value:** The expected target value for the KPI result specified in the target definition.
  - **Deviation:** The standard deviation of the KPI result from the target value.
  - **CSV size (bytes):** The size of the CSV file generated that contains the details of all the infoarea records considered in calculating the target value.
5. Click the link to the generated CSV file listed in the **Document Links** tab under **Parent Data**. The document is downloaded to your local system.
  6. Open the downloaded CSV file and open it in a spreadsheet editor.

InfoAreaId	RecordId	Employees
FI	429496729601	
FI	429496729602	100
FI	429496729603	200
FI	429496729604	150

The spreadsheet shows three columns containing values for the infoarea, the Record Id, and the infoarea field used to calculate the KPI. In this instance it shows the record ids of the records from the infoarea whose Employees field value was used to calculate the total number of employees in all the companies.

If the KPI definition also has conditions defined, then the fields used in the conditions are added to the CSV file following the field used in the definition. For example, if there are two conditions defined then there is two more columns for the two fields used in the conditions.

# Packages

A CRM.Designer package consists of a set of configuration units defined in one CRM.Designer configuration.

A package is defined within CRM.Designer and can then be downloaded and uploaded as XML files using CRM.bulkloader. For further details, see [CRM.bulkloader](#) on page 760.

Contrary to uploading complete configurations (see [Transferring Configurations to Another CRM.Designer Database](#) on page 78), it is not necessary to upload a package into a configuration with the same name: You can select the target configuration before uploading.

---

**Note:** Package uploads are data model-bound: The target database must have the same data model as the source database at least concerning the tables, fields and links used in the package. Currently there is no data model mapping implemented.

---



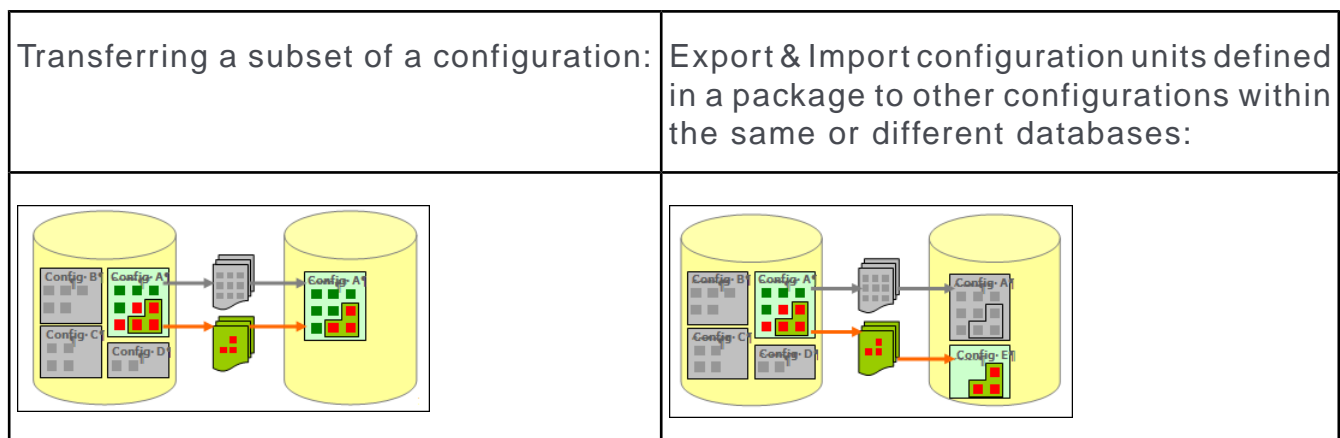
---

**Note:** After upgrading the CRM.Designer database (see [Maintenance](#) on page 768) you need to re-generate (download again) all packages, otherwise changes done by CRM.bulkloader is lost in the target database.

---

Use packages to

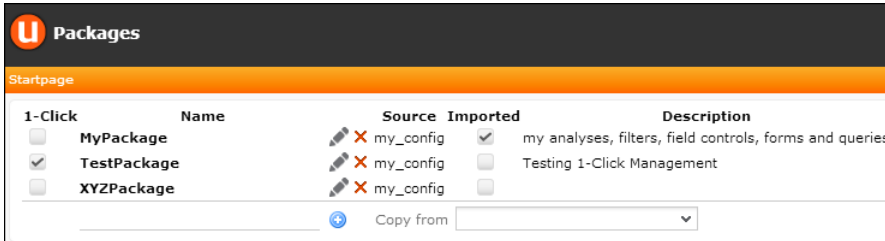
- transfer specific configuration units instead of a complete configuration.
- re-use parts of a customization as one logical package for various projects given that the data model concerning the package content is the same in the source and target databases.
- copy configuration units from one configuration to another within the same CRM.Designer database.



# Defining a Package

Learn to define a package in CRM.Designer.

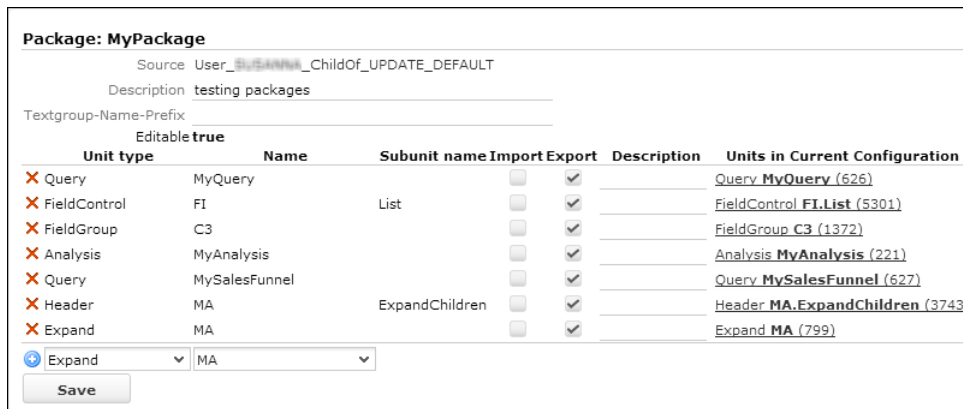
On the CRM.Designer main page in the **Packages** area, click **Packages**.



All packages defined for the current configuration are listed. Imported packages are flagged as **Imported**.

To define a package:

1. Select a package or create a new one. For further details, see [Designing Configuration Units](#) on page 64.



The **Editable** field indicates whether the package is editable or not.





---

If a package was imported, it is by default not editable to ensure its "upgradeability". To take ownership of an imported package, click **Take Ownership**. If you take ownership of a package, you can edit it but you cannot upload an updated definition of this package or package data to your configuration anymore.

2. Select the desired **Unit type**, its **Name** (and **Subunit name**, if applicable). For further details, see [Dependent Units and Sub-Units](#) on page 713. Only configuration units defined in the current configuration are available.

You can add a description for each item.

3. Click + (Add).

Uncheck **Export** for a configuration unit to temporarily remove it from the package.

## Dependent Units and Sub-Units

Learn about adding dependent units and sub-units to package.

In general, dependent units must be manually added to a package.

Examples: If you add a process to the package you need to manually add all required process steps as well. If you add a menu to the package, you need to explicitly add all menu actions as well.

Adding **sub-units** (field controls and headers) to a package:

If you add a field control or a header to a package without adding the field group/header group to which the control/header belongs, the field group/header group is not automatically created when importing the package. If the field group/header group does not exist in the target database, the subunits are not imported. The package displays "not in configuration" for those units in CRM.Designer.

In order to correctly transfer sub-units, you must own the field group/header group they belong to and explicitly add the group to the package as well.

### Handling text references:

All text references used by the configuration units in the package are implicitly added and therefore automatically exported and imported with the package.

---

**Note:** Texts defined in another configuration are not exported within the package - only the reference is exported and imported, and uses the same text defined by `textgroup` and `textid` in the destination database. Please be aware that you cannot define whether a text shall be imported or not unless the text is explicitly included in the package.

---

To make sure that texts implicitly contained in your package do not overwrite existing texts in the target configuration when uploading the package, you can specify a **Textgroup-Name-Prefix** for your package. If a prefix is defined, texts are downloaded and uploaded as a text group named `<Prefix><Textgroup Name>` instead of `<Textgroup Name>`. Thus, all text groups contained in your package gets the prefix - except text groups that were explicitly added to the package, text groups that are not part of the package configuration, and text groups that already contain this prefix.

You can also specify or change the **Textgroup-Name-Prefix** for imported packages: The prefix is applied when the package is uploaded.

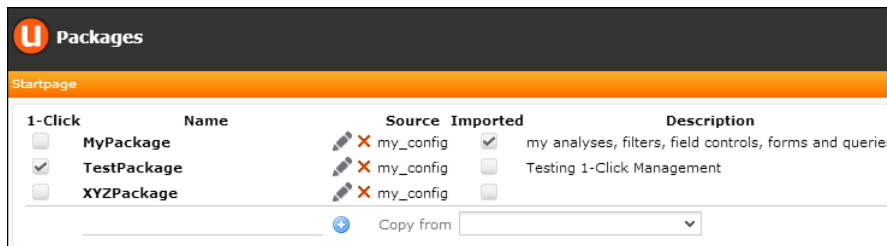
## 1-Click Package Management

Add or delete configuration units from a package using 1-Click Package Management.

With 1-Click package management you can add or delete configuration units from a package with a single click directly from a configuration unit's definition page.

To use 1-Click package management:

1. Check **1-Click** for the desired package(s).



2. Navigate to each configuration unit you want to add to the package.



For each configuration unit defined in the current configuration, the selected 1-Click package(s) are displayed in the upper-right part of the definition page.

3. Activate the check box beside the package(s) to which you want to add the configuration unit. Clicking on the package name navigates to the package definition.

---

**Note:** Be aware that selecting another configuration on the CRM.Designer main page resets the 1-Click settings.

---

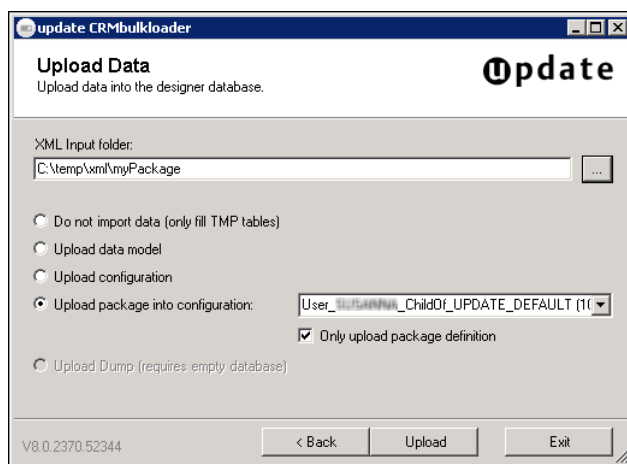
4. Download the package. For further details, see [Transferring Packages](#) on page 715.

## Transferring Packages

Learn how to transfer packages.

To transfer a package:

1. Create a directory for the XML files.
2. Download the desired package to this directory. For further details, see [Downloading Configuration Data from the CRM.Designer Database](#) on page 761.
3. Upload the XML files containing the package to the target database and/or configuration. For further details, see [Uploading Configuration Data to the CRM.Designer Database](#) on page 764.



Check **Only upload package definition** when uploading a package for the first time:

- If you upload a **package** to a configuration that has no package definition of this package yet, everything contained in the package is transferred to that configuration. This means that all configuration units of the target configuration that also exist in the package are replaced by the package definition (example - if you have an analysis named "MyScore" defined in your configuration, and the package contains an analysis of the same name, the analysis from the package overwrites the one already existing in your configuration).
- If you upload just the **package definition**, only the package definition but not its content is added to the configuration. After uploading a package definition, you can use CRM.Designer to decide for each configuration unit from the package whether it should be imported or not. For further details, see [Editing Imported Package Definitions](#) on page 716.

Once you have decided which units of the package should be imported, you can import the package itself with CRM.bulkloader without checking **Only upload package definition** this time.

**Note: Data Model Considerations:** When downloading a package, only tables and fields that are actually used by the package are exported. When importing a package, the same rules as for sub-configurations are applied. For further details, see [Uploading a Sub-Configuration](#) on page 768. This means that tables and fields contained in the package which are unknown to the target CRM.Designer database are added to the data model information but are marked as "deleted".

## Editing Imported Package Definitions

Learn to edit imported package definitions.

Import a package definition (**Only upload package definition** option) instead of the actual package, in order to be able to check the results before actually overwriting existing configuration units.

View the imported package definition in CRM.Designer (**Packages** page):

**Package: MyPackage**

Source my\_config

Description my analyses, filters and field controls

Textgroup-Name-Prefix

Editable **false**

Unit type	Name	Subunit name	Import	Export	Description	Units in Current Configuration
FieldControl FI		Details	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		FieldControl <b>FI.Details</b> (2167)
Analysis	MyCompanyBuyingFrequency		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<b>Not in configuration</b>
Form	MyForm		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<b>Not in configuration</b>
Query	MyOpportunities		<input type="checkbox"/>	<input checked="" type="checkbox"/>		Query <b>MyOpportunities</b> (166)

In this example, a package definition is imported to a configuration where the Details control for the FI info area and a query named `MyOpportunities` already exist. The other units do not yet exist in the configuration because only the package definition is uploaded.

1. Uncheck **Import** for those configuration units which you do not want to overwrite.

In this example, the query definition which already exists in the current configuration is kept (**Import** unchecked), the FI Details control is overwritten.

2. Save the package definition.

3. Upload the package again without **Only upload package definition** to actually import the package contents.

If the package itself had been uploaded (instead the package definition), both the query and the field control would have been overwritten.

# Administration

Learn to configure administration on aurea CRM.

The **Administration** area contains various functions for administrating the CRM.Designer database.

Here you can:

- Register images to be used by configuration units. For further details, see [Images](#) on page 717.
- Manage catalog values used in the CRM.Designer database. For further details, see [Catalogs](#) on page 720.
- Manage the assignment of CRM processes and roles to configuration units. For further details, see [Roles & CRM Processes](#) on page 727.
- View, verify and delete configuration units. For further details, see [Delete Units](#) on page 738, [Consistency Check](#) on page 742, [Configuration Validator](#) on page 744 etc.
- Configure access to CRM.Designer. For further details, see [Authentication](#) on page 748.

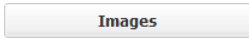


## Images

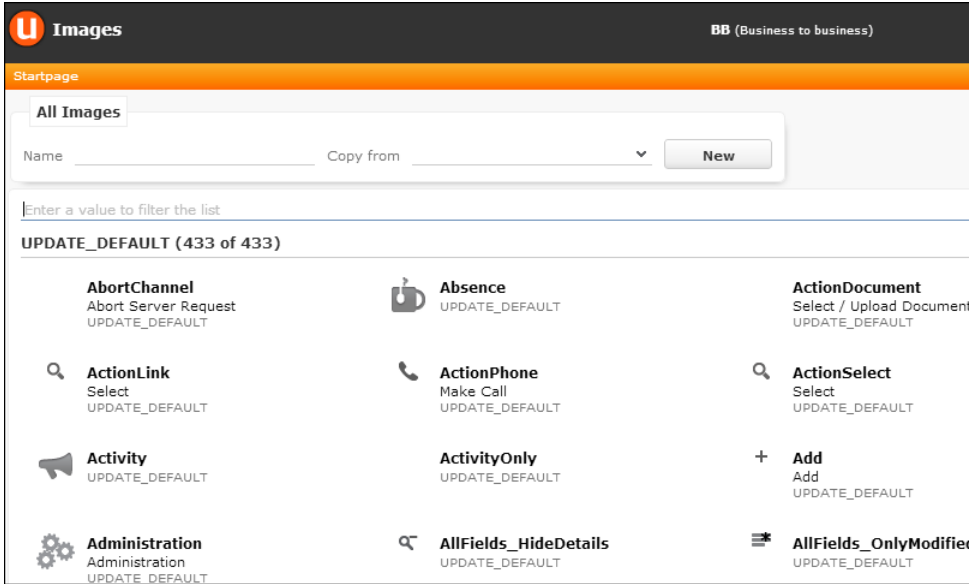
This topic defines how to deal with images in update.CRM web.

The **Images** page of CRM.Designer allows you to "register" images in the CRM.Designer database by specifying a logical and physical name. These images can then be included in table captions, headers, buttons, used for catalog values, etc.

On the CRM.Designer main page in the **Administration** area, click the **Images** button to open the **Images** page.



A list of all available images is displayed:



To add/change images in Aurea CRM.Web:

1. Create the image as PNG.

Depending on where you want to use an image, the following image sizes are required:

32x32 px		"normal" size - used in the navigation bar and in extended page headers
16x16 px	s	small - used by header buttons and icons
14x14 px	xs	extra small - used by menu items

The image sizes are specified by "decorators" in the file name (example - myimage-s.png is the small version of myimage.png). For more information on image naming" see the article "Image Naming Guidelines" at <https://support.aurea.com>.

If a required image size does not exist (example - -s for a button), the image is not displayed. Therefore make sure you add at least the -s and -xs version of your custom images. Depending on where you want to use an image, the following image sizes are required:

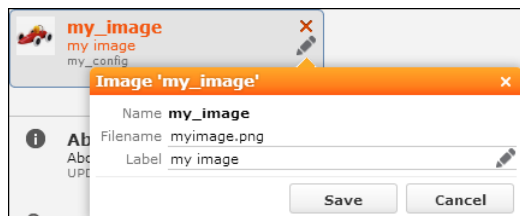
2. Copy the files to the required location (`..\images\<<theme>`). For further details, see [Customizing Styles](#) on page 15.

---

**Note:** To display new images in CRM.Designer as well, copy them to the `..\web\images\<theme>` directory of CRM.Designer.

---

### 3. Register the image in CRM.Designer:



- On the **Images** page in CRM.Designer click  (**Edit**) beside the image or create a new one.
- Enter the **Filename**.
- You can define a **Label** for an image that is then displayed on mouse-over.
- Click **Save**.

---

**Note:** You only have to register one version (i.e. size) of your custom image. The -s, -xs etc. versions are then applied automatically.

---

### 4. Assign the image to the desired configuration unit(s).

---

**Note:** To display your new images in Aurea CRM.Web you need to delete the `.sprites` folder in `..\images`. Furthermore, the users may have to clear their browsing history.

---

---

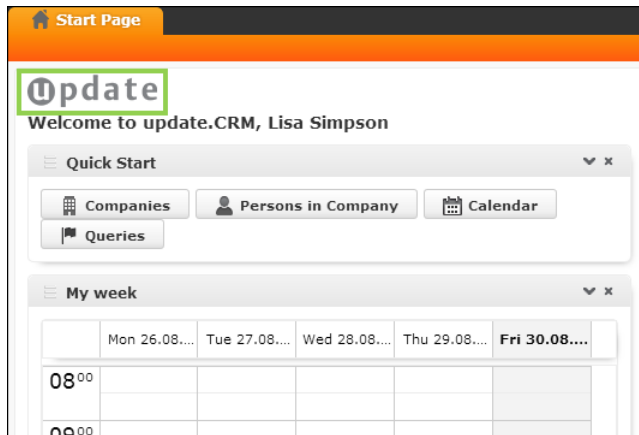
**Note:** To display images for a record (example - a person's photograph), you do not need to register them on the **Images** page. These images need to be uploaded in the Aurea CRM database instead. They can be displayed in any document field with the Image field attribute checked. For further details, see [Image](#) on page 229.

---

For additional information on images in Aurea CRM.Web, see the article “Images” at <https://support.aurea.com>.

## Customizing the Company Logo on the Start Page

Learn to customize company logo on the start page.



The default start page defined by form `Startpage` displays the image `Logo` (`u8-logo.png`).

To display a custom company logo on the start page:

1. Add the desired image to the image directory (`..\images\update`).
2. Copy `Logo` to your configuration and enter the file name of your company logo.

## Catalogs

Learn about catalog in CRM.

### Catalog Value Synchronization

Catalog values have four representations in Aurea CRM:

- **Catalog code:** A numeric ID defining a catalog value within a catalog for a specific Aurea CRM database. If the catalog has a parent catalog, the number is unique within all catalog values that have the same parent.

For fixed catalogs, the numeric ID uniquely defines a value throughout all Aurea CRM databases.

- **Catalog text:** Can be language-dependent and can contain the tenant number.
- **Catalog text in the catalog default language plus the tenant number:** The "primary key" uniquely identifying a catalog value for multiple databases as long as catalog values are not changed using the Aurea CRM catalog maintenance. If the catalog has a parent catalog, the parent value also must be known.
- **External key:** The external key of the catalog value which is optional. Uniquely identifies catalog values throughout the system.



The external key can be used as a unique catalog ID which does not change when transporting the catalogs between different stations (example - development, test and live station). For further details, see **Catalogs** in the *CRM.Core Administrator Guide* ([https://help.aurea.com/crm/crm-core-admin-guide/crm-core-admin-guide/dita/topics/c\\_core\\_admin-1\\_1-Introduction\\_crm.html](https://help.aurea.com/crm/crm-core-admin-guide/crm-core-admin-guide/dita/topics/c_core_admin-1_1-Introduction_crm.html)).

To automatically create external keys when adding catalog values, enable the **Automatically enter external keys** configuration entry (**Configuration** info area (MC), **Catalog** category).

If you apply a CRM.Designer configuration containing catalog values to another Aurea CRM database, you experience problems if you work with the catalog code or catalog text.

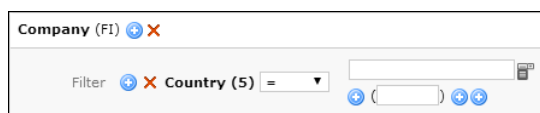
In some cases, if you specify a catalog value, CRM.Designer offers you the synchronization of that catalog by text in catalog base language plus tenant number and you are safe throughout the Aurea CRM database.

These cases are:

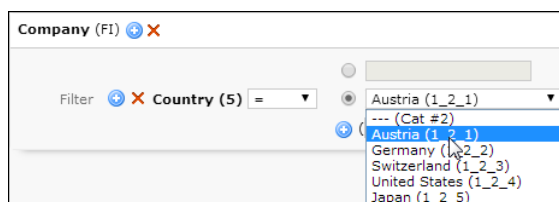
- Filter values in queries and filters
- Alternate Expand configurations
- Analysis categories
- Web Configuration parameter values based on catalog values
- Timeline configurations

**Example** (Defining a filter):

1. Add a catalog field to the filter definition.



2. Click on  (Catalog Input).



- Select the upper radio button to enter the catalog value as text (<text> or #<code>). For further details, see [Filtering by Catalog Values](#) on page 885.
- Select the lower radio button (default) to select a synchronized catalog value from the drop-down list.

If a catalog has not yet been synchronized, the drop-down list only displays the catalog's number. If the desired value is not contained in the drop-down list, the catalog values from the CRM.Designer database might not reflect the entire current content of the catalog. In both cases, update the catalog using CRM.bulkloader. For further details, see [Updating Catalogs](#) on page 778.

In all other cases, when only Aurea CRM.Web recognizes values as catalog values, example - `copySourceFields` in an `Expand` action, you must work around this issue with a different solution.

For this feature, CRM.Designer must have a copy of the catalog values in the default language plus the tenant number in its database. It assigns a catalog value ID (must be unique in each CRM.Designer database) to each entry. Single catalog values are automatically copied if you save a query or filter in the database using Aurea CRM.Web.

Use CRM.bulkloader to copy entire catalogs to the CRM.Designer database. For further details, see [Updating Catalogs](#) on page 778.

Since the CRM.Designer caches the data model information, update recommends that you perform an application reset (IIS reset or application pool recycling) after updating catalogs.

Please note that CRM.Designer's catalog synchronization feature is affected by a data model upgrade if a catalog was completely deleted from the data model – see [Aurea CRM Data Model Upgrade Implications on Synchronized Catalogs](#) on page 781 for more information.

If you transfer a CRM.Designer configuration to another CRM.Designer database, all catalog values used in this configuration are transferred and catalog-value-IDs are adjusted accordingly. If a configuration unit with such a catalog value is accessed, the catalog code is defined by a catalog lookup using the text in the catalog base language plus tenant number in the destination Aurea CRM database.

As catalog content is cached in CRM.Designer, you can press F5 or open a new browser window to see the changes.

## CRM.Designer Catalog Maintenance

Learn to maintain catalog using Aurea CRM Designer.

If you want to modify catalogs using the Aurea CRM catalog maintenance, this is the **supported way** to ensure that you have valid catalog information in the CRM.Designer:

1. Perform a normal catalog synchronization using CRM.bulkloader.
2. Perform the text changes in the Aurea CRM catalog maintenance.
3. Perform catalog synchronization by code or CRM.bulkloader.

**Note:** Catalog entries are not merged or deleted during this process – you have to do this manually as described in this section.

Catalog values which were modified with the Aurea CRM win **Maintenance** module using the catalog maintenance tool is inserted as new values into the CRM.Designer when synchronized, and the old values is not removed! Catalogs in the CRM.Designer database can be maintained in the **Catalogs** section of the designer.

After clicking the **Catalogs** button on the designer main page, you get a list of all catalogs that are currently stored in the CRM.Designer database.

Catalogs

**Catalogs**

BB (Business to business)

Startpage

**VARIABLE CATALOGS**

CatNr	CatName	Parent	Last Synch	ValueCount	Fields	Ref(all)	Refs(Config-Tree)	Refs(Config)
1025	my catalog field		13.03.2013 15:40:22	2		1		
2	Country		13.03.2013 15:43:05	11		21	5	5 5
8	Language		13.03.2013 15:43:05	10		14		
49	Currency		13.03.2013 15:43:05	12		55		
492	Person Type		13.03.2013 15:43:05			10		
45	Interest Grp. 1		13.03.2013 15:40:23	4		23		
46	Interest 1	45	13.03.2013 15:40:23	13		23		
51	Co. Type		13.03.2013 15:43:05	7		5		
213	Relationship type		13.03.2013 15:40:22	1		3		
495	Stock Receipt Type		13.03.2013 15:40:23	5		3		
325	Problem	324	13.03.2013 15:40:23			3		
324	Problem Group		13.03.2013 15:40:23	2		3		
481	Service Level		13.03.2013 15:40:23			1		
195	Opp. Type		13.03.2013 15:40:22	1		2		

**FIXED CATALOGS**

CatNr	CatName	Last Synch	ValueCount	Fields
4	Activity Type	13.03.2013 15:40:22	8	3
5	Status	13.03.2013 15:40:22	3	1
8	Month	13.03.2013 15:40:22	13	1
9	Priority	13.03.2013 15:40:22	3	6
33	Status	13.03.2013 15:40:22	6	3
38	Form of address	13.03.2013 15:40:22	3	5
39	Lang. (update CRM)	13.03.2013 15:40:22	30	1
54	Status	13.03.2013 15:40:22	4	3
67	Status	13.03.2013 15:40:22	3	3
68	Priority	13.03.2013 15:40:22	4	1

For each catalog, you see

- **CatNr:** the catalog number.
- **CatName:** the catalog name.
- **Parent:** the number of the parent catalog only applicable for hierarchical catalogs.
- **Last Synch:** date and time when this catalog was last synchronized.
- **ValueCount:** number of catalog entries contained in the catalog.

- **Fields:** number of fields in various info areas where this catalog is used. Using the ♥ button you get more detailed information which info areas and fields are affected.
- **Refs(all):** number of times this catalog is referenced in the designer e.g. in a query in all configurations.
- **Refs(Config-Tree):** number of references of the catalog within the current configuration and its parent configurations
- **Refs(Config):** number of references of the catalog within the current configuration

If you click on the 🖋 button, you get detailed information about the selected catalog:

**Catalog #2 (Country)**

Synchronisation:  Auto  Text/Tenant  Ext. Key

Parent Catalog: -

Apply Changes
Delete Catalog and all references

	CatValId	Value	ExternalKey	Code	Display-Text Usage
Fields:	89	China	1_2_7		7->text
ID 13 Country	<input type="text" value="OK"/> ▼				
FI 5 Country					
KP 19 Country	90	Deutschland	1_2_2		2->text ♥ Query: 1
OR 0 Country	<input type="text" value="OK"/> ▼				
AK 26 Country					
ZA 6 Country	91	Finnland	1_2_8		8->text
PL 52 Country	<input type="text" value="OK"/> ▼				
Y2 12 Country					
II 15 Country	92	Italien	1_2_6		6->text
I7 5 Country	<input type="text" value="OK"/> ▼				
Z6 11 Country					
PE 6 Country	93	Japan	1_2_5		5->text
OJ 6 Country	<input type="text" value="OK"/> ▼				
OB 9 Construction Address Country					
OB 41 Country	94	Kroatien	1_2_11		11->text
OB 68 Prop. Country	<input type="text" value="OK"/> ▼				
LA 0 Country					
	95	Österreich	1_2_1		1->text ♥ Query: 1
	<input type="text" value="OK"/> ▼				
	96	Schweiz	1_2_3		3->text ♥ Query: 1
	<input type="text" value="OK"/> ▼				
	97	Spanien( tenant=200)	1_2_10		10->text
	<input type="text" value="OK"/> ▼				
	98	USA	1_2_4		4->text ♥ Query: 1
	<input type="text" value="OK"/> ▼				
	99	Vereinigtes Königreich	1_2_9		9->text ♥ Query: 1
	<input type="text" value="OK"/> ▼				

For each catalog you can select the **Synchronization** mode defining how a catalog value stored in CRM.Designer is "linked" to a catalog value in the Aurea CRM database (when the catalog value is used by a configuration unit in Aurea CRM.Web and when synchronizing catalogs):

- **Text/Tenant:** Catalog values are linked if they have the same text in the default catalog language plus either the same tenant number or no tenant number.
- **Ext. Key:** Catalog values are linked if they have the same external key.
- **Auto (default):** Uses **Ext. Key** if all catalog values in both systems have an external key, otherwise **Text/Tenant**.

**Note:** Synchronization by external key is useful if catalog values are created on more than one station (example - development, test and live station). If the text in the catalog default language is changed , catalog values can still be synchronized via the external key.

On the catalog details page, you see whether the catalog has a parent catalog and in which info area it is used for which fields. In addition, for each catalog entry you see:

- **CatValId:** the catalog entry's ID.
- A drop-down list where you can specify an **action** that should be applied to this catalog entry (see below for details)

- **Value:** the text value of the catalog entry in the default language.
- **ExternalKey:** the external key of the catalog
- **Display-Text:** the catalog value's text in the language used with the latest catalog synchronization of that value (i.e. the language selected in CRM.bulkloader).
- **Code:** the catalog code from the Aurea CRM database
- **Usage:** informs you how often this catalog entry is used within the designer (example - in a query or as an analysis category). Using the ♥ button you get more details where this catalog entry is referenced.
- **Child-Count:** depicts if and how many values from child catalogs are referencing this catalog. In other words, a parent catalog is not used if both Usage and Child-Count have no value.

**Catalog #324 (Problem Group)**

Synchronisation:  Auto  Text/Tenant  Ext. Key

Parent Catalog: -

Buttons:

Fields:	KM4 Problem Group	CatValId	Value	ExternalKey	Code	Display-Text	Usage	Child-Count
	SI 8 Problem Group	104 <input type="button" value="OK"/>	Bewilligungen	1_324_2		2->text		
	MA 104 Problem Group	105 <input type="button" value="OK"/>	Software	1_324_1		1->text		3
		164 <input type="button" value="OK"/>	Hardware	1_324_3		3->text		5

Apart from the possibility to delete the entire catalog and all references to it from the CRM.Designer database (**Delete Catalog and all references** button), you can apply specific actions to the individual catalog entries:

- "OK" means the catalog entry is not changed.

CatValId	Value	ExternalKey	Code
95 <input type="button" value="OK"/>	Österreich	1_2_1	1

- "Move" merges catalog entries by moving one entry's ID to another ID – all references are moved and the value itself is deleted.

CatValId	Value	ExternalKey	Code
159 <input type="button" value="Move"/>	Move To: 93 <input type="button" value="Move"/>	1_1_2	2

- "Delete" removes the catalog entry i.e. the value itself and all references to it.

CatValId	Value	ExternalKey	Code
158 <input type="button" value="Del"/>	delete value, remove references	1_1_3	3

**Note:** Deleting catalogs that are still in use is not recommended – you should check and modify/delete the queries, filters etc. before deleting the catalog!

- "Edit" allows you to specify the text in the default language plus the corresponding tenant number. Editing a catalog entry is necessary if the default language text of the entry has been changed by the Aurea CRM catalog maintenance.

CatValId	Value	ExternalKey	Code
160	<input type="text" value="Austria"/> Text: <input type="text" value="Austria"/> Tenant: <input type="text" value="0"/>	<input type="text" value="1_1_1"/>	1

## Catalog Maintenance Tasks and Supported Ways

Overview of Catalog Maintenance Tasks and Supported Ways to Reflect Changes in the CRM.Designer Database

- **Editing a catalog value:** You may either manually change the catalog text value using "Edit" in the "Catalogs" section of the CRM.Designer( see [Catalog Maintenance Tasks and Supported Ways](#) on page 726), or you can synchronize afterwards followed by a "Move" to the new value.
  - manually change the catalog text value using "Edit" in the Catalogs section of the designer (see [Catalog Maintenance Tasks and Supported Ways](#) on page 726) , or
  - synchronize afterwards (i.e. after using the Aurea CRM catalog maintenance) and perform a "Move" to the new value

Another way – recommended for multiple changes – is to

- **Moving a catalog value:** Perform the same "Move" action you did in the Aurea CRM catalog maintenance in the Catalogs section of the CRM.Designer, too.
- **Deleting a catalog value:** Perform the same "Delete" action you did in the Aurea CRM catalog maintenance in the Catalogs section of the CRM.Designer, too. Before you do so, it is highly recommended to check – and manually change – all references to catalog values (e.g. filters, mask fields, etc.) that is deleted!

---

**Note: Removing catalog references:** A catalog reference in the CRM.Designer database is stored beside a "normal value" which normally would be the catalog code. When a value with a catalog reference is evaluated, this "normal value" is ignored. If you delete the catalog value to which a reference points, the reference is deleted and the stored "normal value" becomes valid again.

---

## Removing Unreferenced Catalog Values

---

The more variable catalog values are synchronized to the CRM.Designer database, the longer the application start of Aurea CRM.Web takes.

1. On the **Catalogs** page click **Delete all unreferenced values** to remove all unreferenced catalog values from the CRM.Designer database with a single click.

## Roles & CRM Processes

Learn how to make the access to user interface elements simple by the use of Roles & CRM Processes.

The concept of roles & CRM processes makes the administration of access to certain elements (mainly user interface elements) of the CRM suite easier. And configuration adjustments need only be performed once for many users at the same time. If you have configured role- and CRM process-based access, then the system adjusts itself with regards to functionality to the CRM processes that are defined and linked with the role after the user has logged on.

**Example:** If you don't want your users to use the **Person** info area (PE) because you are only dealing with persons in companies (KP), lock CRM process "Person" (CRM process ID 2470000); all related UI elements is hidden.

A **role** defines tasks, properties and most importantly access rights of a user. Roles can be assigned to many different users and thus lower the configuration effort. A **CRM process** describes a series of individual actions that are carried out step-by-step to achieve a business or operative target.

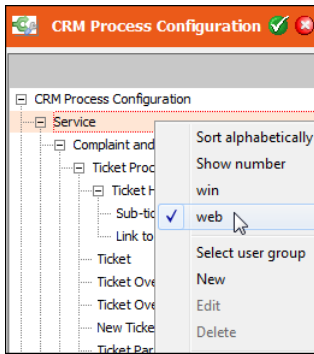
An example for a role would be "marketing employee" and an example for a CRM process would be "execution of a campaign". Depending on the role(s) with which a user logs on to Aurea CRM, he has different CRM processes and possibly different access rights at his disposal.

More information regarding the concept of roles & CRM processes can be found in the *CRM.Core Administrator Guide* ([https://help.aurea.com/crm/crm-core-admin-guide/crm-core-admin-guide/dita/topics/c\\_core\\_admin-1\\_1-Introduction\\_crm.html](https://help.aurea.com/crm/crm-core-admin-guide/crm-core-admin-guide/dita/topics/c_core_admin-1_1-Introduction_crm.html)).

CRM processes and roles are defined exclusively in the **Rights** module in Aurea CRM win, see **CRM Process Configuration** in the *CRM.Core Administrator Guide* ([https://help.aurea.com/crm/crm-core-admin-guide/crm-core-admin-guide/dita/topics/c\\_core\\_admin-1\\_1-Introduction\\_crm.html](https://help.aurea.com/crm/crm-core-admin-guide/crm-core-admin-guide/dita/topics/c_core_admin-1_1-Introduction_crm.html)). In the process tree only values having the setting "web" or "win / web" are relevant for Aurea CRM.Web. If your users use Aurea CRM.Web exclusively, you can hide "win"-only CRM processes for a better overview.

To add a new CRM process for Aurea CRM.Web, you need to check **web** and enter a **Process ID**. This can be any character string and should ideally be descriptive. For further details, see **Adding New CRM Processes** in the *CRM.Core Administrator Guide* ([https://help.aurea.com/crm/crm-core-admin-guide/crm-core-admin-guide/dita/topics/c\\_core\\_admin-1\\_1-Introduction\\_crm.html](https://help.aurea.com/crm/crm-core-admin-guide/crm-core-admin-guide/dita/topics/c_core_admin-1_1-Introduction_crm.html)).

Roles & CRM process defined in the **Rights** module must be transferred to the designer database so that they can be used in CRM.Designer.



## Roles in Aurea CRM.Web

Learn about roles, their configuration and assignment in Aurea CRM.

### Role-Based Access

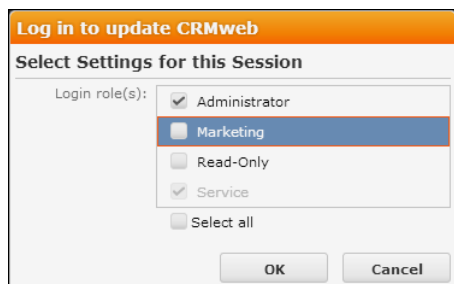
You have the following possibilities to assign roles to a user:

- You can assign roles to users as fixed "rep roles":
  1. Select **Administration > Reps** to access the **Rep** info area (ID). For further details, see [Reps](#) on page 31.
  2. Search for the rep you want to assign a role to.
  3. Select **New Rep role** from the context menu.
  4. Select the desired **Login role**.
  5. Check **Active**.

The assigned role is always active when the user logs on and cannot be changed by the user.

- You can assign several roles to a rep for the user to choose from when logging on:

If a user checks **Select Roles** under **More Options** on the login dialog, he can select from his assigned roles after providing his login credentials.



- Rep roles with **Active** checked cannot be deselected.
- Rep roles with **Default** checked are automatically checked in the selection list.



---

**Note:** To allow users the access the **Select Roles** check box on the login dialog, the `settings.xml` file must contain `<CanChangeRoles>true</CanChangeRoles>` in the `<update.web>` section.

---

**Note:** Roles already be provided in the URL are not displayed in the role selection dialog.

---

- Rep roles can be passed as URL parameters (example -. if single sign-on is configured).

To pass the role in the URL, specify the name(s) of the role(s). Separate multiple roles by comma.

**Example:** `http://<my web address>/crm/start?roles=Admin,Sales`

If a role's name contains special characters (example -. Umlauts), these need to be URL-encoded.

- It is also possible to define in a rep role that access to the CRM processes the role contains is granted exclusively read-only. For further details, see [Read-Only CRM Processes](#) on page 732.

**Changes in the assignment of rep roles** (fixed and optional) are applied as follows: The rep's roles are refreshed only after logout+login.

- without role selection at login: After the time specified in **Cat. refresh frq.** in the **Station configuration** info area

**Default:** 30 min.

- with role selection at login: Immediately after login.

### Information about Active Roles

The menu item **Settings > System Information** lists all roles that are active for the currently logged-in user.

The **SDK** page, **CRMProcessesandRoles** provides an overview over the CRM processes by state, displaying their availability (Read Only/Disabled/Enabled) according to the active role(s) as well as the current info area rights settings. For further details, see [SDK](#) on page 30.

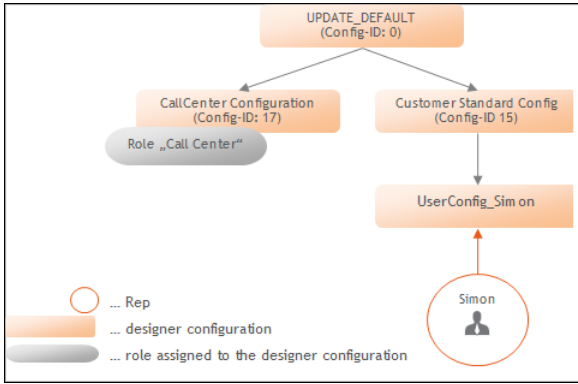
The server log (`web.log`) contains information with which roles and CRM processes a session was started when a user logs on. To get these log entries, set the `logLevel` attribute to "Debug" in the `settings.xml`.

For more information on the server log, see **Server Log** in the *Aurea CRM.Web Installation Guide* ([https://help.aurea.com/crm/crm-web-install-guide/crm-web-install-guide/dita/topics/c\\_web\\_install-1\\_1-Introduction\\_crm.html](https://help.aurea.com/crm/crm-web-install-guide/crm-web-install-guide/dita/topics/c_web_install-1_1-Introduction_crm.html)).

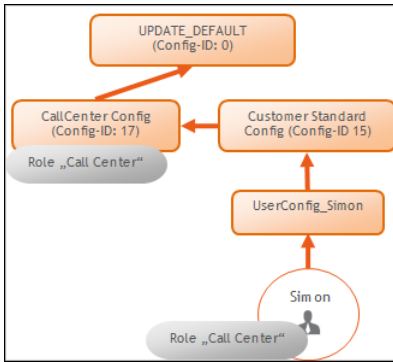
### Dynamic Configuration Assignment Based on Roles

Roles can be assigned to designer configurations. In that case, a user's configuration hierarchy is dynamically determined based on the roles with which he logs on.

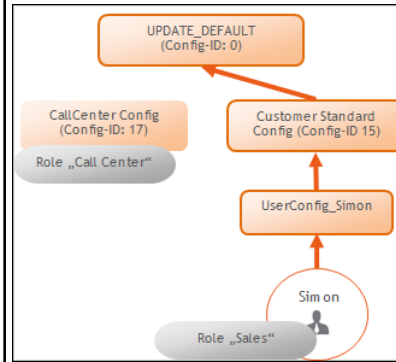
**Example:** Initial situation – designer Configuration



User Simon logs on with the role "Call Center"– configuration 17 (Call Center) is dynamically added to the configuration tree:



User Simon logs on with the role "Sales" – he gets his "normal" configuration tree since there is no configuration assigned to this role:



To assign a role to a configuration:

1. On the CRM.Designer main page click **Users | Configurations**.
2. Select the **Configuration** radio button.

Users / Configurations BB (Business to business) customized\_parent\_config (1001) ->UPDATE\_DEFAULT (0)

Startpage View Users **Configurations**

Root Configurations

Name	Parent Configuration	Role	Description	Users
0 UPDATE_DEFAULT	root	no Role	update CRMweb Default Configuration	none.
2 update.pda	root	no Role	update CRM pda Default Configuration	DefaultUser
6 UPDATE_DEFAULT Offline	UPDATE_DEFAULT	no Role	update CRMweb Default Offline Configuration	none.
7 update.mobile	root	no Role	update CRM mobile Default Configuration	DefaultUser
8 update.touch	root	no Role	update CRM touch Default Configuration	DefaultUser
15 update.tablet	root	no Role	update CRM CRMPad Default Configuration	DefaultUser
16 update.res	root	no Role	update RAS Default Configuration	DefaultUser

Group Configurations

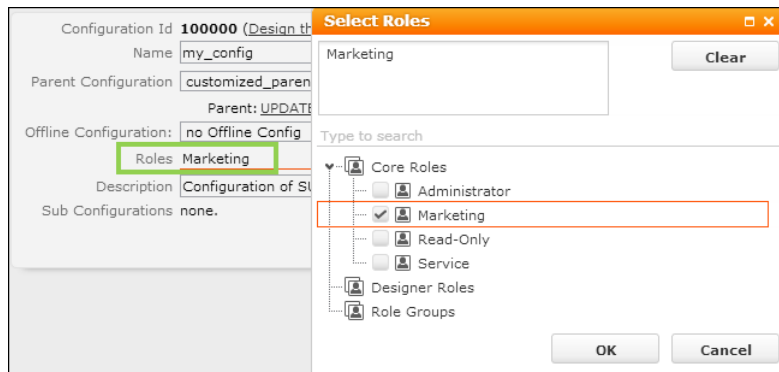
Name	Parent Configuration	Role	Description	Users
1001 customized_parent_config	UPDATE_DEFAULT	no Role		DefaultUser, SU
1002 Maria	customized_parent_config	no Role		MARIA

User Configurations

Name	Parent Configuration	Role	Description	Users
100000 my_config	customized_parent_config	no Role	Configuration of SUSANNA (Child of customized_config)	SUSANNA
100001 User_SYLVIA_ChildOf_customized_parent_config	customized_parent_config	no Role	Configuration of SYLVIA (Child of customized_config)	SYLVIA
100002 User_STEPHEN_ChildOf_customized_parent_config	customized_parent_config	no Role	Configuration of STEPHEN (Child of customized_config)	STEPHEN
100003 User_DORIS_ChildOf_customized_parent_config	customized_parent_config	no Role	Configuration of DORIS (Child of customized_config)	DORIS
100004 User_AKI_ChildOf_customized_parent_config	customized_parent_config	no Role	Configuration of AKI (Child of customized_config)	AKI
100005 User_SIMON_ChildOf_customized_parent_config	customized_parent_config	no Role	Configuration of SIMON (Child of customized_config)	SIMON
100006 User_JOHN_ChildOf_customized_parent_config	customized_parent_config	no Role	Configuration of JOHN (Child of customized_parent_config)	JOHN

3. Click **(Edit)** beside the desired **User / Group Configuration**.

#### 4. In the **Roles** field click (**Select CRM process**).



Use CRM.bulkloader to import the currently defined roles into CRM.Designer. For further details, see [Updating CRM Processes and Roles](#) on page 777.

5. Select the desired roles.
6. Click **OK**.
7. Save.

For each parent configuration of the current configuration it is determined whether for the current role there is a directly dependent configuration that is assigned to the current role. If this is the case, the role configuration is inserted in the configuration tree between these two configurations. If there is more than one role assigned to a session, this happens for each role in the order of their priorities.

For all role configurations determined that way it is also necessary to check recursively if the direct sub-configuration has the same role assigned and in case the whole configuration branch is inserted dynamically.

Each configuration can only exist once in the configuration tree if it exists more than once, only the last occurrence is taken into account.

## CRM Processes in Aurea CRM.Web

Learn about CRM process and their assignment rules.

You can assign CRM processes as defined in the **Rights** module to individual user interface elements. For further details, see [Assigning CRM Processes](#). When a user logs on with a role, he sees only those elements in Aurea CRM.Web that he is allowed to see based on the CRM process definition. Elements that have no CRM process assigned are always visible unless they are hidden based on a rights restriction.

CRM processes can be assigned to the following configuration units:

- menus (context and application menu as well as the assignment to a parent menu)
- menu actions
- buttons
- tabs/groups in a Details control
- tabs in sub-lists (Header configuration for SearchSublist and ExpandChildren)

- nodes in a Tree view
- items in forms (panels, tabs, rows, cards, buttons etc.)
- queries and analyses
- Web Configuration parameters
- processes
- items on `ProcessAsk` pages (except `If...End If`)

In `UPDATE_DEFAULT` CRM processes are assigned to menus, menu actions and buttons.

---

**Note:** Quick Search entries cannot be controlled by assigning CRM processes. You may want to define the Quick Search per role using the `QuickSearch.Definition` Web Configuration parameter. For further details, see [QuickSearch.Definition](#).

---

**Note:** To hide/lock part of a global Quick Search definition, you can use the rights definition (locked info areas are not available in the Quick Search), or if you use a global CRM Process Configuration where all CRM processes are locked by default you can unlock the desired CRM processes per role and thus restrict the Quick Search to the corresponding info areas.

---

The **SDK** page **CRMProcessesandRoles**, provides an overview over the CRM processes by state, displaying their availability (Read Only/Disabled/Enabled) according to the active role(s) as well as the current info area rights settings. For further details, see [SDK](#) on page 30.

### Read-Only CRM Processes

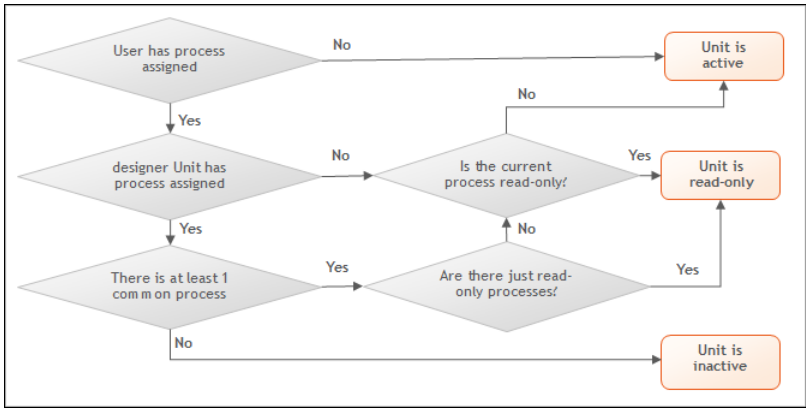
Learn about Read-Only CRM process.

A rep role can define that a user has just read-only access to the CRM processes contained in the role. If a CRM process is read-only, the user cannot modify any data, and furthermore all "dependent" elements are also read-only (i.e. the read-only property is handed down).

Example - If access to marketing campaigns is read-only and a user opens a campaign in details view, not only the campaign but all dependent areas like marketing activities, tasks etc. are read-only as well. This applies even if the user in principle has the right to modify tasks: As soon as he "enters" a read-only CRM process all possibilities contained therein are also read-only.

By default, menu entries and buttons for creating records (no matter if directly or via a process) are hidden when the user is inside a read-only CRM process.

Visibility of designer units based on CRM processes:



The following designer units support read-only mode:

### Menus and Menu Actions / Buttons

The read-only mode can be set either in the action using the `processState` parameter or via the CRM process configuration.

Possible values for argument `processState`:

"ReadOnly"	The action is executed in read-only mode.
NULL or empty	The action is executed in edit mode.
"Disabled"	The action is not available at all.

If a menu is read-only due to its CRM process assignment, the menu item is hidden if `processState` is empty in the action, otherwise the parameter `processState` with value "ReadOnly" is automatically appended.

Menus inherit the `processState` property from the page or the control that shows the menu. All menu actions in such a context menu are read-only, and menu actions with `processState=<empty>` are not shown at all.

### Header

If a header is read-only, all buttons in the header where the associated menu action has `processState=<empty>` are hidden, and all other menu actions are called with `processState=ReadOnly`.

For sub-list headers, the read-only property is passed on to the tabs contained in it.

### Details Control

The Details control is displayed read-only if `processState=ReadOnly` is passed in the action.

Individual tabs can be read-only due to their CRM process assignment.

## List Control

Users cannot edit data in a list if `processState=ReadOnly` is passed in the action.

## Treeview Control

The Treeview control and dependent controls are displayed read-only if `processState=ReadOnly` is passed in the action.

If individual info areas in the tree are read-only because of their CRM process assignment, the whole branch is read-only from there on.

## Form

Individual widgets of a form are read-only if the assigned CRM processes are only available as read-only (**Unit** tab).

The whole form is read-only if the menu is read-only because of the CRM process definition (URL parameter `ReadOnly`). If the action calling the form has `processState=ReadOnly`, the form is read-only as well.

The whole form is read-only if it is located in a tab of a header with `processState=ReadOnly`.

Attention: read-only does not mean that the input controls are automatically no longer editable. It just means that the `ReadOnly` flag is passed on to the Generic Controls using the `oAddInfo` object so that these controls can react accordingly. By default, the read-only functionality is available for the following controls: `DetailsControl`, `ListControl`, `SearchAndListControl`, `HeaderControl`, `TreeControl`.

---

**Note:** The CRM processes where the user has just read-only access are determined right after login.

---

## Roles & CRM Processes in CRM.Designer

Learn about Roles & CRM Processes in CRM.Designer.

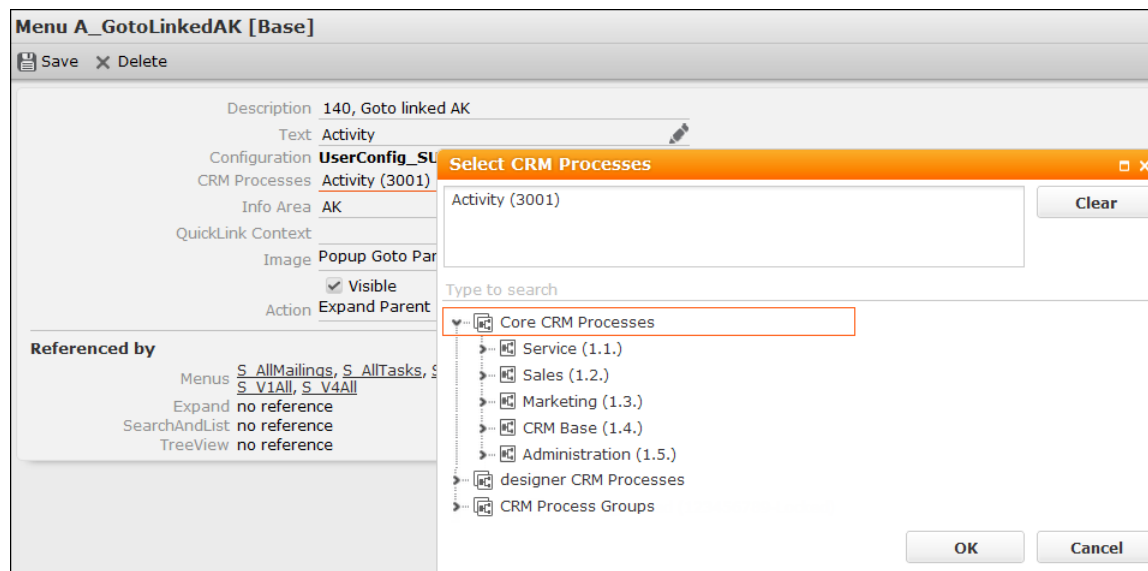
### Transport of Rformoles, CRM Processes and Assignments

The roles and CRM processes defined in the **Rights** module need to be transferred to the CRM.Designer database to be available for use in both CRM.Designer and Aurea CRM.Web. To do this use CRM.bulkloader with the option **Data Model, CRM Processes, Roles and Catalogs**. For further details, see [Updating CRM Processes and Roles](#) on page 777.

CRM process assignments of designer units are transported when exporting/importing designer configurations or packages. The assignment of roles to a configuration is also exported, but is only imported if the configuration does not have a different role assigned in the target database.

## Assigning CRM Processes

All configuration units where CRM processes can be assigned offer the **CRM process tree** in CRM.Designer. All selected CRM processes are listed at the top under **Select CRM Processes** – if you click on such a CRM process, the corresponding position in the process tree is expanded.



The **Core CRM Processes** section contains all CRM processes that were transferred from the Aurea CRM database. If imported CRM processes are not (yet) known to the designer database, they are imported and displayed as **designer CRM Processes**. **CRM Process Groups** exist to make the assignment of multiple CRM processes easier: if you select more than one CRM process, a CRM process group is automatically created (with # at the beginning) containing these CRM processes. Thus it is possible to assign the same group of CRM processes to other units quickly and easily. It is also possible to manually create CRM Process Groups.

Use the search field to search for a CRM process. You can search by name or ID and use wildcards (\*,?).

**Note:** You can use literal character values of '\*' and '?' using the escape sequences '\\*' and '\?'.

To assign a CRM process to a configuration unit, activate the corresponding check box in the CRM process tree.

If two CRM processes are assigned to a configuration unit and only one of them is locked, the configuration unit is available, since a user can log on with two different roles.

## Assigning Roles

For assigning roles to designer configurations, see [Dynamic Configuration Assignment Based on Roles](#).

## Overview of Assigned Roles & CRM Processes

In this topic learn to have an overview of the role and CRM process assigned.

On the CRM.Designer main page (**Processes** area), click the **Roles & CRM Processes** button to open the Roles & CRM Processes overview: This page displays which configuration unit has a role and/or a CRM process assigned in the current configuration.

Roles & CRM Processes

The Roles & CRM Processes tree contains the following nodes:

- **Core CRM Processes/Roles:** All CRM processes/roles available in the CRM.Designer database are listed.
- **designer CRM Processes/Roles:** CRM processes/Roles deleted in the **Rights** module are not deleted when updating CRM processes and roles via CRM.bulkloader but moved to the **designer CRM Processes/Roles** node.
- **CRM Process Groups:** If more than one CRM process is assigned to a configuration unit, a CRM process group is automatically created (with # at the beginning) containing these CRM processes. Click + (**Add**) to manually create a new group.



Roles & CRM Processes BB (Business to business) UserConfig\_SUSANNA\_Sub0 (1033) ->UPDATE\_DEFAULT (0)

Startpage

Type to search

Type: Core CRM Process  
Crm Process ID: 50000320  
Name: Tickets

Assignments Parents

Direct Assignment  All Assignments

Type	Name	Item Inf...	ConfigId	Unit Id	Internal Id
Button	New Proble...		0	75	4
Button	New Ticket...		0	176	4
Button	New Ticket...		0	181	4
Button	New Proble...		0	442	4
Button	New Ticket...		0	446	4
Button	New Ticket...		0	447	4
Button	New Proble...		0	467	4
Button	New Ticket...		0	473	4
Button	New Ticket...		0	474	4
MenuItem	ID NET PR...		0	267	4

« < 1 - 10 of 14 > »

Use the search field to search for a CRM process or role. You can search by name or ID and use wildcards (\*,?).

**Note:** You can use literal character values of “\*” and “?” using the escape sequences “\\*” and “\?”.

Once you select an entry in the tree the overview displays the following information:

- Type, ID and Name of the CRM process/role
- **Assignments** tab:
  - **Direct Assignment:** All configuration units where the selected CRM process is directly – and as the only CRM process - assigned to.
  - **All Assignments:** If multiple CRM processes are assigned to a configuration unit (i.e. the unit is active if a user has access to any one of these CRM processes), the unit is only listed when **All Assignments** is selected together with the direct assignments.

The same logic applies to roles.

- **Parents** tab: For Core CRM processes, the **Parents** tab lists the parents of the selected CRM process.
- **Children** tab: For CRM process groups, the **Children** tab lists the CRM processes belonging to this group.

## Delete Units

Use the Delete Units page to delete configuration units.

**Delete Units** page are used to delete configuration units from the current configuration and/or its child configurations.

Example - Using **Delete Units** you can remove user customizations from child configurations, thus ensuring that changes in the parent configuration are applied to all child configurations.

On the CRM.Designer main page in the **Administration** area, click the **Delete Units** button to open the **Delete Units** page.

A rectangular button with a light gray gradient and rounded corners, containing the text "Delete Units" in a dark gray font.

---

**Note:** If you have authentication enabled (see [Authentication](#) on page 748), the **Delete Units** button is only available for users from **User Group** "Administrator" and "Power User".

---

You can delete configuration units:

- from the current CRM.Designer database, see [Local Deletion](#).
- from another CRM.Designer database using CRM.bulkloader, see [Remote Deletion](#).

Example: If you change the value of a Web Configuration parameter and want to make sure that all sub-configurations use this new value, you can either ensure this by deleting the configuration unit from the sub-configurations if you work on your local CRM.Designer database (Local Deletion), or you can define that the unit is deleted when the configuration is transported to another database i.e. uploaded using CRM.bulkloader (Remote Deletion).

Unit deletion is supported for the following configuration units and sub-units:

• Action Template	• Field Group	• Link Name	• Timeline
• Analysis	• Field Label	• Menu Item	• TreeView
• AnalysisCategory	• Filter	• Process	• Web Configuration Layout
• Analysis Function	• Form	• Process Ask	• WebConfigurationParameter
• Button	• Header	• Process Step	• Web Configuration Value
• Chart Type	• Header Group	• Query	
• Expand	• Image	• Search&List	
• Field Control	• Info Area Label	• Table Caption	

**Local Deletion**

Use the **Local Deletion** tab to delete configuration units from the current CRM.Designer database.

1. On the CRM.Designer main page, select the desired parent configuration.
2. Click **Delete Units**.

The screenshot shows the 'Local Deletion' tab in a CRM interface. At the top, there are two tabs: 'Local Deletion' (active) and 'Remote Deletion'. Below the tabs, there are three dropdown menus: 'Unit Type' (set to 'Field Control'), 'Unit' (set to 'FI'), and 'Sub-Unit' (set to 'Details'). To the right of these dropdowns is an 'Uncheck All' button. The main area contains a table of configurations:

Configuration	Base	Vertical
<i>UPDATE_DEFAULT (0)</i>		#112
<i>Parent (1001)</i>		<input type="checkbox"/> #696
<i>Child1 (1003)</i>		<input checked="" type="checkbox"/> #715
<i>Child2_Nothing (1004)</i>		
<i>Child2_1 (1005)</i>		
<i>User_JL_ChildOf_Child2_1 (100004)</i>		<input checked="" type="checkbox"/> #798
<i>Child3_Everything (1006)</i>		
<i>Child3_1 (1007)</i>		
<i>Child3_1_Sub_Check (1008)</i>		
<i>Child3_1_1 (1012)</i>		
<i>User_ARMINE_ChildOf_Child3_1_1 (100000)</i>		<input checked="" type="checkbox"/> #791
<i>User_DAVID_ChildOf_Child3_Everything (100006)</i>		<input checked="" type="checkbox"/> #802
<i>Child4_Parts (1010)</i>		
<i>User_TOMAS_ChildOf_Child4_Parts (100005)</i>		<input checked="" type="checkbox"/> #801
<i>User_CLINT_ChildOf_Parent (100001)</i>		<input checked="" type="checkbox"/> #793
<i>User_RK_ChildOf_Parent (100002)</i>		<input checked="" type="checkbox"/> #795
<i>User_GOLLUM_ChildOf_Parent (100003)</i>		<input checked="" type="checkbox"/> #797

At the bottom right of the table is a 'Delete' button with a red 'X' icon.

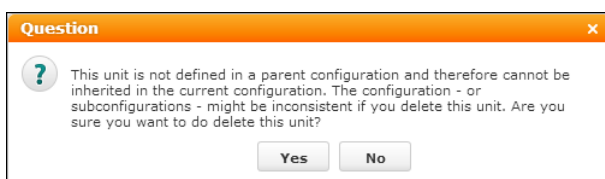
- On the **Local Deletion** tab, select the configuration unit you want to delete: **Unit Type**, **Unit** (i.e. the unit's name) and **Sub-Unit**, if applicable.

All parent and child configurations of the current configuration are listed hierarchically.

For configurations for which the selected configuration unit is configured, the unit's number is displayed either in the **Base** or in the **Vertical** column.

The current configuration is displayed in italics. You can switch to the configuration unit's definition by clicking on the link (unit number).

- Use the check boxes (beside the unit numbers) to select the configurations from which you want delete the selected configuration unit:
  - For parents of the current configuration and for read-only configurations no check box is available since you cannot delete the configuration unit from a parent or a read-only configuration.
  - For the current configuration and its children activate the check boxes for the desired configurations  
**Default:** all children are selected.
  - You can use **Uncheck All** / **Check All** to (un)check all check boxes.
  - If you activate the check box for the current configuration and the selected unit is not defined in a parent, a message is displayed accordingly:



- Click **Delete** to delete the selected configuration unit from the selected configurations.

## Remote Deletion

Use the **Remote Deletion** tab to delete configuration units from another CRM.Designer database by uploading a configuration via CRM.bulkloader.

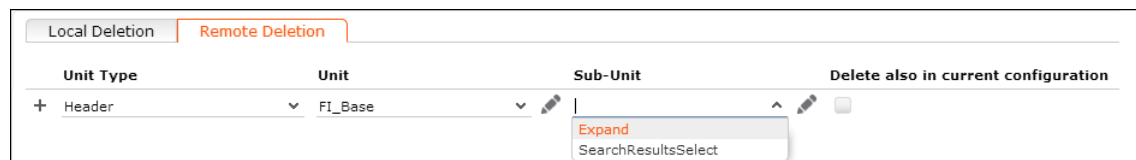
Unit Type	Unit	Sub-Unit	Delete also in current configuration	Date
✗ Action Template	Base		no	2013-05-06 14:27:13
✗ Action Template	Base1		yes	2013-05-06 14:27:19
✗ Header	FI_Base	Expand	no	2013-05-06 15:30:59
✗ Header	FI_Base	SearchResultsSelect	no	2013-05-06 15:31:07
✗ Expand	FI_BB1		no	2013-05-06 17:24:15
✗ Field Control	Double	Matchup	no	2013-05-07 10:32:20
✗ Field Control	Double	List	no	2013-05-07 10:32:43
✗ Field Control	Double	Search	yes	2013-05-07 10:33:21
✗ Field Control	Double	MiniDetails	yes	2013-05-07 10:33:37

Remote deletion procedure:

You change/delete configuration units from your custom parent configuration in the current CRM.Designer database, e.g. the development system. When deploying the customization to the productive system, you define Remote Deletion entries for the outdated / no longer used configuration entries. When uploading the parent configuration to the CRM.Designer database of the productive system, the defined units are deleted.

To remote-delete configuration units:

- On the CRM.Designer main page, select the desired parent configuration.
- Click **Delete Units**.
- On the **Remote Deletion** tab, select the configuration unit you want to delete: **Unit Type**, **Unit** (i.e. the unit's name) and **Sub-Unit**, if applicable.



If the configuration unit has already been deleted from the current database, click **Edit** and manually enter the name of the desired **Unit** (and **Sub-Unit**).

4. Check **Delete also in current configuration**, if you want to remove the selected configuration unit from the parent configuration as well. If unchecked, the unit is only removed from the current configuration's child configurations.

---

**Note:** Before you activate this check box, make sure you really do not need the unit anymore or that this unit is defined in a parent of the current configuration.

---

5. Click + (**Add**) to add the **Remote Delete** entry. The creation date and time are saved as a time stamp (**Date** on the **Remote Deletion** page).

Click **X (Delete)** to delete an entry. Check **Disable single-delete confirmation message box** to disable the message box being displayed each time **X** is clicked.

6. Once you have defined all configuration units you want to delete, download the configuration using CRM.bulkloader. For further details, see [CRM.bulkloader](#) on page 760.
7. Upon uploading the configuration to the target database, the configuration units are deleted from all child configurations of the uploaded configuration and optionally from the configuration itself - **Delete also in current configuration** option.

---

**Note:** Configuration units are only deleted once based on the time stamp: Therefore, units are only deleted the first time a configuration with a respective **Remote Delete** entry is uploaded. Items created or copied after this first upload is not deleted, if you upload the configuration again.

---

---

**Note:** If you want to again delete an item previously set for deletion, you have to create another Remote Delete entry. You can delete the first one, if required.

---

## Consistency Check

Learn about Consistency Check in Aurea CRM.

You can check the consistency of a configuration in the current database. The consistency check is executed on data located in the TMP tables of the CRM.Designer database – this means that the check is either applied to a configuration that was just exported with CRM.bulkloader, or to a configuration that was imported into the TMP tables but not into the database yet. The consistency check also works for packages.

On the CRM.Designer main page in the **Administration** area, click the **Consistency Check** button.

A rectangular button with rounded corners and a light gray background. The text "Consistency Check" is centered on the button in a dark gray, sans-serif font.

To perform the consistency check **for a configuration in your current database**:

1. Export the configuration or package you want to check with CRM.bulkloader. For further details, see [Downloading Configuration Data from the CRM.Designer Database](#) on page 761.
2. On the **Consistency Check** page select the configuration you want to check, which checks to perform and the sort order of the results by unit type or check type.

3. Click **Start**.

To perform the consistency check on the target database **for an imported configuration or package**:

1. Import the configuration (or package) with CRM.bulkloader. Make sure that the **Do not import data (only fill TMP tables)** option is checked!
2. In CRM.Designer (target database) select the configuration you uploaded.

If you uploaded a package, select the target configuration to where the package is uploaded.

The following types of checks are available:

- **Text References:** This check reports invalid text references or references that cannot be found in the destination database.
- **Parent References:** Field controls, headers and texts belong to parent units that need not necessarily be in the same configuration. To upload a control from a field group that is not in the current configuration, the parent configuration and the field group must exist in the destination database; the same applies to headers and texts.

This check reports field controls, headers and texts that cannot be uploaded.

- **Case Sensitivity:** This check reports whether all unit names are unique in a case-insensitive database (i.e. it lists all configuration units that are duplicated when treated case-insensitive).

---

**Note:** This check does not report units on (case-insensitive) SQL databases.

---

- **Empty Names:** This check reports whether empty unit names (i.e. unit names having the value null or having only blanks as value) exist.
- **Trailing Blanks:** This check reports unit names that end with one or more blanks because these trailing whitespaces may get lost on bulk XML transport because such names should be avoided.

## Configuration Validator

Use the Configuration Validator to identify faulty configuration units, migration and configuration issues in the selected configuration.

On the CRM.Designer main page in the **Administration** area, click the **Configuration Validator** button.

Configuration Validator



The screenshot displays the Configuration Validator tool interface. At the top, there is a search bar with the text "Enter a designer unit name or type to filter the list". Below the search bar, there are several expandable sections for different configuration types: Menu Item (10), Action Template (2), Button (11), Wizard (3), Wizard Step (5), Chart Type (1), Web Configuration Parameter (1), Web Configuration Layout (1), and Field Control (8). The Action Template section is expanded, showing two issues: "The action template refers to legacy folder 'mm/' or 'framework/'." (2 issues) and "The action has an empty URL/Function argument." (1 issue). The Field Control section is also expanded, showing several issues related to deprecated code, unsupported field attributes, and deprecated placeholders. The Image section shows one issue: "Unsupported image format." (3 issues).


The page lists two types of problems:

- **Migration issues** (with ID Mxxxx): Configuration units that are broken or use deprecated properties, syntax etc. due to migrating an update.seven designer database to Aurea CRM.
- **Configuration issues** (with ID Cxxxx): General configuration issues, e.g. a button referencing an action template that was deleted from the database.



The configuration units are listed by type. You can filter for specific configuration units using the search field.

Use the  /  (**Expand/Collapse**) buttons to expand/collapse the whole list or all entries of a group (= unit type).

You can filter for errors only by clicking on the  icon beside the search field.

Each issue is displayed in a "bubble" containing the following information:

- Problem type (and description):

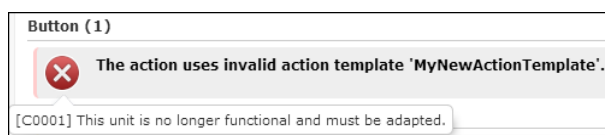


Error, i.e. action required: The unit is broken and needs to be fixed manually.



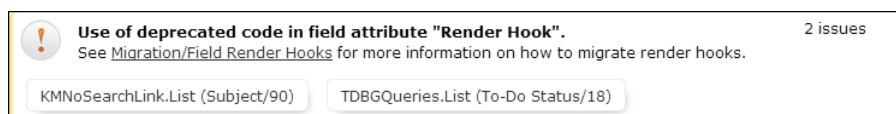
Warning, i.e. action suggested: The unit is still working but uses e.g. deprecated syntax.

- On mouse-over the icon the issue's ID (and a help text) is displayed:



For detailed documentation (description and solution) of all issues (listed by this ID), see the article "Issues Reference" at <https://support.aurea.com>


- A link to a related article (if available) in the Aurea CRM.Web Developer Guide located at <https://support.aurea.com>.
- The number of configuration units concerned ("x issues").
- Clicking on a bubble expands it and displays a link to each configuration unit that needs to be fixed.



The link also displays the affected field (name and number), if applicable, e.g. for render hooks and select-functions.

- For migration issues that cannot be solved a  (**Delete**) button is displayed inside the bubble:



Clicking  (**Delete**) deletes the designer unit from the configuration. This functionality is available for the `Item Attributes` field attribute (issue M0035) and for legacy styles (M0009).

Click  (**Export validation issues**) to export the list as `Configuration issues.csv`.

The export file contains the following information:

- **Unit Type:** Button, Menu Item, Field Control, etc.
- **Unit Name:** The configuration unit's name as defined in CRM.Designer.
- **Sub-Unit Name:** For field controls, the the control type and field name are written to this column.
- **Additional info:** The actual attribute definition (select function, render hook definition etc.) causing the problem.
- **Severity:** Warning or Error.
- **Code:** The issue's ID (Cxxxx or Mxxxx).
- **Message:** The text from the bubble describing the issue.

## Compare Field Controls

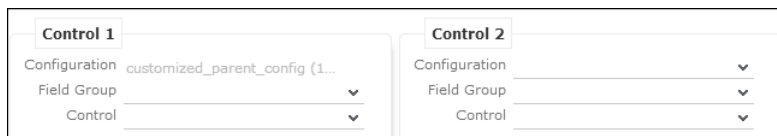
Learn about Compare Field Controls on Aurea CRM.

Use the **Compare Configurations** options to compare field control definitions from different configurations.

Compare Field Controls

To compare control definitions:

1. On the CRM.Designer main page select a configuration.
2. In the **Administration** area click the **Compare Configurations** button.



Control 1	Control 2
Configuration: customized_parent_config (1...	Configuration: [v]
Field Group: [v]	Field Group: [v]
Control: [v]	Control: [v]

3. Select the **Field Group** and **Control** type for the left control.
4. Select the configuration you want to compare under **Control 2**.
5. The **Field Group** is automatically selected, but can be changed.
6. Select the **Control** type for the second control.

---

**Note:** The **Field Group** and **Control** drop-down lists only contain items that are defined in the selected configuration.

---



---

**Note:** You can not compare Default controls.

---

The controls are displayed side by side. Panels are compared by **Name**, fields by ID. The following information is available:

- **+**: The element (panel or field) has been added.
- **✗**: The element has been deleted.
- **🔧**: The element has been modified.
- **↕**: The element has been moved.

On mouse-over a tooltip displays information on the change, e.g. which field attribute was changed etc.

- When selecting an element (panel or field) that was edited, the element's attributes are displayed.
- When selecting an element, all occurrences of the element are highlighted with a colored border. The elements that are currently compared are highlighted with a colored background.

The Compare Configurations page supports the 1-click package functionality (for the current configuration, i.e. Control 1). For further details, see [1-Click Package Management](#) on page 714.

## Authentication

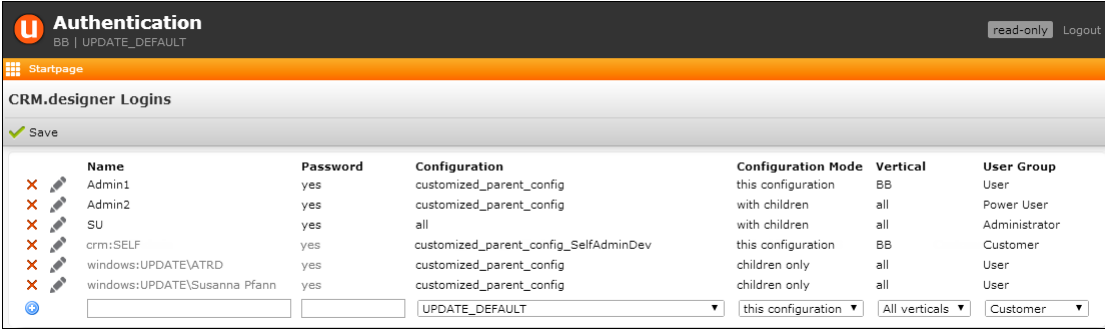
Enable authentication to restrict access to CRM.Designer.

To enable authentication to restrict access to CRM.Designer, see [Settings](#) on page 750. For information on enabling single sign-on for CRM.Designer, see the article “How to Enable SSO in the CRM.Designer” at <https://support.aurea.com>.

On the CRM.Designer main page in the **Administration** area, click **Authentication** to manage CRM.Designer logins.

### Authentication

**Note:** Aurea software strongly recommends enabling CRM.Designer authentication for security reasons as well as for providing meaningful output in the change history. For further details, see [Change History](#) on page 69.



Name	Password	Configuration	Configuration Mode	Vertical	User Group
Admin1	yes	customized_parent_config	this configuration	BB	User
Admin2	yes	customized_parent_config	with children	all	Power User
SU	yes	all	with children	all	Administrator
crm:SELF	yes	customized_parent_config_SelfAdminDev	this configuration	BB	Customer
windows:UPDATE\ATRD	yes	customized_parent_config	children only	all	User
windows:UPDATE\Susanna Pfann	yes	customized_parent_config	children only	all	User

You can define the following:

- **Name:** CRM.Designer login name (max. 8 characters, not case-sensitive).

**Note:** Starting with SP2 login names must be unique. If your CRM.Designer database contains multiple logins with the same name, only the login with the lowest ID is kept when upgrading to SP2.

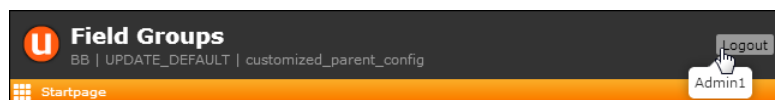
- **Password:** (Re-)set passwords. Passwords are case-sensitive.
- **Configuration:** Specify the configuration a user is allowed to modify.
- **Configuration Mode:** Define whether a user is allowed to change only the specified **Configuration** ("this configuration"), only child configurations of the specified configuration ("children only"), or both ("with children").

**Note:** For "children only" mode, users can see but not modify parent configurations.

- **Vertical:** Defines the vertical version(s) for which a user is allowed to change configurations.
- **User Group:** Depending on the **User Group** the CRM.Designer main page displays a sub-set of available options, >> [Login-Dependent CRM.Designer Main Page](#) on page 749.

Logins and authentication settings are stored in the `PORTAL_LOGIN` table in the CRM.Designer database.

If authentication is enabled, a login screen is displayed when accessing CRM.Designer (except with single sign-on enabled) and a **Logout** link is displayed in CRM.Designer's header:



Alternatively (in a scenario with anonymous access enabled), you can define a fixed configuration and configuration mode via CRM.Designer's `settings.xml` (`<FixedConfigId>` and `<ConfigMode>` tags).

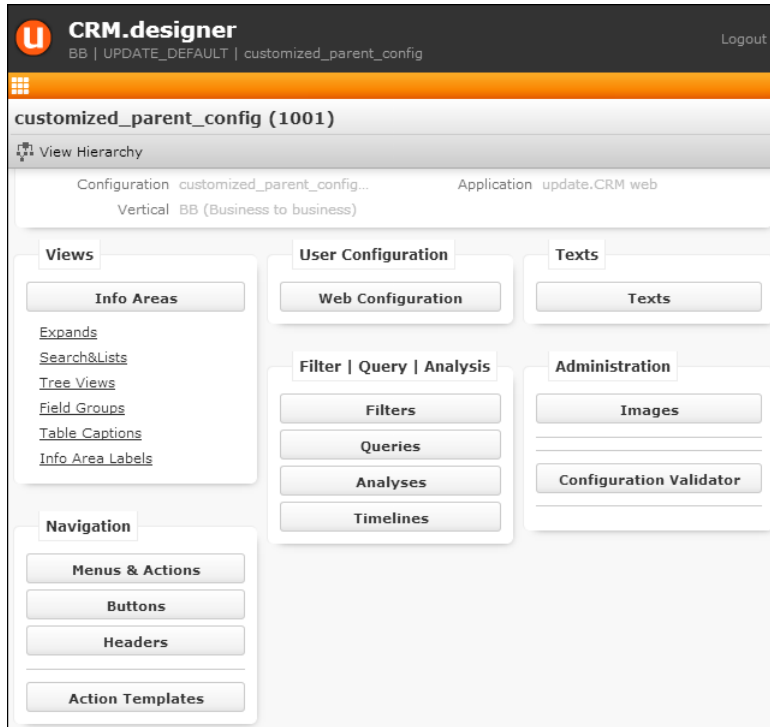
## Login-Dependent CRM.Designer Main Page

Learn about Login-Dependent CRM.Designer Main Page.

Depending on the **User Group** of a login, the user sees a sub-set of options on the CRM.Designer main page. This enables you to implement scenarios where there is a central CRM.Designer administration with full access to all CRM.Designer functions and additional country-specific administration with restricted access.

The following user groups are available:

- **Administrator:** These users can access **all** items on the CRM.Designer main page.
- **Power User:** Can access all items except **Authentication** and **Settings**.
- **User:** Can access only the most commonly used and (relatively) easy to configure items.



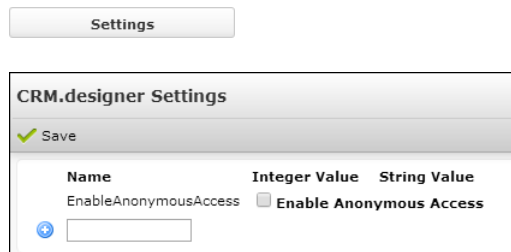
Administrators from this user group need less intensive training compared to administrators working in the other modes, and the risk of damaging a configuration is lower.

- **Customer:** Can only access the **Field Groups** page. This user group is used for customer administrators in an SaaS environment. For further details, see [Customer Admin Mode](#) on page 865.

## Settings

Use the **Settings** page of CRM.Designer to enable/disable authentication.

On the CRM.Designer main page in the **Administration** area, click the **Settings** button to open the **Settings** page.



If **Enable Anonymous Access** is checked (or undefined), CRM.Designer is accessible without login.

If **Enable Anonymous Access** is not checked, only users specified on the **Authentication** page are allowed to access CRM.Designer.

The setting is saved in the `PORTAL_SETTINGS` table.

**Note:** To apply a change of **Enable Anonymous Access**, you need to recycle CRM.Designer's application pool.

---

Alternatively, you can control access to CRM.Designer by commenting out the `<EnableAnonymousAccess />` entry in the `settings.xml` file. The setting in the `settings.xml` is overruled by the setting in the CRM.Designer database.

# 4

## Other Programs and Functions

---

Explore additional capabilities offered by Aurea CRM solution.

Following are the additional feature offered by Aurea CRM:

### Affiliation View

The Affiliation View allows users to view data linked to a set of companies and/or persons.

Example - All health packages subscribed by the members of a family.

To activate this functionality, you need to:

- activate CRM process 1.4.1.4.7. "Affiliations" (CRM process ID 70000260): This enables all necessary UI elements in Aurea CRM.Web. For further details, see [Configuration Units](#) on page 754.
- specify the info areas (linked to FI, PE or KP) from which collective data should be displayed via an MC entry (**Affiliations** category, **Applies to the following info areas** option). For further details, see [Configuration Info Area](#) in the *CRM.Core Administrator Guide*.

Affiliations are defined via **Relationship** records (PB). By default, the **Affiliate** flag in the PB record defines a related company/person as an affiliate. For further details, see [Alternative Condition for Affiliation View](#) on page 757.

Indirect links (IDs 26000-27999) allow reading child records of companies/persons based on these PB records. For further details, see [Link IDs](#) in the *CRM.Core Administrator Guide*.

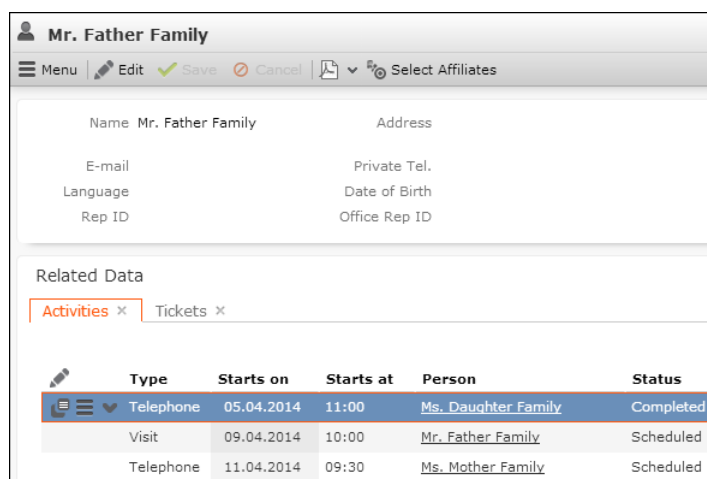
For general information on [relationships](#), see [Web Business Logic User Guide](#).

Users can add/remove companies and persons (FI, KP, PE and CP records) to/from affiliation and immediately refresh the page. Selecting a record in the graph and clicking on **Add to Affiliation** sets the **Affiliate** flag in the PB record. (Selecting a record and clicking **Remove from Affiliation** removes the flag. Multiple selections (via Ctrl+Click) is supported.





After clicking **Close&Refresh**, the user sees child records (e.g. activities) for all members of the current affiliation in the Related Data area of Expand and List views, in Tree and Timeline views:



**Important:** Affiliation View is an option that effects all queries in CRM Web. Heavy queries (because of a lot of links) combined with cascading update from triggers can cause transactions to lock each other. Queries involving a lot of records with multiple table concentrations on every page slows down the performance of the CRM.Web application. To improve the performance you need to create indexes and optimize them. Sorted queries requires sorted indexes on relevant fields.

**Note:** MSSQL suffers from a performance degradation for certain types of queries when affiliation view is enabled. In such cases, using queries that force merge joins

by using the `OPTION (MERGE JOIN)` query hint is useful. If you are using the PB-MB join then enable the MSSQL specific MC core configuration parameter `Enable merge join query hint for MB join with PB performance flag` to improve performance. For more information on setting this parameter, see [Configuration Info Area](#) in the *CRM Core Administrator Guide*.

You can define fields to be displayed in sub-lists only when the Affiliation View is active by defining `showOnlyForAffiliations=true` as Extended Option. This allows you to provide context information, e.g. the affiliated company to which the activity is actually related. For further details, see [Extended Options](#) on page 228.

## Configuration Units

This topic describe the items available for users with access to the CRM process 1.4.1.4.7. "Affiliations" - CRM process ID 70000260.

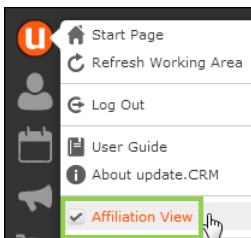
All texts are defined in the `Associations` text group.

### Global Switch / Query-Specific Switch

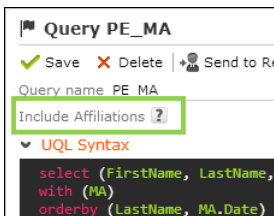
The Aurea CRM menu (`$System`) contains the dynamic menu item `D_ToggleAssociationView` (**Affiliation View**), allowing the user to globally switch the Affiliation View on or off.

To determine the initial default state of the global switch, create and set the `Database.AssociationMode.Default` Web Configuration parameter, see [Database.AssociationMode.Default](#). The user's selection overrules this default.

The user's selection is saved between sessions (in the user's roaming settings), see [User Settings](#) on page 25.



The Query Editor in Aurea CRM.Web as well as in CRM.Designer contains a separate **Include Affiliations** switch. This enables users to define and execute queries independently of the state of the global switch.



The global switch can have the following states:

- **checked:** `IncludeConfigured`, i.e. records from the info areas specified in the MC entry **Affiliations** category, **Applies to the following info areas** option are read for all members of an affiliation (instead of just the current company/person).
- **unchecked:** `Default`, i.e. affiliations are not taken into account when displaying child records.

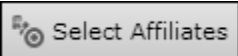
The query-specific switch can have the following states:

- **Pass-through (question mark):** `Default`; when the query is executed the setting of the global switch takes effect.
- **checked:** `IncludeConfigured`, i.e. the query is executed taking affiliations into account (for the child info areas specified in the MC entry), i.e. the indirect PB links (link IDs 26000-27999) are used instead of the links defined in the query statement and regardless of the current setting of the global switch.
- **unchecked:** `Exclude`, i.e. the query is executed using the links specified in the query statement.

For background queries the setting to be used is retrieved from the state of the global and query-specific switches at the moment of submitting the query. It is stored in the XML file as a numeric value:

- `<UseAssociations> 2 </UseAssociations>` = `IncludeConfigured`
- `<UseAssociations> 4 </UseAssociations>` = `Exclude`
- If pass-through is set with the global switch unchecked or unavailable, no `<UseAssociations>` tag is saved, i.e. the links specified in the query statement are used.

## Select Affiliates (Button and Dialog)

The FI and PE `Expand` headers contain the  (**Select Affiliates**) button `SelectAssociationMembersGraph` which calls the `A_SelectAssociationMembersGraph` context menu action.

Clicking the button opens the **Select Affiliates** popup displaying the current record and all companies/persons linked to it via a relationship record. The graph uses the `CircleOfInfluence` widget (limited to one level, i.e. "relationships of relationships" are not displayed).

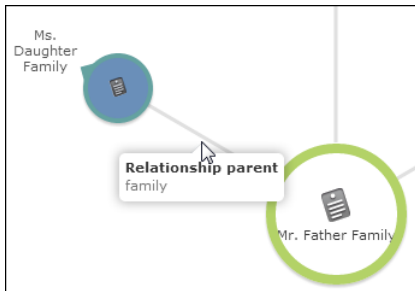


To filter the displayed records (e.g. by relationship type), add a `fixedFilterName` argument to the menu action containing a filter on PB. For further details, see [fixedFilterName](#).

The header displays the default table caption and MiniDetails for the currently selected record (not configurable).

The header buttons `AddToAssociation` and `RemoveFromAssociation` allow users to add/remove companies/persons to/from the association.

On mouse-over the lines connecting the items the `PB.Associations` table caption is displayed in a tooltip (fallback: `PB`).



The labels of the company/person bubbles are defined by `FI/PE/KP/CP.Associations` table captions (fallback: `FI`, `KP`, etc.)

Clicking **Close&Refresh** saves the defined affiliates and refreshes the working area.

Clicking **Close** saves the defined affiliates without refreshing the working area.

### Select Affiliates List

To display a list of PB records instead of the graph, you can use the `A_SelectAssociationMembers` menu action.

---

**Note:** This menu action is deprecated. Use `A_SelectAssociationMembersGraph` instead.

---

Company	Person	Relationship type	Relationship	In association
Gabrieli	Mr. Giovanni Gabrieli		uncle	<input type="checkbox"/>
Merulo S.p.A.		business	partner	<input checked="" type="checkbox"/>

Users can select/deselect affiliates by checking/unchecking the **Affiliate** flag.

The dialog uses the `PB.SelectAssociationMembers` Search&List configuration and field group. To use another Search&List configuration, add a `searchAndListName` argument to the menu action. For further details, see [searchAndListName](#).

To filter the displayed records (e.g. by relationship type), add a `fixedFilterName` argument containing a filter on PB to the menu action. For further details, see [fixedFilterName](#).

## Alternative Condition for Affiliation View

Learn about the alternative condition for affiliation view.

You can use any other field from PB (instead of the **Affiliate** flag) by defining a Z5 variable `CORE::INDIRECT-PB-LINK` of type "Condition":

Variable configuration details:

- Name: CORE::INDIRECT-PB-LINK
- Info Area: Relationship
- Field Name: FreeC1
- Type: Condition

Variable Value table:

Country	Var. Text	Var. Number/Catalog
	F24='is connected with'	

In this example, adding a related company or person to an affiliation sets **FreeC1** to the specified value.

Restrictions for the variable definition:

- only '=' is supported as operator (exception: for Boolean fields '<> N' is supported as well as '= Y')
- empty values are not supported
- the catalog code '0' (i.e. the default value of catalog fields) is not supported
- wildcards are not supported

---

**Note:** You can use literal character values of '\*' and '?' using the escape sequences '\\*' and '\?'.  


---

## Campaign Graph

The CampaignGraph widget displays a CRM campaign and all its marketing activities, responses and follow-up marketing activities in a graphical, hierachical way.

The screenshot shows the 'Campaign Sommer CM the second' interface. At the top, there is a menu bar with options: Menu, Edit, Save, Cancel, Calculate, and Print. Below the menu, the campaign details are displayed: Camp. Type SommerAktion, Camp. Name Sommer CM the second, Resp. Rep ID, Camp. Start 01.07.2066, Event Location, Camp. Status Scheduled, Camp. End 31.12.2066, and Calculation Date. The interface is divided into several tabs: Points, Activities, Mailings, Work Orders, Costs, and Approval & Verification. The 'Points' tab is active, showing 'Act. Total Pts.', 'Targ. Total Pts.', and 'Total Pts. Perc.' (0,00%). The 'Related Data' section is open, showing a hierarchy of marketing activities and responses. The main campaign 'Sommer CM the second' is at the top, with four main branches: Alpha Sommer, Variation; Bravo Sommer, Variation; Charly Sommer, Variation; and Delta Sommer, Variation. Each branch has sub-branches: Alpha (Alpha Sommer), Alpha (Bravo Sommer), Bravo (Bravo Sommer), Alpha (Charly Sommer), Alpha (Delta Sommer), Bravo (Delta Sommer), Charly (Delta Sommer), and Echo Sommer, Variation. The Echo Sommer, Variation branch has sub-branches: Alpha (Echo Sommer) and Bravo (Echo Sommer).

For campaigns and marketing activities the **Graph** tab (configured in both the `SearchSubList` and `ExpandChildren` headers of CM and AK) displays a graph of the selected campaign containing the campaign, its marketing activities and responses.

Each item displays the xs info area image (see [Images](#) on page 717) and a table caption. You can define special table captions `<InfoAreaId>.Graph` to be displayed for the items of the graph (CM, AK, AN).

For information on the `CampaignGraph` widget, see the article “How to use Campaign Graph” at <https://support.aurea.com>.

## Creating/Updating Records From email Signatures/Websites

This topic describes the configuration and extension possibilities of the Signature Parser functionality in Aurea.CRM

You can configure a button allowing users to create/update company/person records directly from email signatures or websites.

**Note:** The actual signature parser is not part of Aurea CRM.Web but must be implemented separately (using either your own parsing logic or obtaining a signature parser from a third party provider).

The parsing results (key/value pairs as JSON) are mapped to fields in the Aurea CRM database via field functions.

Users copy a signature or address information from the source. In Aurea CRM.Web (e.g. in the `New` header for FI) they click a button (**Create from signature** in the above example) calling the **Copy from Signature or Website** dialog. For further details, see [Configuration Units](#) on page 759. Pasting the signature/imprint in the text field and clicking **Analyze** calls the parser. The result is displayed in the fields below the text box.

Users can manually correct the mapping between values and fields.

Invalid catalog values are marked with a red exclamation mark (plus tooltip) allowing the user to select an existing catalog value or create a new one.

Clicking on **Apply** transfers the values to the record. (Empty values are not transferred.)

## Configuration Units

Learn to configure signature parsers.

To make the signature parser available in Aurea CRM.Web, the following configuration units are required:

### Signature Parser Action

Create a button with action template `Command set to DataQuality/SignatureParser`. Add it to the desired headers, e.g. the `New` and `Expand` headers for FI, PE etc.

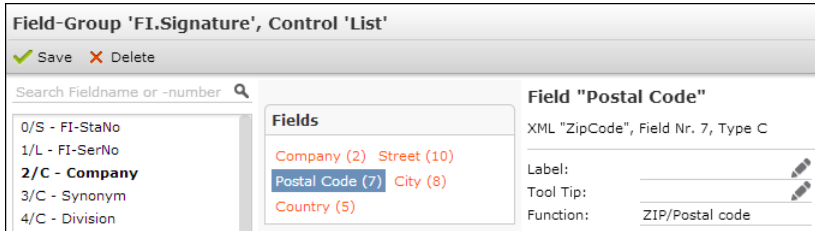
---

**Note:** Buttons calling the signature parser action are only active when in Edit mode.

---

## Field Mapping

Create a field group `<infoAreaId>.Signature` containing a List control.



This List control defines

- the mapping between the parsing result and the fields in the Aurea CRM database: Enter the key of the respective value as **Function** for each field in the control.
- the fields displayed in the **Copy from Signature or Website** dialog.

Supported field types: character, fixed/variable catalog, numeric

---

**Note:** Hierarchical catalogs are not supported.

---

## Parsing Modes

If you use multiple signature parsers (e.g. per language, or one parser for company data, one for person data, etc.), create a Web Configuration parameter `Expand.SignatureParsingModes` as Combobox and add your parsing modes as values. For further details, see [Expand.SignatureParsingModes](#).

Users can select the desired parsing mode from the **Use rules for** drop-down list.

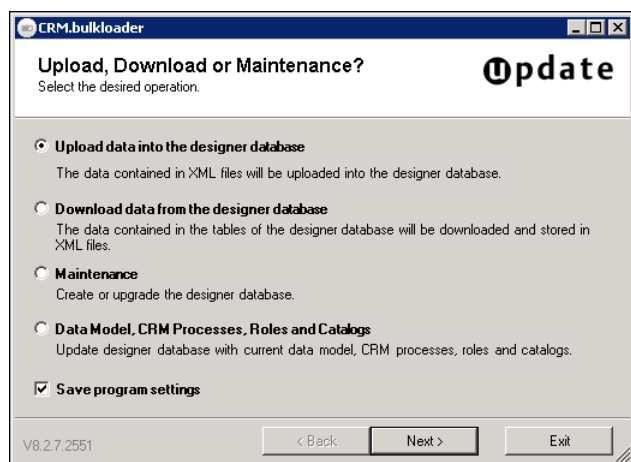
## CRM.bulkloader

Learn to use CRM.bulkuploader to perform a variety of tasks.

Use CRM.bulkloader (`..\designer\database\update.bulkloader.exe`) to

- up-/download CRM.Designer configurations (XML files) and packages
- create or upgrade the CRM.Designer database
- update the CRM.Designer database with the current data model, CRM processes, roles and catalogs from the Aurea CRM database.





Check **Save program settings** to save the selected options (file paths, database connection settings etc.) between sessions. The password is encrypted. The settings are saved per user and version at:

```
%USERPROFILE%\AppData\Local\update_Software_AG\update.bulkloader.*\version\user.config
```

You can access all CRM.bulkloader functions via the command line. For further details, see [CRM.bulkloader Command Line Parameters](#) on page 782.

All actions performed by CRM.bulkloader are logged in the `update.bulkloader.log` file located in the `..\designer\database` directory.

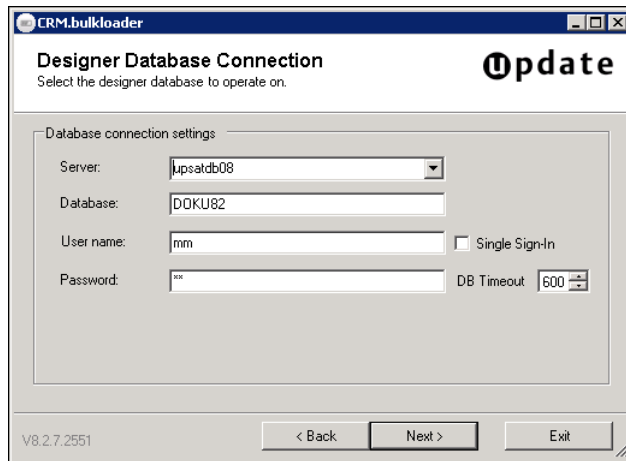
## Downloading Configuration Data from the CRM.Designer Database

Learn to download Configuration Data from the CRM.Designer Database.

You can download CRM.Designer configurations, packages (see [Packages](#) on page 711) or a complete CRM.Designer database ("database dump") from your CRM.Designer database as XML files.

To download configuration data from the CRM.Designer database:

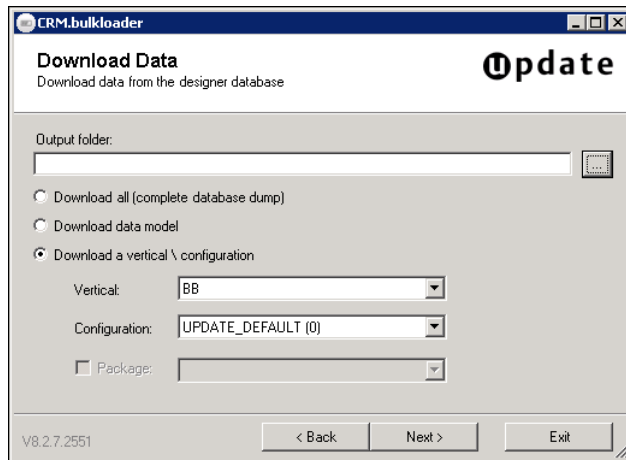
1. On the CRM.bulkloader main page select **Download data from the designer database**.
2. Click **Next**.



3. Enter the connection settings to your CRM.Designer database:

- **Server:** Name of the server where the CRM.Designer database is located. Enter "localhost" or "(local)" if you operate on a local database.
- **Database:** Name of the CRM.Designer database
- **User name/Password:** Account of the database owner, i.e. user with "db\_owner" rights for the CRM.Designer database.
- **Single Sign-In:** If checked, you are connected using the credentials of your current Windows login. (The **User name** and **Password** fields are disabled.)
- **DB Timeout:** The database timeout in seconds.

4. Click **Next**.



5. Enter the path to the **Output folder** (for the XML files containing the configuration) or browse to the desired location.

---

**Note:** The output folder must already exist.

---

## 6. Select the desired download function:

- **Download all (complete database dump):** Creates a complete database dump, i.e. downloads the contents of a whole CRM.Designer database. You can create database dumps for analysis or debugging purposes. You can also upload a database dump to another (empty) CRM.Designer database. For further details, see [Database Dump](#) on page 767.
- **Download data model:** Downloads the Aurea CRM data model as currently present in your CRM.Designer database.
- **Download a vertical \ configuration (default):** Downloads the selected **Vertical** and **Configuration**.

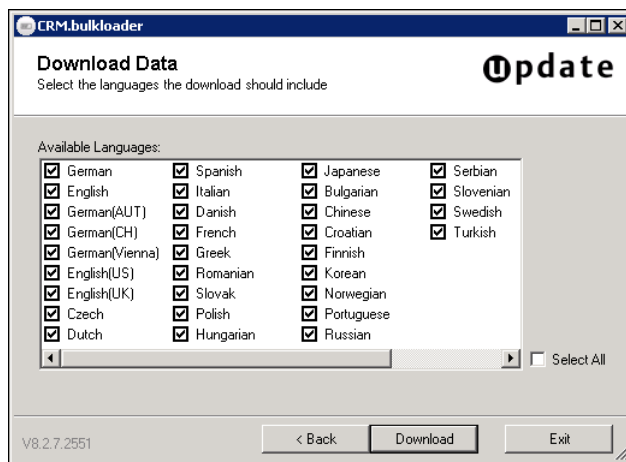
If packages are defined in the selected vertical and configuration, the **Package** check box is checked and you can select a package to download. For further details, see [Packages](#) on page 711.

---

**Note:** If you try to download a configuration from an invalid or outdated CRM.Designer database, you are automatically redirected to maintenance mode. For further details, see [Maintenance](#) on page 768.

---

## 7. Click **Next**.



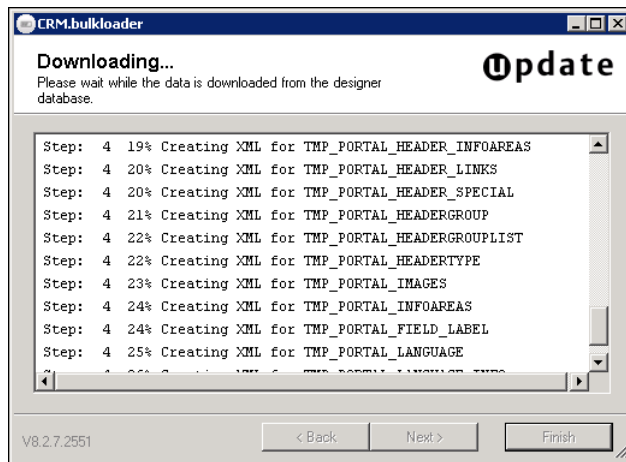
8. Select the languages you want to include in the configuration (or package) download. The default selection reflects the **Active** flag on the **Global Language Settings** page. For further details, see [Global Language Settings](#) on page 173.

---

**Note:** Download at least two languages, even if you need one. The additional language does not have a significant impact on the size of the XML files and the software. If the XML files contain only one language, on uploading CRM.bulkloader assumes that you are uploading a language pack. For further details, see [Uploading and Downloading Texts](#) on page 185.

---

9. Click **Download**. The download process is launched. A dialog box displays the progress of the download.



10. When the upload is completed, a message is displayed. Click **OK**.

## Uploading Configuration Data to the CRM.Designer Database

Learn to upload Configuration Data from the CRM.Designer Database.

You can upload CRM.Designer configurations, packages (see [Packages](#) on page 711) or a complete CRM.Designer database ("database dump") to your CRM.Designer database.

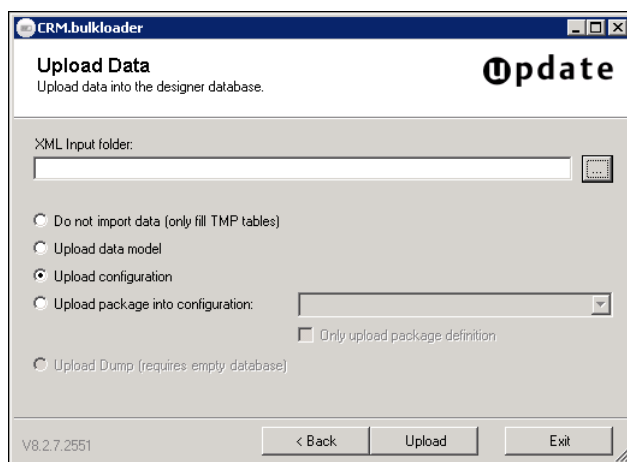
---

**Note:** It is strongly recommended to back up the CRM.Designer database before uploading configuration data.

---

To upload configuration data to the CRM.Designer database:

1. On the CRM.bulkloader main page select **Upload data into the designer database**.
2. Click **Next**.
3. Enter the connection settings to your CRM.Designer database as described above. For further details, see [Downloading Configuration Data from the CRM.Designer Database](#) on page 761.
4. Click **Next**.



5. Enter the path to the **XML Input folder** (containing the configuration as XML files) or browse to the desired location.
6. Select the desired upload function:
  - **Do not import data (only fill TMP tables):** Uploads the configuration into temporary tables only. The XML files are uploaded into temporary tables to check the results before overwriting an existing configuration. For further details, see [Consistency Check](#) on page 742. To finally upload the configuration, select **Upload configuration**.
  - **Upload data model:** Used when first installing Aurea CRM.Web and CRM.Designer. For further details, see [Setting Up the CRM.Designer Database](#) in the *Aurea CRM.Web Installation Guide*.
  - **Upload configuration** (default): Uploads the configuration (located in the **XML Input folder**) into your CRM.Designer database. If you selected **Do not import data (only fill TMP tables)** for a previous upload, **Upload configuration** uploads the configuration from the TMP tables.
  - **Upload package into configuration:** Uploads a package or a package definition. For further details, see [Packages](#) on page 711.
  - **Upload Dump (requires empty database):** Uploads a complete database dump. For further details, see [Database Dump](#) on page 767. This option is only available for empty CRM.Designer databases.

---

**Note:** If you try to upload a configuration to an invalid or outdated CRM.Designer database, you are automatically redirected to maintenance mode. For further details, see [Maintenance](#) on page 768.

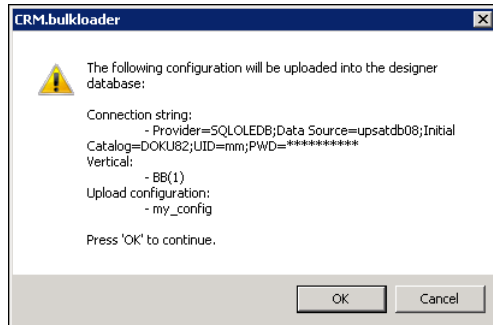
---

**Note:** If you try to upload a configuration that does not exist in the target database, the system confirms whether the new configuration should be created by CRM.bulkloader. Example - when transferring a configuration from your test environment to your production system.

However, configurations are created only if its parent configuration already exists in the target database.

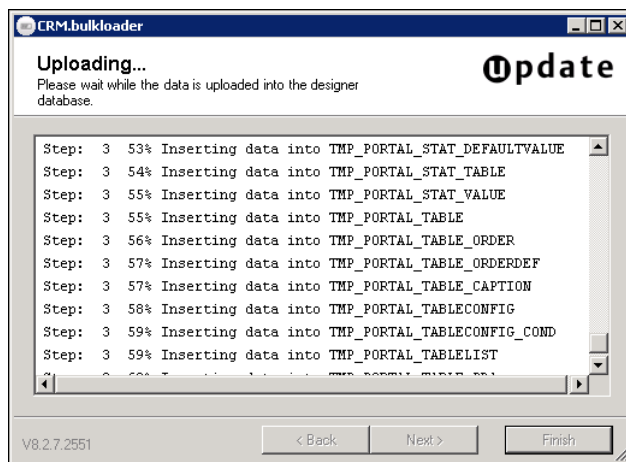
**Note:** If more than one configuration with the same name as the configuration you are trying to upload exists, the operation is aborted with an error message.

- Click **Upload**. A confirmation message displayed the database connection and the items that is uploaded.



**Note:** It is strongly recommended to back up the CRM.Designer database before uploading configuration data.

- Click **OK**. The upload process is launched. A dialog box displays the progress of the upload. The upload process can take a few minutes depending on the amount of data, the speed of the machines and network, etc.



- When the upload is completed, a message is displayed. Click **OK**.

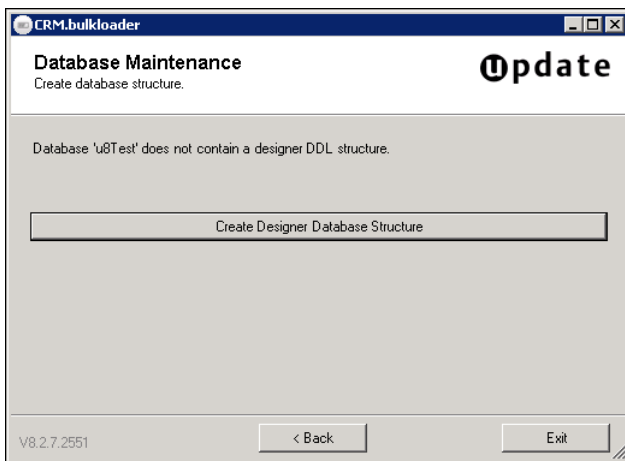
## Database Dump

Learn to create database dump.

When downloading data from a CRM.Designer database, you can also select to download a complete database dump, i.e. the contents of a whole CRM.Designer database e.g. to provide information for a support call.

You can upload such a database dump into an empty CRM.Designer database:

1. Create a database dump with CRM.bulkloader. For further details, see [Downloading Configuration Data from the CRM.Designer Database](#) on page 761 (**Download all (complete database dump)** option).
2. Upload the dump into a new (empty) database:
  - When uploading via CRM.bulkloader UI you are first asked to create the database structure:



Select **Create Designer Database Structure**.

Click **Next**.

Select **Upload data into the designer database**, see [Uploading Configuration Data to the CRM.Designer Database](#) on page 764 (**Upload Dump (requires empty database)** option).

---

**Note:** The dump upload feature is only available for empty databases: Do not upload the UPDATE\_DEFAULT configuration (or any other root configuration) before uploading a database dump.

---

## Uploading a Root Configuration

Learn to upload a root configuration.

When a root configuration is imported, the data model in the CRM.Designer database is upgraded - but no deletion of tables and fields takes place because standard root configurations typically do not contain generic fields.

If tables/fields are contained in the root configuration, but not in the CRM.Designer database, they are automatically added using the next free table ID/field ID. The root configuration to be imported is then migrated to match the data model in the destination designer database.

As with importing a data model, deleted fields are reused if contained in the imported configuration.

## Uploading a Sub-Configuration

Learn to upload a sub-configuration.

When a sub configuration is imported, the data model of the designer is not affected. Instead, the imported data is "migrated" to the data model of the destination CRM.Designer database.

Importing a sub configuration works similar to importing a root configuration – the difference is that tables and fields which are not known to the destination designer database are created (using the next free table ID/field ID) **and** marked as deleted (since these fields are not valid for the current data model).

## Maintenance

Use the CRM.bulkloader maintenance mode to create or upgrade your CRM.Designer database.

---

**Note:** Creating or upgrading a database changes the structure of the selected database. Therefore, it is strongly recommended to back up the CRM.Designer database before upgrading it.

---

---

**Note:** Before upgrading the CRM.Designer database you need to restart the web application (iisreset).

---

---

**Note:** Concerning packages: After upgrading the CRM.Designer database you need to re-generate all packages (i.e. download them again), otherwise changes done by CRM.bulkloader is lost in the target database.

---

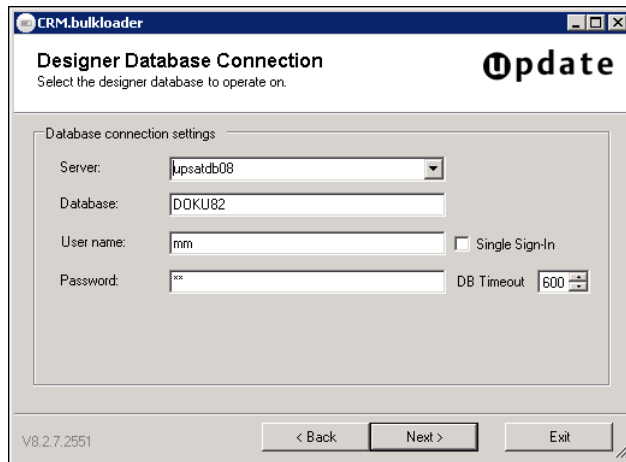


---

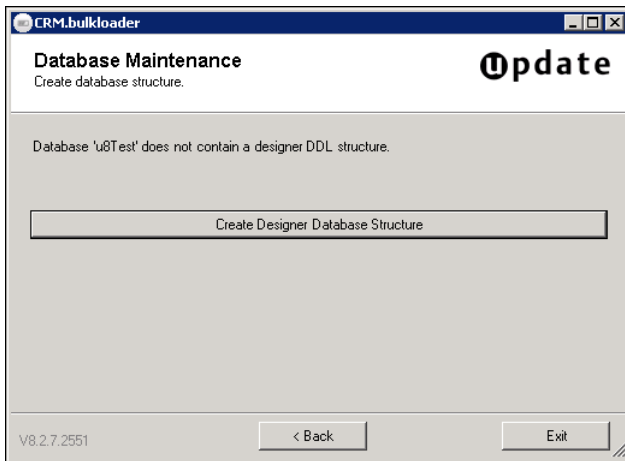
**Note:** When upgrading from an update.seven designer database, CRM.bulkloader performs a few "clean-up" operations; for details, see Migration Guide from Version 7 to Version 8 available from <https://support.aurea.com> (core (win) section).

---

1. On the CRM.bulkloader main page select **Maintenance**.
2. Click **Next**.

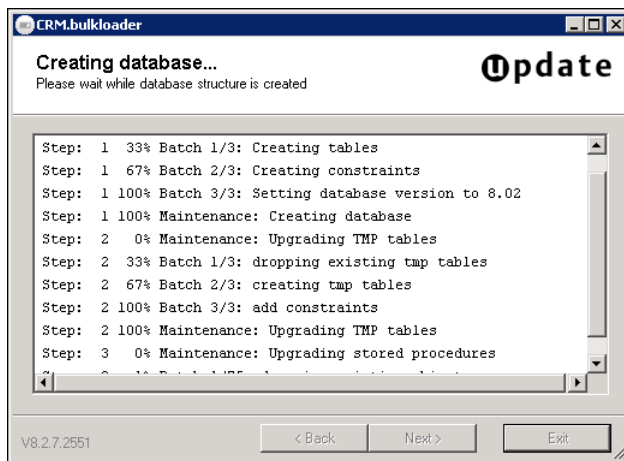


3. Enter the connection settings to your CRM.Designer database:
  - **Server:** Name of the server where the CRM.Designer database is located. Enter "localhost" or "(local)" if you operate on a local database.
  - **Database:** Name of the CRM.Designer database
  - **User name/Password:** Account of the database owner, i.e. user with "db\_owner" rights for the CRM.Designer database
  - **DB Timeout:** The database timeout in seconds.
4. Click **Next**. CRM.bulkloader analyses the database. Depending on the result, one of the following screens is displayed:
  - The database is not a valid CRM.Designer database:

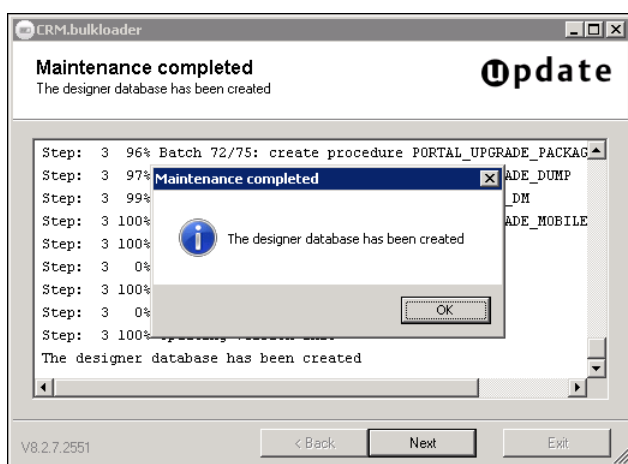


1. Click **Create Designer Database Structure**.

CRM.bulkloader executes built-in SQL scripts creating the database:



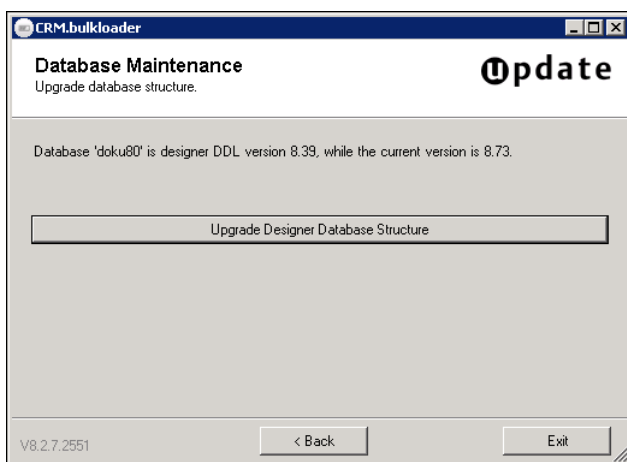
Once the database and database structure has been created, a message is displayed.



2. Click **OK**.

3. Click **Next** to return to the CRM.bulkloader main page.

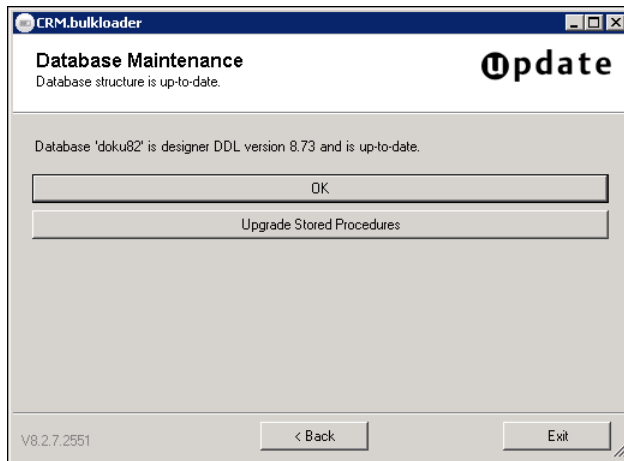
- The database is a valid CRM.Designer database but from a previous version:



**Note:** It is strongly recommended to back up the CRM.Designer database before upgrading it. Before upgrading the CRM.Designer database you need to restart the web server (iisreset).

Click **Upgrade Designer Database Structure**. CRM.bulkloader upgrades the database.

- The database is up to date:
  1. Click **OK** to return to the CRM.bulkloader main page.



2. Click **Upgrade Stored Procedures** to upgrade the stored procedures.

## Data Model, CRM Processes, Roles and Catalogs

Use CRM.bulkloader to update the CRM.Designer database with the current data model, CRM processes, roles and catalogs from the Aurea CRM database.

---

**Note:** It is strongly recommended to back up the CRM.Designer database before updating it.

All information depending on formats (data model, rights etc.) in Aurea CRM win is only applied to Aurea CRM.Web after refreshing the formats (default: 30 min). To apply these changes immediately, you need to restart your application pool.

---

1. On the CRM.bulkloader main page select **Data Model, CRM Processes, Roles and Catalogs**.
2. Click **Next**.

### 3. Enter your Aurea CRM login data:

- **User name:** Enter your Aurea CRM login name.
- **Password:** Enter your password.
- **Single Sign-In:** If checked, you are logged on with the credentials of your current Windows login. (Neither **User name** nor **Password** have to be entered.)

---

**Note:** For CRM.bulkloader to work with single sign-on, change the configuration file `..\crm.services\web.config` as follows: Change the value of the `clientCredentialType` attribute from "None" to "Windows".

---

- **Language:** The language for the info area and field names of the data model. If you select "Users default language", the language entered in the user's record is used. (If undefined, "ger" is used as fallback.)

The data model is transferred to CRM.Designer in the selected language. This setting does not apply to catalogs: Catalog values are transferred in the catalog base language.

---

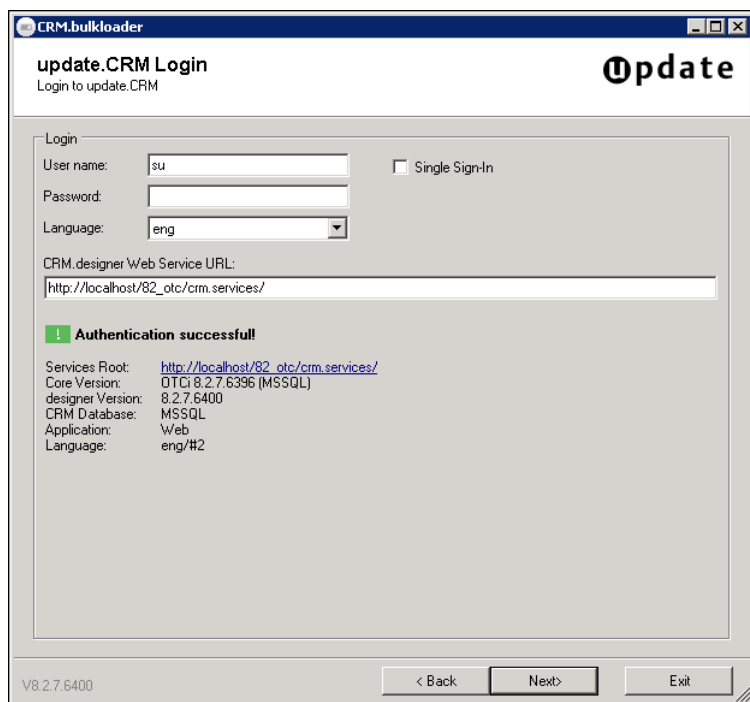
**Note:** If a maintenance user is defined, the login is performed with this user's credentials (and language). For further details, see [Maintenance User](#) in the *Aurea CRM.Web Installation Guide*.

---

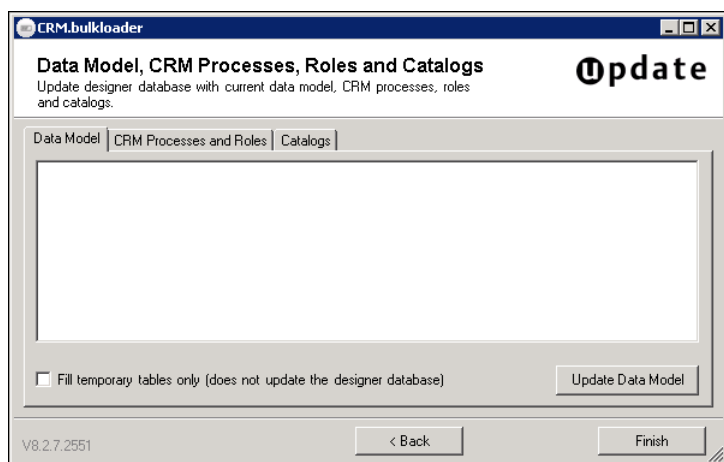
**Note:** update highly recommends defining a dedicated maintenance user and enter a language in this user's rep record to avoid updating the CRM.Designer database in the wrong language.

- **CRM.Designer Web Service URL:** Enter the path to the Aurea CRM.Web web service. The web service is located at `<web url>/crm.services/`.

4. Click **Login**.



5. After successful authentication click **Next**.



6. Select the desired function:

- [Updating the Data Model](#) on page 775
- [Updating CRM Processes and Roles](#) on page 777
- [Updating Catalogs](#) on page 778

7. Click **Finish** to return to the CRM.bulkloader main page.

---

**Note:** Since CRM.Designer caches the data model information, update recommends that you perform an application reset (IIS reset or application pool recycling) after updating the data model.

---

## Updating the Data Model

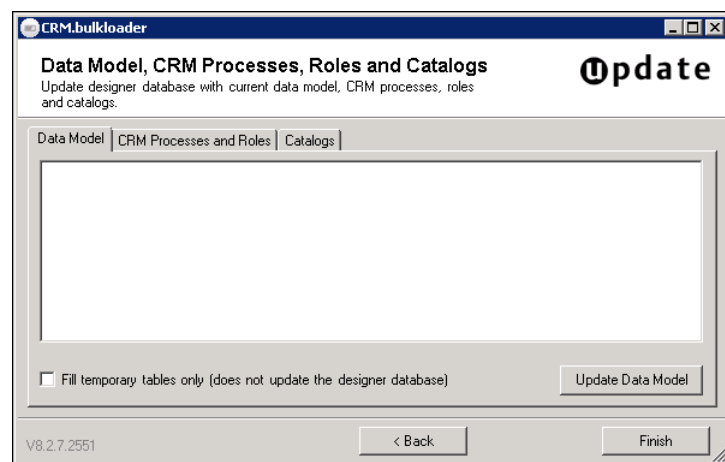
Learn to update the Data Model.

In order to be able to configure Aurea CRM.Web for the current Aurea CRM data model, you need to transfer it to CRM.Designer.

---

**Note:** Before updating the data model, update recommends recycling the application pool of Aurea CRM.Web (or iisreset).

---



1. On the **Data Model** tab click **Update Data Model**.

---

**Note:** If you want to fill the temporary tables without updating the data model, enable **Fill temporary tables only** before clicking **Update Data Model**. The option is only necessary if you got a timeout error while trying to update the data model. For further details, see [Manually Upgrading the CRM.Designer Data Model](#).

---

The current Aurea CRM data model is transferred to CRM.Designer:

- New tables and fields are added with the next unused table ID / field ID.
- Deleted tables and fields that are not referenced within the entire CRM.Designer database are deleted from the database.
- When deleting fields that are part of a field control definition, these fields are not deleted from the control and therefore from CRM.Designer.
- References to tables or fields that do not exist in the imported data model references a deleted field after upgrading.
- If a table or field which is currently marked as deleted in the CRM.Designer database is contained in the newly imported data model, then this deletion mark is cleared. This means that no new field is created, but instead, references to that field become valid again. Since more than one deleted table or field point to the same "logic field" (i.e. same info area, same unique field ID), the entry with the lowest table ID/field ID may be always reactivated.

### Manually Upgrading the CRM.Designer Data Model

If upgrading the CRM.Designer data model via CRM.bulkloader maintenance fails due to a timeout, you can re-start the upgrade with the **Fill temporary tables only** option enabled and then manually start the `PORTAL_UPGRADE_MMINFO` stored procedure, with the vertical ID as parameter (1 = BtB; 2 = F $\bar{S}$ ; 3 = Ot $\bar{C}$ ).

The upgrade applies the following actions to the database:

1. Activate tables marked as "deleted" if they are contained in the imported data model.

---

**Note:** A deleted table is only activated if there is no active table with the same info area ID, and if there is no table with the same info area that has a lower table ID.

---

2. Clear info area for duplicate deleted tables (set info area ID to blank).
3. Mark tables that are not included in the destination data model as "deleted".
4. Insert new tables.
5. Activate fields marked as "deleted" if they are contained in the imported data model.
6. Clear unique field ID of duplicate deleted fields.
7. Mark fields that are not included in the destination data model as "deleted".

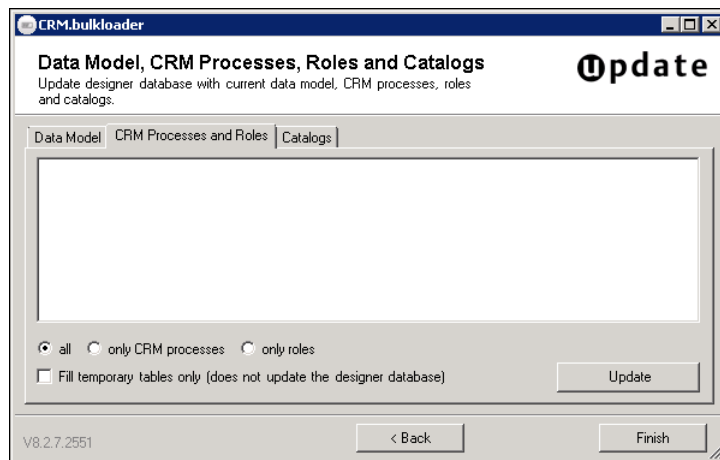


8. Insert new fields (if a new field has a unique field ID that is not already used as field ID, the new field is inserted with unique field ID = field ID).
9. Modify catalog tables according to the imported data model. Catalogs that no longer exist (i.e. catalogs that are not referenced at all or only referenced from fields that are marked "deleted") - are deleted and all references are removed.
10. All fields marked as "deleted" that are not referenced in the entire CRM.Designer database are deleted
11. All tables marked as "deleted" that are not referenced in the entire CRM.Designer database (and do not contain any referenced "deleted" fields) are deleted.

## Updating CRM Processes and Roles

Learn to update the CRM Processes and Roles.

To assign configuration units to CRM processes and roles, you need to transfer the currently defined CRM processes and roles to CRM.Designer.



1. Select what you want to update:
  - **all:** Both CRM processes and roles are updated.
  - **only processes:** Only CRM processes are updated.
  - **only roles:** Only roles are updated.

---

**Note:** Check **Fill temporary tables only** to only fill the temporary tables without updating the CRM processes and roles in CRM.Designer. This allows you to verify the changes before actually modifying existing data.

---

2. Click **Update**.

The currently defined CRM processes and roles are transferred to CRM.Designer. For further details, see [Roles & CRM Processes in CRM.Designer](#) on page 734.

CRM processes and roles that have been deleted in Aurea CRM are handled as follows in CRM.Designer:

- Deleted processes are moved to the **designer CRM Processes** node.
- Deleted roles are moved to the **designer Roles** node.
- Deleted groups are deleted for good.

For more information, see [Overview of Assigned Roles & CRM Processes](#) on page 736.

## Updating Catalogs

Learn to update the Catalogs.

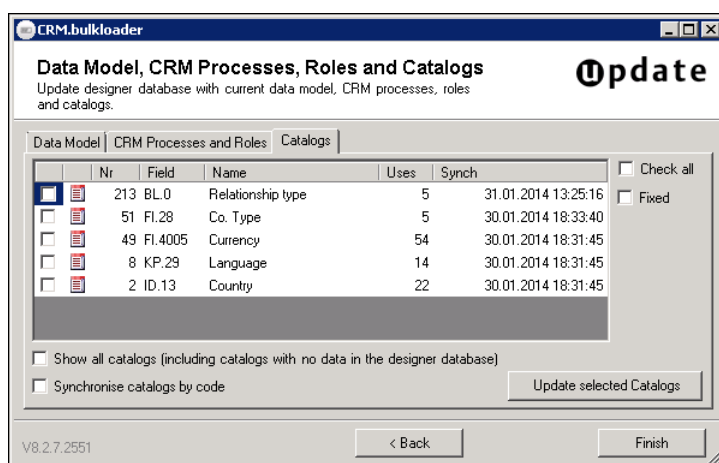
To use the catalog values (instead of numeric catalog codes or IDs) as input support in CRM.Designer, you need to transfer the catalog values to CRM.Designer. Catalog values are transferred to CRM.Designer in the catalog base language.

When using variable catalog values in CRM.Designer e.g. in filters, default values etc., you **have to synchronize** these catalogs because the codes of catalog values are not unique across stations. For further details, see [Catalogs](#) in the *CRM.Core Administrator Guide*.

Since fixed catalogs have unique IDs, synchronizing fixed catalogs is optional and only necessary if you want a more comfortable way to enter a fixed catalog value in CRM.Designer, e.g. when defining filters or queries. For further details, see [Defining Filters](#) on page 489.

To update catalog information in CRM.Designer:

1. Update the data model, see [Updating the Data Model](#) on page 775.
2. Update the catalogs:



Only catalogs that are actually used by a field in the data model are listed and therefore updated.

**Uses:** Number of fields that refer to the catalog.

---

**Synch:** Last time the catalog was transferred to CRM.Designer.

3. Check **Show all catalogs** e.g. to display newly created catalogs that have not yet been transferred to CRM.Designer. Check **fixed** to display and select fixed catalogs as well.
4. Select the catalogs you want to update by activating their check boxes.

---

**Note:** You should only synchronize catalogs whose values you actually use in CRM.Designer (or in your add-on programming).

---

---

**Note:** Synchronizing all catalogs (**Check all** option) is not necessary and not recommended as it just prolongs loading times.

---

5. The option **Synchronize catalogs by code** allows text changes made by the Aurea CRM catalog maintenance to be reflected in the CRM.Designer. If this option is not checked, catalog entries that do not exist in the CRM.Designer (= same default text and same tenant number) are created and entries that exist but have a different code are corrected. If this option is checked, entries that do not exist in the CRM.Designer (= same catalog code) are created and entries that have the same code, but different text and/or tenant number, are changed.

---

**Note:** If you are using external keys for variable catalogs (recommended procedure), the **Synchronize catalogs by code** option has no effect as the catalog values are always synchronized by external key.

---

6. Click **Update selected catalogs**.

The values of the selected catalogs are transferred to CRM.Designer.

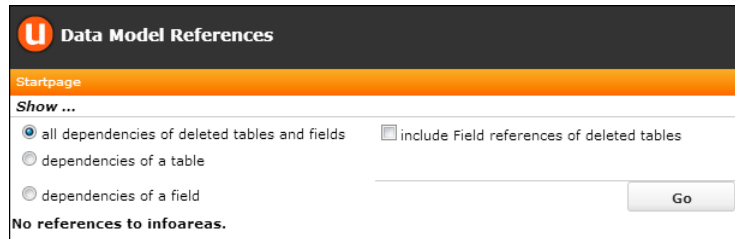
Deleted catalogs and catalog values are not deleted from the CRM.Designer database. You need to delete them manually, see [CRM.Designer Catalog Maintenance](#) on page 722.

## Data Model References

Learn about Data Model References.

When a configuration is exported with CRM.bulkloader, the entire data model is also written to the CRM.bulkloader XML files – this means that tables and fields that are marked as deleted but referenced within the exported configuration are exported as well. Tables and fields that are marked as deleted and not referenced within the exported configuration are not exported.

You might face problems when importing configurations to another CRM.Designer database if your configuration has multiple references to different deleted tables that have the same info area ID, or fields that have the same unique field ID. To avoid this problem, check the Data Model References section of the CRM.Designer to show all references to tables or fields that are marked as "deleted".



Although it is not necessary to completely remove references to deleted tables and fields for the export / import to work properly, it is **highly recommended** to clear your configurations from such references – they may cause unpredictable behavior of the configured application.

When a configuration is imported the following steps take place to adjust the data models of the designer database and the imported configuration:

1. (only for root configurations) Clear the "deleted" flag from tables that are contained in the imported root configuration but are marked as deleted in the CRM.Designer database
2. Add tables that are only contained in the imported configuration. If a root configuration is imported, the tables are imported (and the "deleted" flag is set depending on the flag's state in the imported configuration), for sub configurations they are always imported marked as deleted.
3. (only for root configurations) Clear the "deleted" flag for fields that are contained in the imported root configuration but are marked as deleted in the CRM.Designer database
4. Add fields that are only contained in the imported configuration (as with tables, either marked "deleted" or not for root configurations, and always flagged "deleted" for sub configurations). If possible, the fields are added with unique field ID = field ID.
5. (only for root configurations) Field Information (name, type, catalog number, ...) is updated for existing fields
6. All table and field references in the configuration to be imported are changed so they match the data model in the destination designer.

---

## Aurea CRM Data Model Upgrade Implications on Synchronized Catalogs

Learn about the data model upgrade implications on Synchronized Catalogs.

CRM.Designer's catalog synchronization feature is affected by a data model upgrade if a catalog was completely deleted from the data model (i.e. if no fields are referencing the catalog anymore). Synchronization feature to store catalogs and values in the CRM.Designer database to provide a safe reference to catalog values. For further details, see [Catalog Value Synchronization](#).

In this case, a catalog is deleted from table `PORTAL_CATALOG`, and all values of this catalog are also removed from `PORTAL_CATALOG_VALUES`. In addition, all references of catalog values are removed from the database.

---

**Note:** This has the side effect that instead of the synchronized catalog values, the text values is used in queries and filter of Aurea CRM.Web, leading to unpredictable and wrong results! Thus when deleting a catalog, you should always make sure to manually correct those filters and queries that made use of it!

---

The deletion of all catalog values for catalogs that are not referenced anymore means that all references to values of this catalog in the CRM.Designer are permanently destroyed.

Synchronized catalog values are offered for the following features within CRM.Designer:

- Filter values in Queries and Filters (table `PORTAL_QUERY_FILTERVALUES`)
- Web Configuration Values based on catalogs (table `PORTAL_WEBCONFIG_VALUE`)
- Analysis Categories (table `PORTAL_STAT_CATEGORY_CRIT`)
- Alternate Expand Configurations (table `PORTAL_TABLECONFIG_COND`)

The catalog synchronization feature writes all values of a catalog to the CRM.Designer database (if they are not already there). After synchronizing a catalog, the CRM.Designer contains

1. all values that are currently stored in the catalog, and
2. all values that are not stored in the catalog anymore but where known to the CRM.Designer prior to the synchronization.

Please note that catalog values which were modified with the Aurea CRM catalog maintenance (in Aurea CRM win **Maintenance** module) is inserted as new values into the CRM.Designer database, and the old values is not removed!

Catalogs in the CRM.Designer database are maintained on the **Catalogs** page of CRM.Designer. For further details, see [CRM.Designer Catalog Maintenance](#) on page 722.

## CRM.bulkloader Command Line Parameters

Learn about CRM.bulkloader Command Line Parameters.

<bulkloaders>'s functions can be accessed via the command line.

Use the following syntax:

```
update.bulkloader <Operation> <Command> [options]
```

Below operations can be performed:

### **Download: Downloads designer configurations from your CRM.Designer database**

**Syntax:** `update.BulkLoader Download <Command> <-vertical> <-configname> <-languages> <-packageid> <-packagename> <Connection> [Options]`

Parameters for the `Download` operation:

Name	Description
<Command>	All Config Package DataModel
All	Downloads a complete database dump, i.e. the contents of a whole CRM.Designer database.
Config	<p>Downloads a specific CRM.Designer configuration.</p> <p>Use the following options with <code>Config</code>:</p> <p><code>-vertical:&lt;VerticalName&gt;</code> ...the name of the vertical (Base BB BC FS)</p> <hr/> <p><b>Note:</b> <code>-vertical</code> overrides the vertical specified in the <code>settings.xml</code> file (<code>-configFile</code>).</p> <hr/> <p><code>-configName:&lt;ConfigName&gt;</code> ...the name of the configuration (e.g. <code>UserConfig_XY_Sub0</code>)</p> <p><code>-languages:"&lt;Languages&gt;"</code> ...comma-separated list of languages to download (e.g. <code>"English,German"</code>). If undefined, all available languages are exported.</p> <hr/> <p><b>Note:</b> Hidden languages (<b>Visible</b> unchecked on the <b>Language Settings</b> page) are not exported.</p> <hr/> <p><b>Note:</b> Download at least two languages, even if you only need one. The additional language does not have significant impact on the size of the XML files and the software. If the XML files contain only one language, upon upload CRM.bulkloader assumes that you are uploading a language pack.</p> <hr/>

Name	Description
Package	<p>Downloads a specific package from a given configuration. Use the following options with <code>Package</code>:</p> <pre>-vertical:&lt;VerticalName&gt; -configName:&lt;ConfigName&gt; ...the name of the configuration where the package is defined -languages:"&lt;Languages&gt;" -packageid:&lt;packageId&gt; ...the numeric package ID -packagename:&lt;PackageName&gt; ...the name of the package</pre> <hr/> <p><b>Note:</b> You need to specify either <code>-packageid</code> or <code>-packagename</code>. If both are defined, <code>-packageid</code> is used.</p>
DataModel	<p>Downloads the Aurea CRM data model as currently present in the CRM.Designer database. Use the <code>-vertical:&lt;VerticalName&gt;</code> option with <code>DataModel</code>.</p>
<Connection>	Specify the connection to the CRM.Designer database.
-datasource:"<connection string>"	<p>The connection string to the CRM.Designer database, e.g. <code>-datasource:"Data Source=&lt;DBserver&gt;;Initial Catalog=&lt;DB&gt;;UID=&lt;loginName&gt;; PWD=&lt;plaintextpassword&gt;"</code></p> <p>To use single sign-on you need to specify the DSN: Example: DSN for SQL driver with single sign-on enabled:</p> <pre>-datasource:"Provider=SQLOLEDB;Data Source=mysqlserver;InitialCatalog=mydatabase;IntegratedSecurity=SSPI;"</pre> <p>For information on how to enable single sign-on, refer to the documentation of your database driver.</p>
-configFile:"<path>"	<p>The <code>settings.xml</code> file containing the connection string (<code>designerDsn</code>) and default download folder (<code>&lt;XMLFilePath&gt;</code>), e.g. <code>-configFile:" c:\updateCRM\bb\web\system\settings\settings.xml "</code></p> <hr/> <p><b>Note:</b> You need to specify either <code>-datasource</code> or <code>-configFile</code>. If both are defined, <code>-configFile</code> is used.</p>
[Options]	



Name	Description
<code>-verbose</code>	Displays full progress information on screen.
<code>-xml- FilePath:"&lt;path&gt;"</code>	<p>Path where the XML files are downloaded to, e.g. <code>-xml-FilePath:"c:\temp\xml"</code></p> <hr/> <p><b>Note:</b> You must specify an existing path. CRM.bulkloader does not create new folders.</p> <hr/> <p><b>Note:</b> <code>-xmlFilePath</code> overrides the <code>&lt;XMLFilePath&gt;</code> setting stored in the <code>settings.xml</code> file.</p> <hr/>
<code>-logfile:"&lt;path&gt;"</code>	Path and name of the log file. If undefined, no log is written.
<code>-y</code>	Automatically answers all prompts during download with "Yes".
<code>-dbTimeout:&lt;s&gt;</code>	Database timeout in seconds (default = 300 seconds)

### Examples:

```
update.bulkLoader.exe Download All -configFile:" c:\update-CRM\bb\web\system\settings\settings.xml " -xmlFilePath:" c:\temp\xml\dump "
```

Dumps the complete database (connection specified by `-configFile`) to the specified folder (`-xmlFilePath`).

```
update.bulkLoader.exe Download Config -configName:CustomizedCore_Config -vertical:BB -languages:"English,German" -configFile:" c:\update-CRM\bb\web\system\settings\settings.xml " -verbose
```

Downloads the configuration `CustomizedCore_Config` for the BTB vertical and the languages English and German to the default XML folder. The database connection and default XML folder are read from `c:\updateCRM\bb\web\system\settings\settings.xml`.

```
update.bulkLoader.exe Download Package -vertical:BB -configName:UserConfig_XY_Sub0-packageName:MyPackage-configFile:"c:\updateCRM\bb\web\system\settings\settings.xml " -xmlFilePath:" c:\temp\xml "
```

Downloads the package "MyPackage" from the specified vertical and configuration to the specified folder.

### Upload: Uploads a designer configuration into your CRM.Designer database

**Syntax:** `update.bulkloader Upload <Command> <Connection> [Options]`

Parameters for the `Upload` operation:

Name	Description
<Command>	Temporary   Final   Package   DataModel   Dump   CreateDB
Temporary	The configuration is only uploaded to temporary tables (TMP_PORTAL*).
Final	The configuration is uploaded directly into the PORTAL* tables.
Package	<p>Uploads a CRM.Designer package into a configuration. Use the following options with Package:</p> <p>-ConfigName:&lt;ConfigName&gt; ...the name of the configuration to which the package is uploaded</p> <p>-OnlyDefinition ...if specified, only the package definition is uploaded.</p>
DataModel	Uploads the Aurea CRM data model into the CRM.Designer database.
Dump	Creates the database structure and uploads a database dump (only works with an empty database).
CreateDB	Creates the CRM.Designer database structure.
<Connection>	Specify the connection to the target database.
-datasource:"<connection string>"	<p>The connection string to the CRM.Designer database, e.g. -datasource:"Data Source=&lt;DBserver&gt;;Initial Catalog=&lt;DB&gt;;UID=&lt;user name&gt;; PWD=&lt;plaintextpassword&gt;"</p>
-configFile:"<path>"	<p>The settings.xml file containing the connection string (designerDsn) and default upload folder (&lt;XMLFilePath&gt;), e.g. -configFile:"c:\updateCRM\bb\web\system\settings\settings.xml"</p> <hr/> <p><b>Note:</b> You need to specify either -datasource or -configFile. If both are defined, -configFile is used.</p> <hr/>
[Options]	
-verbose	Displays full progress information on screen.

Name	Description
-xml- FilePath:"<path>"	Path to the configuration files to upload, e.g. -xml- FilePath:"c:\updateCRM\bb\designer_Configuration\SubCon- figuration\MyConfig" <hr/> <b>Note:</b> -xmlFilePath overrides the <XMLFilePath> setting stored in the settings.xml file.
-logFile:"<path>"	Path and name of the log file. If undefined, no log is written.
-y	Automatically answers all prompts during upload with "Yes".
-dbTimeout:<s>	Database timeout in seconds (default = 300 seconds)
-c	Creates a non-existing configuration on upload, if possible. A configuration can only be created if its parent configuration already exists in the target database.

## Examples

```
update.bulkloader.exe Upload Final -configFile:" c:\update-  
CRM\bb\web\system\settings\settings.xml " -verbose -y
```

Uploads a configuration using the `XmlFilePath` and database credentials defined in the `settings.xml` specified by `-configFile`.

```
update.bulkloader.exe Upload Dump -configFile:" c:\update-  
CRM\bb\web\system\settings\settings.xml " -xmlFilePath:"c:\update-  
CRM\designer\myDump "
```

Uploads a database dump located at the path specified by `-xmlFilePath` to the database specified by `-configFile`.

---

**Note:** It is strongly recommended to back up the CRM.Designer database before uploading configuration data.

---

**Datamodel:** Updates the CRM.Designer database with the current data model, CRM processes, roles and catalogs from the Aurea CRM database

**Syntax:** `update.bulkloader DataModel <Command> <Connection> [UpdateOptions] [CatalogOptions] [ProcessOptions] [Options]`

Parameters for the `Datamodel` operation:

Name	Description
<Command>	Update   Catalog   Process
Update	Updates the CRM.Designer database with the current Aurea CRM data model.
Catalog	Updates the CRM.Designer database with the catalogs currently defined in Aurea CRM.
Process	Updates the CRM.Designer database with the CRM processes and roles currently defined in Aurea CRM.
<Connection>	
-serviceUri:<Uri>	The path to the designer web service, e.g. <code>http://localhost/update/crm.services</code>
-user:<name>	The Aurea CRM user name used to log on to the designer web service.
-password: <plaintextpassword>	The password used for login to the designer web service.
-language:<language>	The language used for login to the designer web service.(e.g. <code>eng</code> or <code>ger</code> ). If undefined, <code>eng</code> is used as default. For the three-character language codes, >> <b>Language IDs</b> in the <i>CRM.Core Administrator Guide</i> .
[UpdateOptions]	Use with the <code>Update</code> command.
-temponly	Fills only the temporary tables.
[CatalogOptions]	Use with the <code>Catalog</code> command.
-variable:<list>	Comma-separated list of variable catalogs to synchronize. Enter <code>*</code> for all.
-fixed:<list>	Comma-separated list of fixed catalogs to synchronize. Enter <code>*</code> for all.
-bycode	Synchronizes catalogs by code.

Name	Description
[ProcessOptions]	Use with the <code>Process</code> command. If undefined, both CRM processes and roles are updated.
-temponly	Fills only the temporary tables.
-processes	Updates CRM processes only.
-roles	Updates roles only.
[Options]	
-single	Enables single sign-on. <code>-user</code> and <code>-password</code> are ignored.
-logFile:"<path>"	Path and name of the log file. If undefined, no log is written.
-y	Automatically answers all prompts with "Yes".

### Maintenance: Creates or upgrades the CRM.Designer database

**Syntax:** `update.bulkloader Maintenance <Command> <Connection> [Options]`

Parameters for the `Maintenance` operation:

Name	Description
<Command>	CreateDb   UpgradeDb   StoredProcedures
CreateDB	Creates the CRM.Designer database structure.
UpgradeDB	Upgrades the CRM.Designer database structure to the current version.
StoredProcedures	Re-creates the stored procedures.
UpgradeTmp	Re-creates the TMP tables.
<Connection>	Specify the connection to the CRM.Designer database.
-datasource: "<connection string>"	The connection string to the CRM.Designer database, e.g. -datasource:"Data Source=<DBserver>;Initial Catalog=<DB>;UID=<loginName>; PWD=<plaintextpassword>"
-config-File:"<path>"	The settings.xml file containing the connection string (designerDsn), e.g. -configFile:"c:\updateCRM\bb\web\system\settings\settings.xml"  You need to specify either -datasource or -configFile. If both are defined, -configFile is used.
[Options]	
-verbose	Displays full progress information on screen.
-log-File:"<path>"	Path and name of the log file. If undefined, no log is written.
-y	Automatically answers all prompts with "Yes".
-dbTimeout:<s>	Database timeout in seconds (default = 300 seconds)

**Note:** It is strongly recommended to back up the CRM.Designer database before upgrading the database structure.

## Bulkloader Database configuration settings

Learn about the settings to be performed in the bulkloader configuration file.

Do the following settings in the `applicationSettings` node of the `update.BulkLoader.exe.config` file:

`DatabaseConnectionProvider`: used to change the database connection provider as the default provider used is `SQLOLEDB`.

`ConnectionStringParameters`: used to pass extra connection string parameters to connect to the database based on the supported parameters for the connection provider.

```
update.BulkLoader.exe.config x
146     </setting>
147   </update.BulkLoader.Properties.Settings>
148 </userSettings>
149 <applicationSettings>
150   <update.BulkLoader.Properties.Settings>
151     <setting name="DBVersion" serializeAs="String">
152       <value>7.50</value>
153     </setting>
154     <setting name="createdesigner_script" serializeAs="String">
155       <value />
156     </setting>
157     <setting name="upgradedesigner_script" serializeAs="String">
158       <value />
159     </setting>
160     <setting name="upgradetmptables_script" serializeAs="String">
161       <value />
162     </setting>
163     <setting name="upgradeobjects_script" serializeAs="String">
164       <value />
165     </setting>
166     <setting name="DatabaseVersion" serializeAs="String">
167       <value>9.02</value>
168     </setting>
169     <setting name="DatabaseConnectionProvider" serializeAs="String">
170       <value>SQLNCLI11;;</value>
171     </setting>
172     <setting name="ConnectionStringParameters" serializeAs="String">
173       <value>DataTypeCompatibility=80</value>
174     </setting>
175   </update.BulkLoader.Properties.Settings>
176 </applicationSettings>
177 <startup>
178   <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.5" />
179 </startup>
180 </configuration>
```



## CRM.launcher(Enterprise)

Learn about the standalone CRM.launcher feature available on Enterprise edition.

CRM.launcher is a stand-alone Windows executable (communicating with Aurea CRM.Web via secure web sockets) that allows performing actions that require communication with applications "outside the browser sandbox", i.e. MS Office and CRM.phone.

Once installed on the client, CRM.launcher acts as a host for plug-ins providing the following functions:

- Document check-in/-out, editing, mail merge and quick campaigns
- Using CRM.phone with Aurea CRM.Web
- Aurea CRM.Web offline

The plug-ins are loaded as defined in the `update.launcher.exe.config` file:

```
<PlugIns>
  <!--<PlugIn type="update.Launcher.Offline.PlugIn,update.Launcher.Offline"
    required="true" />-->
  <PlugIn type="update.Launcher.WebBrowser.PlugIn,update.Launcher.WebBrows-
    er" required="false" />
    <PlugIn type="update.Launcher.RemoteControl.PlugIn,update.Launcher.Re-
    moteControl" required="false" />
  <PlugIn type="update.Launcher.Documents.PlugIn,update.Launcher.Documents"
    required="false" />
  <PlugIn type="update.Launcher.Phone.PlugIn,update.Launcher.Phone" re-
    quired="false" />
</PlugIns>
```

CRM.Launcher can connect to CRM Server via non-transparent firewalls provided by third parties. It does this by authenticating once using the proprietary mechanism of the firewall provider and then saving the authentication cookies for subsequent connections. For more information, see

The saved authentication cookies are also used by CRM Web Offline to connect to the CRM server through the same firewalls. For information on CRM.launcher in connection with Aurea CRM.Web offline, see [Aurea CRM.Web offline Installation Guide](#).

## Installing CRM.launcher

Learn to install the CRM.launcher.

You can install CRM.launcher for the following environments:

- Single user environment
- Shared mode environment

The shared mode installation is done for the following scenarios:

- For Citrix and Terminal servers the launcher is installed on the corresponding server that in turn allows remote users to connect and use the services offered by the CRM.launcher. For more information see, [Installing CRM.launcher in the Citrix environment](#) on page 802.
- In other shared environments, the CRM.launcher can be installed on the CRM.Web server or any other server machine. When a user (client machine) connects to the web server and tries to access the launcher services, a launcher instance is installed on the client machine (single user environment) ensuring that the launcher services are available to the user. The user should be part of the domain that is set up by the domain administrator when installing the launcher on the shared mode environment.

The single user installation is for users who are part of a domain or a standard CRM server installation that allows a single user. CRM.launcher is also available as MSI package that can be deployed on systems running in a networked environment using SCCM (System Center Configuration Manager).

## Single user environment

Installing CRM.launcher on a Single user environment/personal environment.

Below is the setup for users who do not have a shared environment:

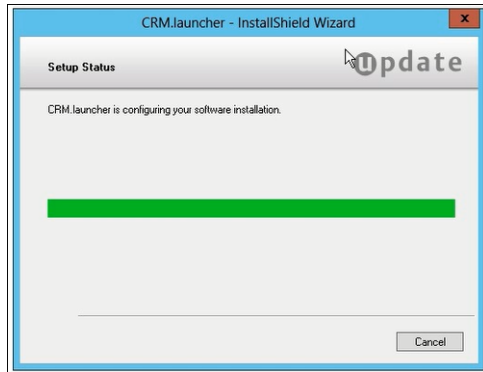
To install the CRM.launcher for a single user, perform the following steps:

1. Double click the `CRM_launcher.exe` installer file. The CRM.launcher InstallShield Wizard displays.

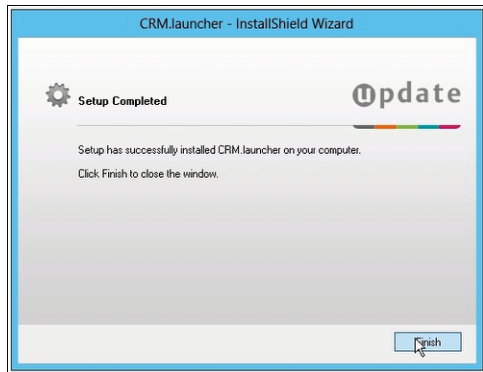


Notice that you can install the CRM.launcher for all users in a shared environment, if you are the administrator of the system. You can also install it for yourself (Single User) if you are not the administrator of the system.

2. Select **Single User** option and click **Next**.



3. The InstallShield Wizard starts the installation without requiring any further inputs.

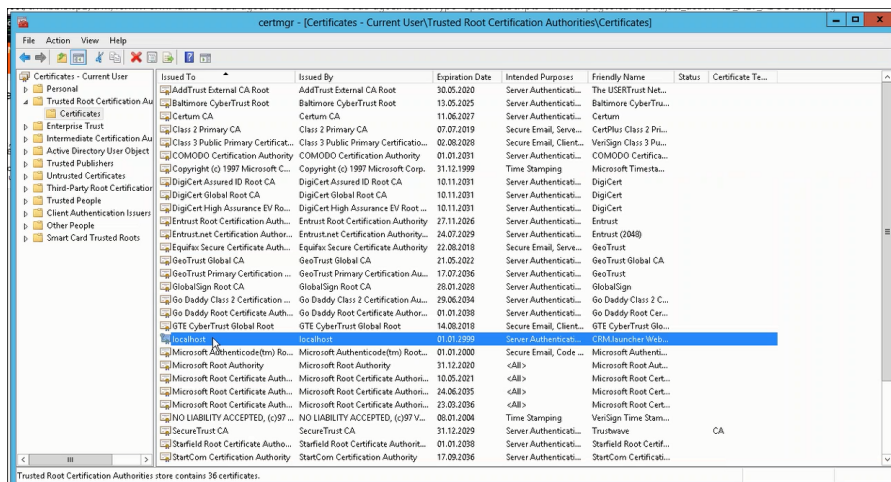


4. After installation is complete, the Setup Completed page displays.

5. Click Finish to complete the installation.

**Note:** The CRM.launcher is installed in the `AppData\local\update.CRM\CRM.launcher` folder. The CRM.launcher is automatically started when the user is logs into the system.

6. The CRM.launcher also installs a certificate which is used to authenticate with the CRM.Web server.



**Note:** The default certificate type X.509 and is issued by Aurea. You can also use external certificates issued by other certification providers. If you want to use third party certificates then set the path to the certificates in the configuration files. For more information on configuration files, see [Configuring the CRM.launcher](#).

---

## Deploying CRM.Launcher using SCCM

CRM.launcher is available as an MSI package for installation in a single user mode. This allows organizations to safely deploy CRM.launcher on user machines using Microsoft SCCM (System Center Configuration Manager). This method provides a controlled centralized method of installing, uninstalling, and upgrading CRM.launcher on target systems. For more information, see [Deploying Applications Using SCCM](#).

## Shared Environment

In a shared environment, the CRM.launcher can be installed for all the users who use the system.

The application is installed in the `Program Files` folder. All communication with the Aurea.CRM.Web is handled through a named pipe server.

---

**Note:** Using CRM.launcher for multiple user login sessions is an enterprise feature and is available only for enterprise editions of Aurea CRM installations. While you can install it for all users, if you do not have an enterprise license then only the admin can use the CRM.launcher.

---

To install the CRM.launcher for all users in a shared environment, perform the following steps:

1. Double click the `CRM.launcher.exe` installer file.

The InstallShield wizard displays.



Notice that you can install the CRM.launcher for all users in a shared environment, if you are the administrator of the system. You can also install it for yourself (Single User) if you are not the administrator of the system.

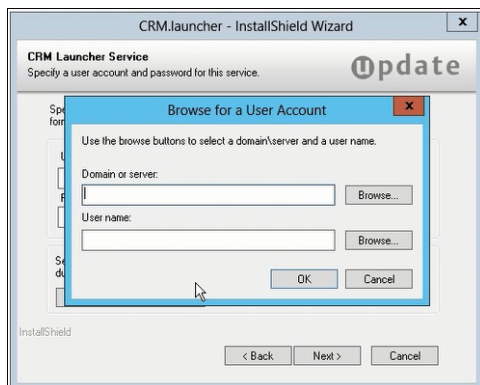
2. Select **All Users** and click **Next**, the **CRM Launcher Service** window displays.



The CRM.launcher application is installed as a windows service. The windows service acts as a proxy for all CRM.launcher sessions for all users.

**Note:** Only a user belonging to the administrator group can install the CRM.launcher for all users.

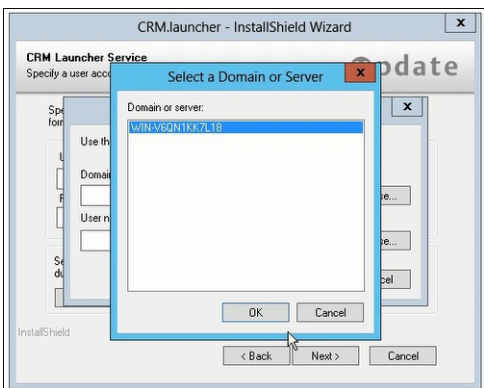
3. Enter the **User name** in the format `<Windows domain name>\<username>`. Alternatively, click **Browse**, the **Browse for a User Account** window displays.



You can enter the administrator username and password for this system if you are a system administrator user.

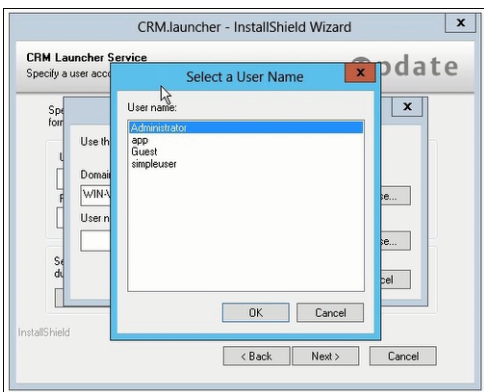
If you are a domain administrator group user, then perform the following steps to grant access to all the domain users. The launcher is installed as a system service.

- In the **Browse for a User Account** window, click the **Browse** button for the **Domain or Server** text box. And select the required domain or server from the **Select Domain or Server** window. Click **OK**.



**Note:** The domain user can be a member of the active directory services.

- Click **Browse** button for the **User name** text box, and select the administrator user from the **Select a User name** window. Click **OK**.

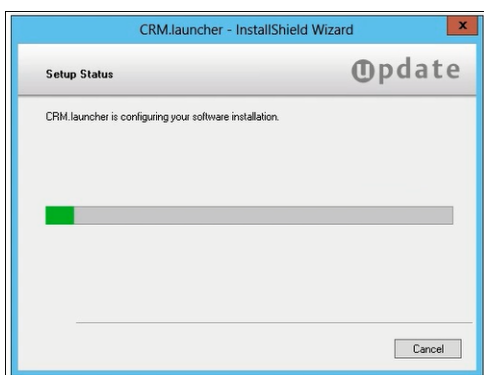


- In the resulting CRM Launcher Service window, enter the Password for the selected user.

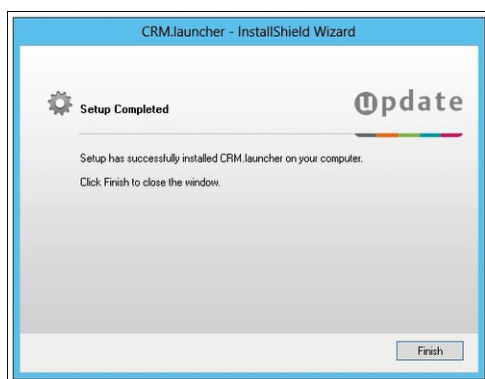


**Note:** You can also add new users to the windows system who can access the CRM.launcher application. Use the **New User Information** button.

- Click **Next**. The Setup Status window displays.

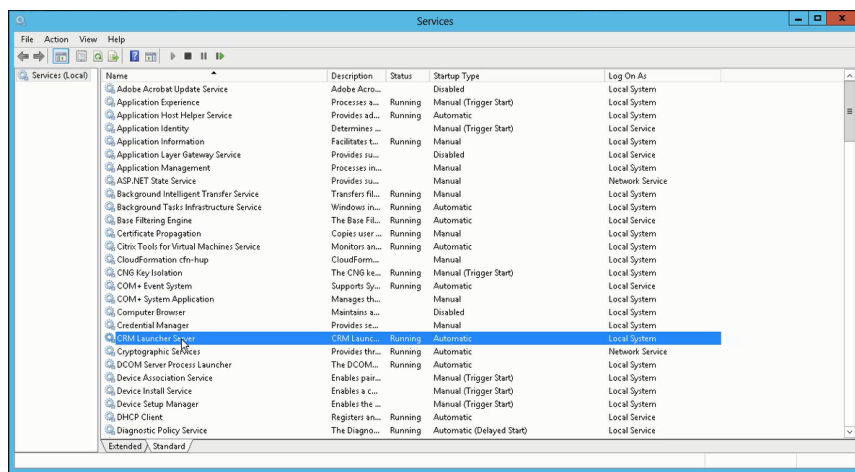


#### 4. Click Finish to complete the installation.

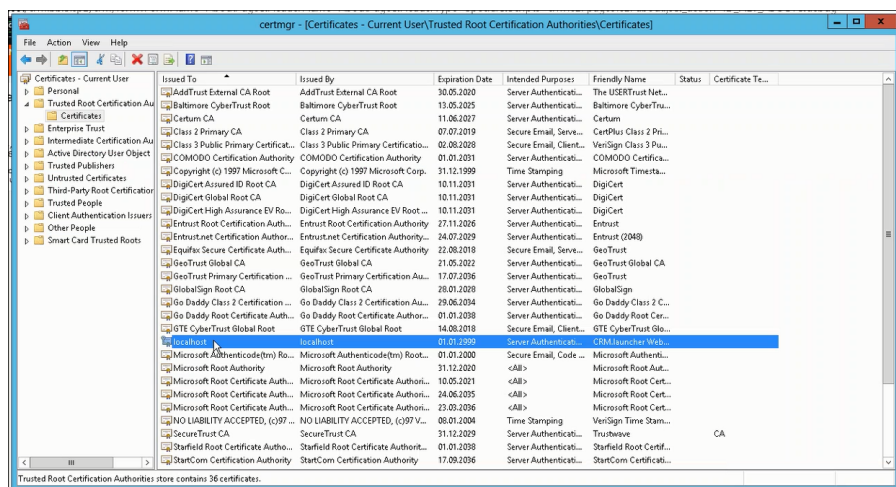


**Note:** It is started when the user logs into an Aurea.CRM.Web application and tries to start a plugin that requires CRM.launcher. The user is then asked to activate the CRM.launcher application on his system by clicking the link provided in a message window of the web browser.

#### 5. The CRM.launcher server is started immediately on installation as a windows service. The startup type for the service is set to automatic.



6. When a domain user initially connects to the CRM.Web server a certificate is installed on his system which is used for authentication with the CRM.Web server.
7. The CRM.launcher also installs a certificate which is used to authenticate with the CRM.Web server.



**Note:** The default certificate type X.509 and is issued by Aurea. You can also use external certificates issued by other certification providers. If you want to use third party certificates then set the path to the certificates in the configuration files. For more information on configuration files, see Configuring the CRM.launcher.

## Silent Mode Installation (commandline and batch files)

You can install CRM.launcher in silent mode by running the installer from a commandline shell or a batch file using the `--silent` option.

**Note:** If you run the installer from the commandline shell without the `-silent` option, then the installation wizard dialog is launched.

The installer also takes the following options:

- `--mode`—specify if the installation is for a single user, all users, or for a shared environment (for example, Citrix XenDesk or a Terminal Server).

**Note:** If this option is not specified, the default value SINGLE is assumed.

It takes the following values:

- ALL — installs CRM.launcher for all users configured on the system



---

**Note:** This mode assumes administrator rights for the installer.

---

- SINGLE — installs CRM.launcher for the current user

---

**Note:** This mode assumes the current user.

---

- SHARED — installs CRM.launcher as a windows service in a shared environment such as Citrix XenDesktop or Microsoft Terminal Services

---

**Note:** In this mode, CRM.launcher installer creates a Launcher Windows service that runs under a service-user account. And the username and password options are mandatory. Sometimes, in a shared environment like Citrix, credential for the launcher service-user is set up after the installation. For such cases, use the `--ignorecredential` option to avoid having to provide the username and password. You can also use the `--nostart` option to prevent the automatic restart of Luncher after installation.

---

- `--username`—specify the name of the user account used to execute the launcher installer. This account is used to run the launcher service.
- `--password`—specify the password for the user account used to execute the CRM.launcher installer.
- `--i`—specify the path to the installation folder.
- `--nostart`—use this option to prevent the Launcher from starting after installation. This is useful when a software distribution tool is used to install CRM.launcher.
- `--ignorecredential`—use this option only for SHARED mode installation. This installs CRM.Launcher as a service using the default local system account. The service-user and credentials can be setup later.

---

**Note:** After the silent installation, the `sc` command can be used to change the service-user details. See the following example:

- `sc stop "CRMLauncherServer"`
  - `sc config "CRMLauncherServer" obj= ".\CrmServiceUser" password= "crmser-vice- password"`
  - `sc start "CRMLauncherServer"`
-

### Some examples of Silent mode installation

To install CRM.launcher in silent mode, in a shared environment, use the following in a command shell:

```
CRM.launcher.exe --silent --mode=SHARED --username=Administrator --password=hG&4e%km  
--i=C: \Aurea\CRM\launcher
```

To install CRM.launcher in silent mode, in a shared environment, and service-user details to be provided after the installation, use the following in a command shell:

```
CRM.launcher.exe --silent --mode=SHARED --ignorecredential --i=C:\Aurea\CRM\launcher  
--nostart
```

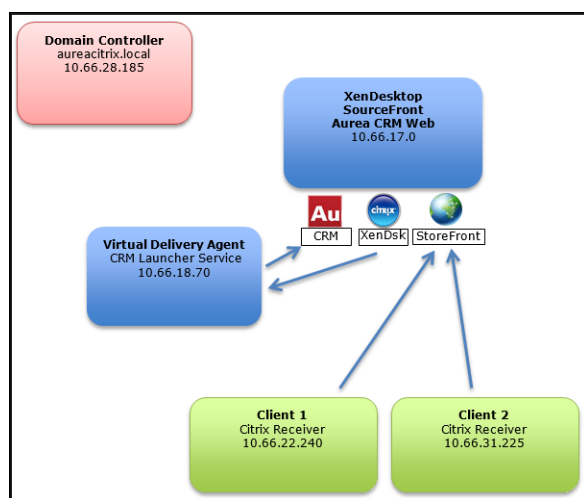
## Installing CRM.launcher in the Citrix environment

In the Citrix environment, the CRM.Launcher is installed only once in shared mode on the Virtual Delivery Agent.

The Virtual Delivery agent then presents the CRM.launcher instance to each of the different Citrix client users connecting to the Store front. When the user requests an action from CRM.web that requires the CRM.launcher, the launcher service running in the background is invoked.

**Note:** Sometimes a popup appears on the CRM.Web with a link to start the Launcher service. This only happens when the CRM.launcher service is not already running. The reasons could be if the launcher service is restarted and the communication between the CRM.Web and the CRM.launcher breaks down. Or the Citrix user is logging in for the first time to the Store front.

The following screenshot shows a typical Citrix environment:

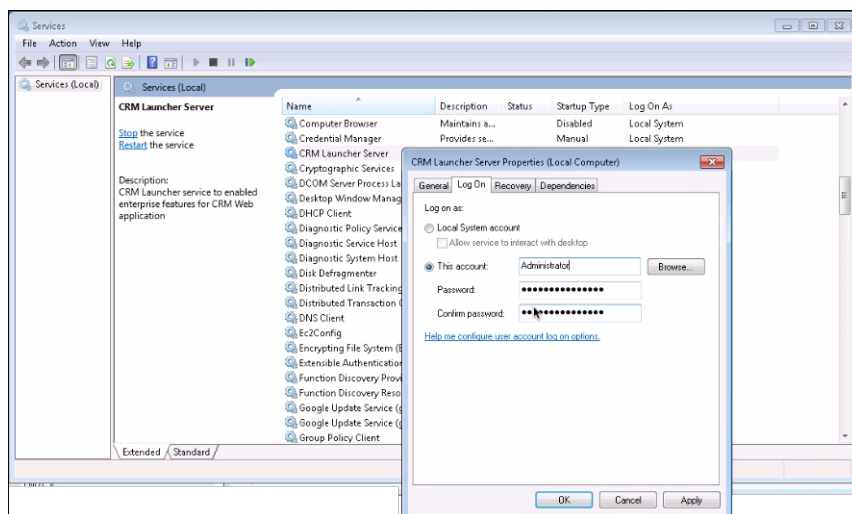


The following description outlines the roles of the different players in the Citrix environment interacting with CRM.Launcher:

- Domain Controller—this is the primary domain controller in the Citrix environment and every other machine is a member of this domain.
- Xen Desktop Server—Citrix XenDesktop configured with StoreFront to provide virtual desktops for clients. Both, Aurea CRM.Web and the SQL Server Database could be installed here or on separate servers.
- Virtual Delivery Agent—the base desktop that is presented to the user through Store front. CRM.launcher enterprise should be installed here. The CRM.launcher service is running using a domain account. Ensure that the X.509 certificate is installed for this account. The certificate is installed by the CRM.launcher client.
- Clients—these Citrix client users connect to StoreFront using the Citrix Receiver web plugin.

To install the CRM.launcher in a Citrix environment, perform the following steps:

1. Login to the Virtual Delivery Agent.
2. Run the CRM.launcher installer as an administrator.
3. Install the CRM.launcher in shared mode using a local account. For instructions on installing CRM.launcher in shared mode environments, see Shared Environment on page .
4. After installing CRM.launcher open the local services page and select the **CRM.launcher Server**, as shown in the screenshot below.



5. Double click the CRM.launcher Server and to open the properties window.
6. Now configure the CRM.launcher Server to start with a domain account. Enter the account name and Password.
7. Restart the CRM.launcher server.

8. The CRM.launcher service is now available to all client users in the configured Citrix domain.
9. After installing the CRM.launcher Server, run the CRM.launcher client. This ensures that the certificates required by the CRM.launcher service and client are installed on the virtual delivery agent. For more information on configuring the certificates, see Certificate Configuration.

## Upgrading the CRM.launcher

Learn how to upgrade the CRM.launcher.

If you already have a previous version of CRM.launcher installed, then running the new installer detects the previous installation and upgrade the previous installation automatically. There is no need to uninstall the previous version and re-install the new version.

In the case of weboffline system, when you run the CRM.launcher installer it detects the previous weboffline installation and shows a message box with the warning “In order to upgrade the Launcher you need to use a newer web offline installer.” In this case you can enforce, just the launcher upgrade by using silent mode command line installation, with the following command options:

- `MigrateLauncher` – forces the upgrade of the launcher with out upgrading weboffline.
- `Vertical=<VERTICAL NAME>` - Specify the appropriate edition of the CRM.Web vertical for which the CRM.launcher is being upgraded, from the following verticals: BTB, FS, OTC.

For example, to force upgrade the CRM.launcher for a BTB vertical edition, without upgrading the weboffline, use the following command: `CRM.webOffline_installer.exe -MigrateLauncher -Vertical=BTB .`

If you are upgrading the weboffline installer, the weboffline installer opens in a maintenance mode and allows you to reinstall or remove the current version of CRM.launcher.

## Installation on the domain user (client) machine

Installing CRM.launcher on client machine.

Non-administrative domain users are offered a download link when they first access a function that requires CRM.launcher. Clicking on the link downloads the installation file from the server and automatically starts the installation. The CRM.launcher is not installed as a service on the user’s machine but only as a program that is started up automatically when the machine starts up.

On the server the setup file is located at: “<CRM launcher root>\..\data\plugins\ CRM.launcher.exe ” by default.

For information on customizing the setup, see the article “How to Customize the set up of CRM.Launcher” <https://support.aurea.com>.

---

**Note:** The CRM.launcher setup requires .NET framework 4.5. If .NET framework 4.5 is already installed on the client, setup can be executed within the user context. Otherwise setup installs .NET framework 4.5. This installation requires administrator rights. CRM.launcher does not support "virtualizations" of shared desktops, i.e. it requires a virtual desktop integration where networking is dedicated per virtual machine, and IP addresses and ports are not shared with other users. **Restriction:** If the setup is executed from within Internet Explorer 10, no log file is created. (Workaround: If setup is started manually in the download folder, LauncherSetup.log is created in the same directory.)

---

For more information on installing CRM.launcher, see [Installing CRM.launcher](#) on page 793.

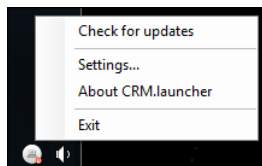
Once the setup is finished, CRM.launcher is started and the user is prompted to install a certificate that is necessary for a secure web socket communication. The prompt is displayed every time CRM.launcher is started until the certificate is installed.

For information on enabling the certificate for Mozilla Firefox, see the article "CRM.Launcher Firefox" at <https://support.aurea.com>.

Information on the currently used version of CRM.launcher can be found on Aurea CRM.Web's System Information page (**Settings > System Information**).

## Configuring CRM.launcher on user machines

As long as CRM.launcher is running, an icon is displayed in the system tray. Right- or double-clicking on the icon opens the context menu.



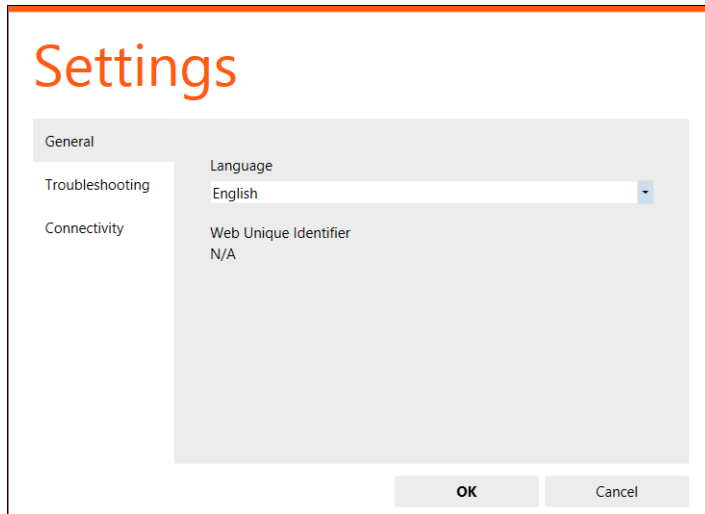
Available options:

- **Check for updates:** Checks if the version installed on the client is up-to-date, i.e. the same as on the server, and prompts the user to install patches if available, see [Updating CRM.launcher](#).

You can hide this menu item by setting `<EnableManualUpdates>` to `false` in the `update.launcher.exe.config` file.

If the installation of a patch is aborted by the user, the first menu entry reads **Resume Patching** instead: This allows the user to resume the patching process.

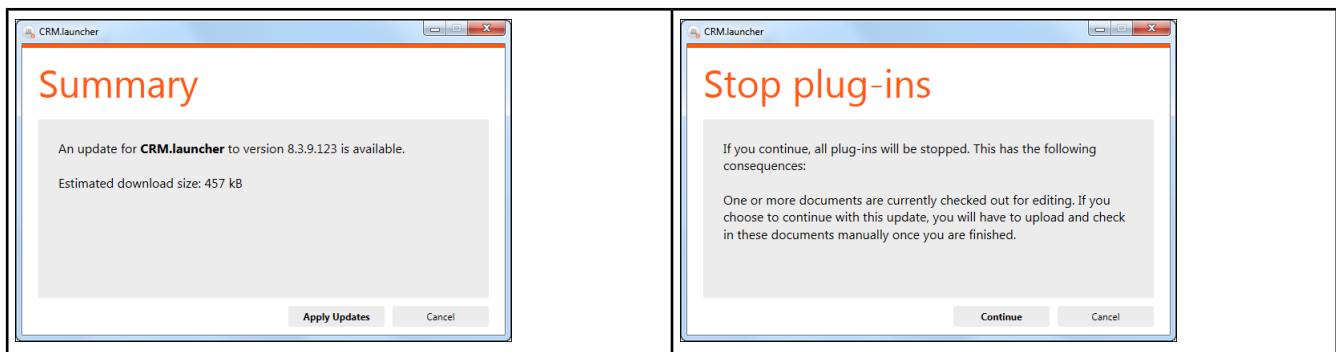
- **Settings:** The following options are available:



- **General:** Allows users to select a language, see [CRM.launcher Languages](#) on page 823.
- **Troubleshooting:** Allows users to create a bug report, see [Creating a Bug Report](#) on page 807.
- **Connectivity:** Allows users to set another port used for the connection between CRM.launcher and the browser (default: 18188).
- **About CRM.launcher:** Displays CRM.launcher's version number.
- **Exit:** Exits CRM.launcher.

## Updating CRM.launcher

CRM.launcher periodically checks for updates on the server and informs the user if an update is available. A short summary of the found files is displayed and the user is asked for confirmation. The user is also informed upfront about possible connectivity losses and advised about further actions before continuing with the update (e.g. a warning stating that open documents have to be uploaded manually). Once all files are downloaded and the update has been successfully installed, CRM.launcher restarts itself and reconnects to Aurea CRM.Web.



CRM.launcher searches for new patches in the `<GlobalPatchDirectory>` on the server (default path: `.. \web.data\GlobalPatchDirectory`). The `<GlobalPatchDirectory>` is defined in the `offline.xml` file located at `.. \system\settings`.

\* .zip files are unpacked in the root installation folder, executables are executed provided they meet the security guidelines: The <PatchingSettings> section in the update.launcher.exe.config file needs to be configured, and the files have to be signed with a valid certificate.

For details, see the articles “CRM.Launcher 8.2 Feature Pack, self patching” and “How to Configure Patching Settings for CRM Launcher” at <https://support.aurea.com>.

Executables need to be placed in the root folder. They are executed with the location of update.launcher.exe as working directory. Executables in subfolders is not executed.

Patches must comply with the following naming conventions:

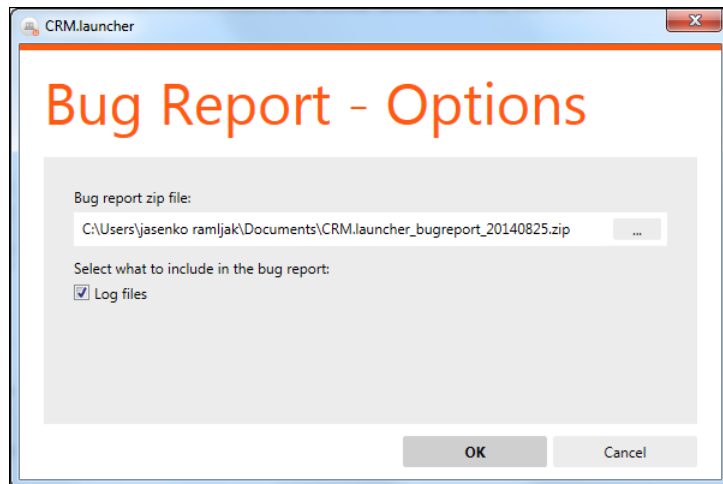
- File names must start with "launcher\_".
- To enable patching of different versions on the client, you can use the words "from", "to" and "any" in the file name.

CRM.launcher patches are processed in **numeric order** (i.e. starting with the lowest to\_<version> number).

## Creating a Bug Report

Learn about creating bug report for issues with CRM.launcher.

If there is an issue with one of the CRM.launcher plug-ins or with the update process, users can create a bug report containing all of CRM.launcher's settings files and logs (**Settings > Troubleshooting** or by clicking **Create Bug Report** directly on the error screen).



The report is created as a \*.zip package that users can then send to the administrator.

For troubleshooting information, see the article “Trouble Shooting CRM.Launcher 8.2 Feature Pack” at <https://support.aurea.com>.

## Configuring the CRM.launcher

Learn to configure the CRM.launcher.

The configuration file `Update.Launcher.NamedPipeServer.exe.config` contains the configuration settings for the CRM.launcher server installation. For individual users machines the configuration file `update.launcher.exe.config` contains the configuration settings for the CRM.launcher. The logging and certificate configuration are discussed in the below:

## Persisting User-session Data in Shared Environments

Learn about Persisting User-session Data in a shared environments in CRM.launcher.

In a shared environment, the **CRM.LauncherNamedPipe** service needs to identify and communicate with multiple Launcher user-client instances. For each client instance, CRM.Web generates a unique id (UUID) and persists the UUID by storing it in the US infoarea. Similarly, the UUID session received by the Launcher client instance is stored in a secure location in the shared environment. By persisting the UUID on CRM.Web and the CRM.Launcher clients, the Launcher session is resumed when the user logs back to the shared environment. The UUID received by the corresponding Launcher client instance is stored in a location that is not cleared by the session manager of the shared environment, when the user logs off.

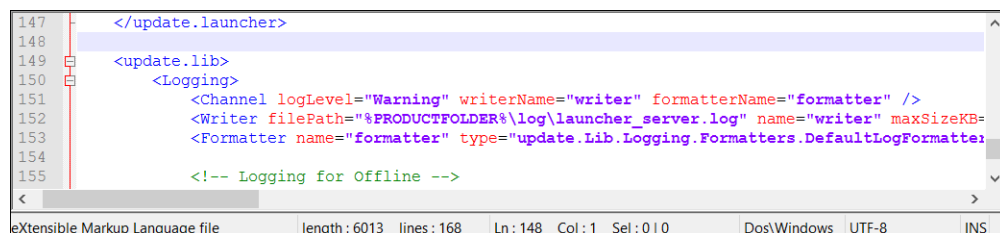
For example, the Citrix session manager cleans up the local storage used by all applications that ran in a Citrix user-client's session, including CRM.Launcher's data like the UUID, stored locally. CRM.Launcher stores the client session data and files in a folder location that is specified by the `<UserProfilePath>` parameter defined in `update.launcher.exe.config` file. If this parameter is not defined then a default path `%userprofile%\AureaCRM` is used. See the example configuration below:

```
<update.net>
  <update.launcher>
    <UserProfilePath>%userprofile%\AureaCRM</UserProfilePath>
```

## Logging Configuration

Learn to configure logging on CRM.launcher.

The following screenshot shows the `update.Launcher.NamedPipeServer.exe.config` file with the `loglevel` attribute set to "warning" and the `filepath` that points to the `launcher_server.log` file.



```
147 </update.launcher>
148
149 <update.lib>
150 <Logging>
151 <Channel logLevel="Warning" writerName="writer" formatterName="formatter" />
152 <Writer filePath="%PRODUCTFOLDER%\log\launcher_server.log" name="writer" maxSizeKB=
153 <Formatter name="formatter" type="update.Lib.Logging.Formatters.DefaultLogFormatter
154
155 <!-- Logging for Offline -->
```

The screenshot shows a code editor window displaying XML configuration for logging. The configuration includes a channel with logLevel="Warning", a writer named "writer" writing to "%PRODUCTFOLDER%\log\launcher\_server.log", and a formatter named "formatter". A comment indicates logging for offline.



The following sample code provides the logging configuration details:

```
<update.net>
...
<update.lib>
  <Logging>
    <Channel logLevel="Warning" writerName="writer" formatterName="formatter" />
    <Writer filePath="%PRODUCTFOLDER%\log\launcher_server.log" name="writer"
maxSizeKB="8192" closeOnIdleTimeout="5"
    type="update.Lib.Logging.Writers.FileLogWriter,update.Lib" />
    <Formatter name="formatter"
type="update.Lib.Logging.Formatters.DefaultLogFormatter,update.Lib" />
  </Logging>
</update.lib>
</update.net>
```

CRM.launcher allows the following log levels:

- NoLogging – Nothing is written to the log file.
- Emergency – The system is unusable.
- Alert – Actions requiring immediate attention are logged.
- Critical – Only critical conditions are recorded in the log file.
- Errors – Only errors are written to the log file.
- Warning – Only warnings are written to the log file.
- Notice – Normal but significant conditions are written to the log file.
- Info – Errors, Warning, and information messages are written to the log file.
- Debug – Everything is written to the log file.

## Certificate Configuration

Learn to configure certificates on CRM.launcher.

Aurea CRM uses the the default certificate type X.509 and is issued by Aurea. As an administrator you can also use external X.509 certificate issued by other certification providers. If you want to use third party certificates then you have to install the third party certificates on the server machine running launcher and also the client machines that connect to the CRM server.

To configure third party X.509 certificates, set the path to the certificates in the configuration file. The following screenshot shows the external certificate `mycertificate.pfx` located at the location `c:\mycertificates\`. Also, notice the URL format.



```
139 | -->
140 | <PlugIn type="update.Launcher.WebBrowser.PlugIn,update.Launcher.WebBrowser"
    | required="false">
141 |   <Url>wss://www.xyz.com:8181</Url>
142 |   <Certificate>c:\mycertificates\mycertificate.pfx</Certificate>
143 | </PlugIn>
144 | <!--
```

The screenshot shows a code editor window with an XML configuration file. The file is named "eXtensible Markup Language file" and has a length of 6133 characters and 171 lines. The current line is 143, column 26. The XML content shows a PlugIn element with type "update.Launcher.WebBrowser.PlugIn,update.Launcher.WebBrowser" and required="false". Inside the PlugIn element, there is a Url element with value "wss://www.xyz.com:8181" and a Certificate element with value "c:\mycertificates\mycertificate.pfx". The status bar at the bottom indicates the file encoding is UTF-8 and the line ending is Dos\Windows.

The following code sample shows the configuration setup in the `Update.Launcher.NamedPipeServer.exe.config` configuration file.

```
<update.net>
...

<update.launcher>
...

<PlugIns>

  <PlugIn type="update.Launcher.WebBrowser.PlugIn,update.Launcher.WebBrowser"
required="false" >
  <Url>wss://www.xyz.com:8181</Url>
  <Certificate>c:\mycertificates\mycertificate.pfx</Certificate>
  </PlugIn>
</PlugIns>
</update.launcher>
</update.net>
```

---

**Note:** The certificate must have intended purpose "Server Authentication" and its canonical name must match the host name in `wss://<url>`. The URL is the one configured for CRM.launcher. For example, in the example above `www.xyz.com`.

---

### Configuring Aurea CRM.Web to use `wss://` or `ws://` connection protocol

To provide a `wss://` (WebSocket Secure) or `ws://` (WebSocket) connection protocol for CRM.Launcher connecting to Aurea CRM Web, use the web configuration parameter `System.LauncherUrl`. The parameter provides three possible connection options, listed below:

- **System.LauncherUrl is not set** - This is the default state and in such a case the Launcher URL is automatically set to `wss://localhost:<System.LauncherPort>`. This provides a WebSocket secure connection for the Launcher, even if Aurea CRM Web server provides an insecure HTTP connection.
- **System.LauncherUrl is set to an absolute protocol** - Set it to an absolute protocol `wss://` to ensure that the Launcher always uses a WebSocket Secure connection protocol. Aurea recommends that you use this approach if you intend to use a hostname other than **localhost** and custom certificates.
- **System.LauncherUrl is set to a relative protocol** - Set it to a relative protocol URL starting with two slashes, for example, `//www.xyz.com:8181`. The protocol relative Launcher URL is prefixed with `wss:` if the CRM Web server provides a **HTTPS** (HTTP Secure) connection or prefixed with `ws:` (WebSocket connection) if the CRM Web server provides a plain **HTTP** connection.

## Adding CRM.Launcher Embedded Certificate to the Certificate Store

CRM.Launcher's certificate is installed when the Launcher is run for the first time. On further runs CRM.Launcher checks for the certificate and tries to refresh it. If the Launcher cannot refresh the certificate due to user permission restrictions, a pop up dialog informs the user that the certificate is not available. To prevent the Launcher from checking for the certificate and trying to refresh the certificate, set the `ForceRefreshCertificate` property to `false`, in the `Update.Launcher.NamedPipeServer.exe.config` or the `Update.Launcher.exe.config` configuration file. See the sample configuration below:

```
<update.net>
  ...

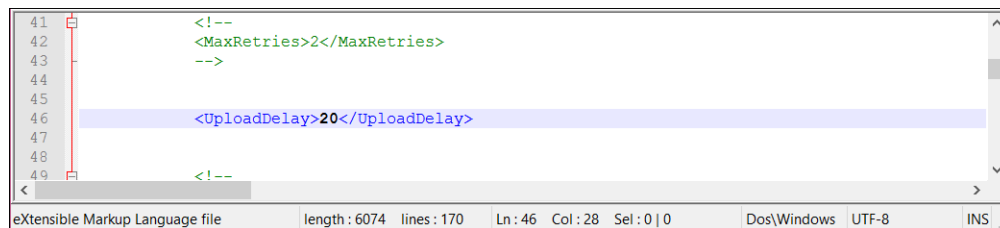
  <update.launcher>
    ...

    <PlugIns>
      <PlugIn type="update.Launcher.WebBrowser.PlugIn.update.Launcher.WebBrowser"
        required="false">
        <ForceRefreshCertificate>true</ForceRefreshCertificate>
      </PlugIn>
    </PlugIns>
  </update.launcher>
</update.net>
```

## Upload Delay Configuration

Learn to configure Upload Delay on CRM.launcher.

Sometimes when you checkout a document like Microsoft Word and you update and close the document the Microsoft Word might take more than the default upload time (5 seconds) to unlock the document. In such cases CRM.launcher might fail to upload the updated file and the document on the CRM.launcher remains locked. In such cases it makes sense to increase the default upload time. You can set that in the configuration file as shown in the screenshot below.



The following code sample shows the upload delay settings in the configuration file:

```
<update.net>
  ...
  <update.launcher>
    ...
    <UploadDelay>20</UploadDelay>
    ...
  </update.launcher>
</update.net>
```

## Configuring certificate handshake security feature

Learn to configure the enhanced certificate handshake security feature on CRM.launcher.

System administrators can configure the enhanced certificate handshake security feature supported by Aurea.CRM to ensure secure communication between CRM.Web and CRM.Launcher. This ensures that only CRM.web signed commands initiated and authorized by both the system administrator and user can be executed by CRM.launcher.

To configure the enhanced certificate handshake security feature, perform the following steps:

1. Ensure you have administrative access to the following:
  - to Aurea CRM.web deployed on a target host
  - to all the client hosts deploying CRM.launcher
2. Ensure you have a certificate for both signing and encryption (with key exchange to establish public and private keys between the CRM.launcher host and the CRM.Web host).
  - You require a certificate container (folder) on the CRM.Web host for the private key.
  - Public key for deployment on the clients hosting CRM.launcher.
  - Configure the CRM.Web and CRM.launcher host machines to synchronize time online, between varying time zones. Local NTP servers may be used.
3. Certificates can be obtained by one of the following methods:
  - Purchase a signed certificate from a certification authority, such as VeriSign, Inc.
  - Prepare your own self signed certificates. This procedure uses a self signed certificate. To create your own self signed certificate in a windows powershell, run the `New-SelfSignedCertificate` command with the following options:

```
New-SelfSignedCertificate -FriendlyName "CRM Launcher Handshake Certificate"
  -Type Custom
    -Subject "CN=CRM Launcher" -KeySpec KeyExchange -KeyUsage
DigitalSignature,DataEncipherment
  -KeyAlgorithm RSA -KeyLength 2048 -HashAlgorithm "SHA256"
-CertStoreLocation
  "Cert:\CurrentUser\My" -NotAfter (Get-Date).AddYears(99) -Provider
"Microsoft Enhanced RSA and
  AES Cryptographic Provider"
```

---

**Note:** : Run this command on the CRM.launcher host and change the value of the **CertStoreLocation** option to the current user's personal certificate container folder. You can also customize the other options according to your needs. For a synopsis of the command, see [New-SelfSignedCertificate](#).

---

This command generates a certificate valid for 99 years from current date and place the certificate into the Current User's personal certificate storage. You can then export both private and public keys from it to secure the CRM Launcher communication.

Note down the certificate thumbprint as it is required for configuring CRM.Web and CRM.launcher. For more information, see [How to: Retrieve the Thumbprint of a Certificate](#).

4. Deploy the private key to the CRM.Web host and the public key to the CRM.launcher host machines.

The private key must be deployed to the TrustedPeople of LocalMachine certificate secure windows store of the machine hosting CRM.Web. To allow the windows processes running the CRM application pool access to the private key perform the configurations described in the following article: [How to: Make X.509 Certificates Accessible to WCF](#). For example, you can use the following windows batch command to grant IIS\_USRS group read access to stored private certificate:

```
for /F %f in ('FindPrivateKey.exe TrustedPeople LocalMachine -t "CEERTIFICATE
THUMB-PRINT
    HERE" -a') do icacls %f /grant BUILTIN\IIS_IUSRS:R
```

Similarly, the public key must be deployed to TrustedPeople of LocalMachine certificate windows store of client machines hosting CRM.launcher.

---

**Note:** The FindPrivateKey.exe tool helps you to find the location and name of the private key file associated with a specific X.509 certificate in the certificate store. This program needs to be compiled from the sample source code provided as part of the Windows Communication Foundation framework.

---

5. Configure CRM.Web and CRM.launcher with the certificate thumbprint values.
  - a. To configure CRM.Web, add the thumbprint to the web.config file as shown below.

```
<appSettings>...<add key="AuthenticationCertificateThumbprint"
    value="CERTIFICATE THUMBPRINT HERE" /></appSettings>
```

- b. To configure the CRM.launcher, add the certificate thumbprint to the update.Launcher.exe.config and update.Launcher.NamedPipeServer.exe.config(if CRM.Launcher is used in a shared mode service) files as shown below:

```
<update.net>
  <update.launcher>
    ...
    <AuthenticationCertificateThumbprint>CERTIFICATE THUMBPRINT
    HERE</AuthenticationCertificateThumbprint>
  </update.launcher>
</update.net>
```

---

**Note:** The Thumbprint must be entered using uppercase letters.

---

- Restart the CRM.launcher client manually or the CRM.launcher service if you have installed it as a service.
- CRM.launcher now establishes secure certificate handshake negotiation and allows user actions without the use of the authorization request popup dialog.

---

**Note:** If for some reason secure handshake negotiation is not possible, unauthorized user actions is denied and no popus are presented either.

---

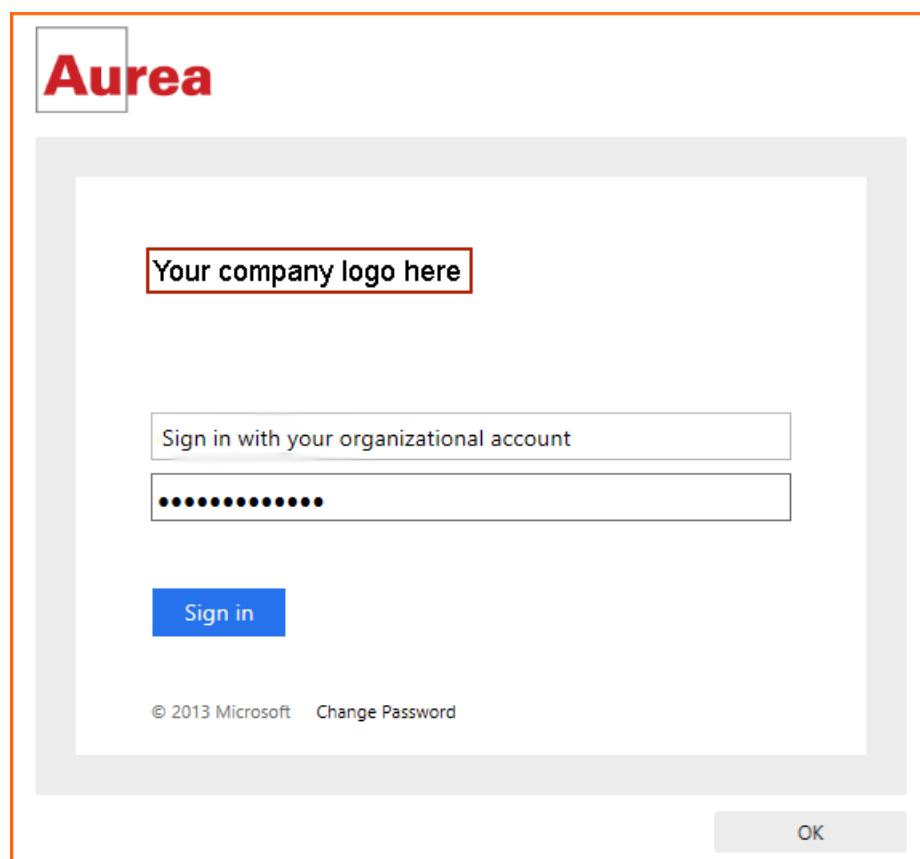
## Launcher Support for External Login via Firewalls

Learn about CRM.launcher for external third party users.

Organizations not only allow users to connect to CRM.Web from within their intranets but also allow external users to connect over the internet via non-transparent firewalls. To authenticate the external users they use their proxy or domain servers as well as external third-party providers.

Third-party providers allow organizations to customize and use their web interface to authenticate external users. CRM.Launcher supports authenticating external users by displaying their customized login forms in a pop-up window.

A sample screenshot is shown below:



To enable CRM.Launcher to authenticate users as described above, set the **ShowWebLoginPopup** property in the Launcher's application configuration file:

```
<configuration>
  <update.net>
    <update.launcher>
      <ShowWebLoginPopup>true</ShowWebLoginPopup>
    </update.launcher>
  </update.net>
</configuration>
```

As CRM.Launcher needs to communicate with the server several times and to prevent showing this popup every time a request to the server is served, it can re-use the cookies received in the first response after a successful login. To save these cookies for all future logins, set the `SaveAuthenticationCookies` property in the Launcher's application configuration file:

```
<configuration>
  <update.net>
    <update.launcher>
      <SaveAuthenticationCookies>true</SaveAuthenticationCookies>
    </update.launcher>
  </update.net>
</configuration>
```

## Controlling payload size

CRM.Launcher allows users to have full control over the payload size using the `maxJsonLength` property in the Launcher's application configuration file.

The QuickCampaign feature works better with small to medium set of records. A large set of records may lead to performance issues.

---

**Note:** Implement the changes to the launcher config file for both the client and the server

---

Add the following parameters to the config file:

```
<system.web.extensions>
  <scripting>
    <webServices>
      <jsonSerialization maxJsonLength="50000000">
      </jsonSerialization>
    </webServices>
  </scripting>
</system.web.extensions>
```

The value of the `maxJsonLength` attribute is maximum size in bytes of serialized/deserialized Json content. It is used to pack and unpack payload during communication between Launcher client and Launcher server.

## CRM.launcher Logging and Debugging Reports

Learn how to fetch login and debugging reports for CRM.launcher.

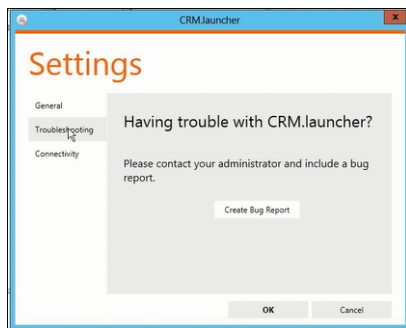
The CRM.launcher allows its different applications and the Windows server services to store and transmit logs for bug reports. It creates the following log files:

- Personal Log files: The CRM.launcher application log files for a user's session are stored in a user's personal data folder.
- Shared log files: The CRM.launcher Windows service log files are stored in the `Program Files` folder and shared between all users.
- Bug report files: The bug report functionality of the CRM.launcher collects all the relevant log files and creates a zip file.
- Log levels for application and server: CRM.launcher configuration file allows the different logging levels to be configured for the application and the Windows service.

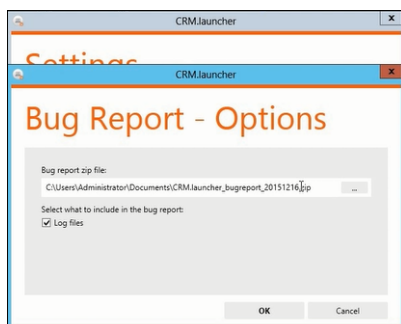
### Creating the CRM.launcher debugging reports

To create a CRM.launcher debugging report, perform the following steps:

1. Right click on CRM.launcher icon in the task bar and select `Settings`. The CRM.launcher Settings window is displayed.
2. Click the Troubleshooting tab on the left.

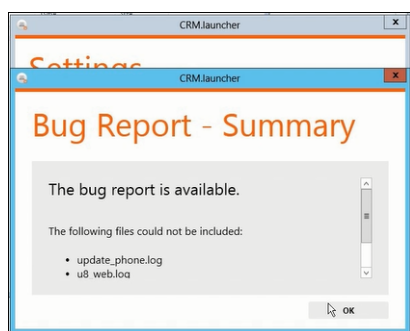


3. Click on **Create Bug Report**. The **Bug Report – Options** window displays.



4. Enter the name and path to save the zip file. Select **Log files** to include all the log files and click **OK**. The Bug Report summary displays.





5. Click **OK** to complete the debugging report generation.

## Troubleshooting CRM.launcher issues

Learn to troubleshoot CRM.launcher issues.

Below are the commonly observed issues with the CRM.launcher and their solutions:

### Certificate errors

Sometimes the CRM.launcher cannot authenticate itself with the CRM.Web server because of faulty certificates. This could be one of two reasons:

1. A security policy might be preventing the launcher from getting and adding the correct certificate to the windows certificate manager.
2. CRM.launcher has an old certificate from a previous launcher version.

**Solution:**Delete the current certificate from the "Windows Certificates Manager", and restart the CRM.launcher. It should acquire the latest certificate and use it for authentication.

The following error messages related to certificate errors are listed below along with the reasons and solutions:

- **Message** - “Cannot load external certificate <path to pfx file set in `.config` file>”

**Reason** – This message is issued when you are using your own certificates and the `pfx` cannot be accessed or does not exist.

**Solution** – Ensure that the `.config` file for the CRM.launcher contains the correct path to the certificates. For more information on configuration files, see Configuring the CRM.launcher.

- **Message** – “Cannot load embedded certificate.”

**Reason** – This can be caused by a faulty installer, which did not have the default certificate embedded in it.

**Solution** – Reinstall the CRM.launcher using the latest copy from the support site.

- **Message** – “Failed to add CRM.launcher embedded certificate to the certificate store. <detailed description>.”

**Reason** – The <detailed description> usually contains the reason for this kind of error and can vary from system to system. Two reasons commonly encountered are described below and solutions provided:

1. The user that is running the CRM.launcher has limited rights and cannot add the certificate to the certificate store.
2. The user cancels the certificate installation from the confirmation dialog box.

**Solution** – Check the detailed description provided and check with your system administrator for a solution as these errors are system specific. For the two reasons listed above, the solutions are as follows:

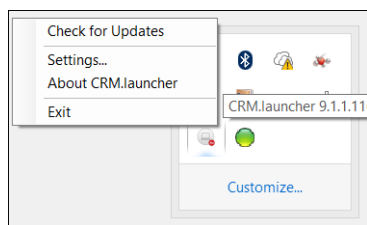
1. Expand the rights of the user to include permissions to add the certificate to the certificate store.
2. Reinstall the CRM.launcher and ensure that you accept the installation of the certificate.

## Port Conflicts

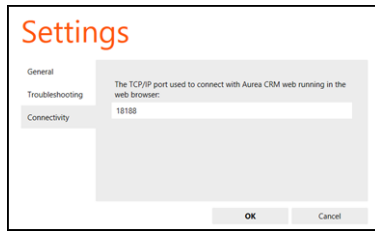
The default port used by CRM.launcher is 18188. If you are getting a port conflict, you can change the default port to another that does not conflict with other applications.

To change the port settings perform the following:

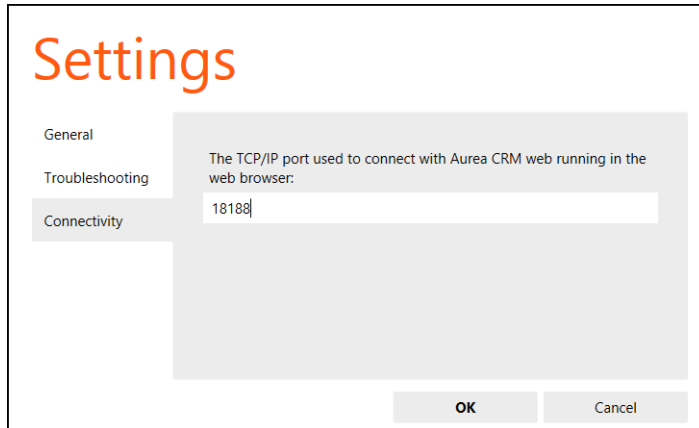
1. Right click on the CRM.launcher icon (  ) in the system tray. It displays the context menu.



2. Select **Settings**. The CRM.launcher settings window opens.



### 3. Click **Connectivity**.



### 4. Enter the new port number in the text box and click **OK**.

## IE security policy denies websocket communication with localhost

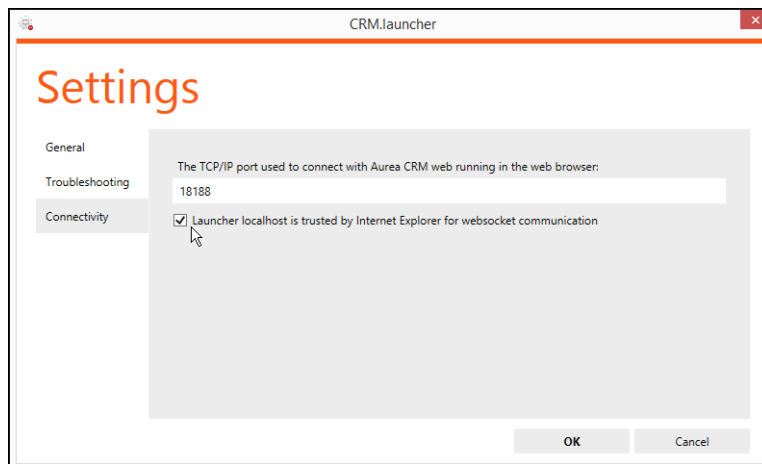
If you are using Internet Explorer, its security policy might prevent communication with the CRM.launcher server running on the localhost. To work around this, perform the following steps:


1. Set `<EnableIELocalhostWebsocketCommunication>` in the `update.launcher.exe.config` file, as shown in the code snippet below:

```
<update.net>
  <update.launcher>
    <!--
      Set this to false to disable setting up IE compatibility flag
      Default: false
    -->
    <EnableIELocalhostWebsocketCommunication>true</EnableIELocalhostWebsocketCommunication>
  </update.launcher>
</update.net>
```

If set to **true**, on startup, the CRM.launcher service ensures that the IE policy is configured to allow websocket communication with the localhost. All flags in all the CRM.launcher configuration files are updated and set to true. If this setting is not configured at all, then all flags in all the CRM.launcher configuration files are set to false.

2. If the `<EnableIELocalhostWebsocketCommunication>` setting is enabled and set to true then the **Connectivity** tab in the CRM.launcher settings window shows the “**Launcher localhost is trusted by Internet Explorer for websocket communication**” option as checked. See the screenshot below. It is not checked if the `<EnableIELocalhostWebsocketCommunication>` setting is not enabled at all.



**Note:** To open the Settings window, right click on the CRM.launcher client icon (  ) running in the system tray, and select **Settings....**

If you wish to disable or enable websocket communication between the Internet Explorer and the localhost then you can do it from the above settings window by unchecking and checking the the **Launcher localhost is trusted by Internet Explorer for websocket communication** option in the settings window. This is valid only for the current session. In the next session, when the CRM.launcher service is started up, the configured setting is activated again. For example, if the `<EnableIE-LocalhostWebsocketCommunication>` setting is enabled and set to true and you had disabled websocket communication with localhost by unchecking the option in the Settings window, then the new session has the websocket communication enabled.

## Communication with Aurea CRM.Web

CRM.launcher communication with Aurea CRM.Web.

The CRM.launcher performs the following tasks to mediate communication between the browser running in a shared environment and Aurea CRM.Web application:

- It maintains different unique identifiers to distinguish between user's web browser login sessions.
- It allows the user to activate the CRM.launcher session from the web browser.
- If the user closes the web browser and re-opens the browser again, the CRM.launcher gets the unique identifier for the user's previous session.
- The CRM.launcher is independent of the Windows service. In case the CRM.launcher is not launched as a windows service, a user can still launch the CRM.launcher server for this session.

---

**Note:** The user has to be a part of the administrator group, in order to be able to start the service.

---

- CRM.launcher service for multiple users is not activated for the standard version of Aurea CRM.Web application. This is only allowed for the Enterprise version of Aurea CRM web application.

---

**Note:** If multiple users try to activate CRM.launcher session from the Standard version of Aurea CRM, the following message is displayed: Multiple login sessions for CRM.launcher is an enterprise feature only. To use this functionality, please upgrade your system to an enterprise edition.

---

CRM.launcher and Aurea CRM.Web communicate over a secure Web Socket sub-protocol named WAMP. For further details, see <http://wamp.ws>. The communication only accepts a secure connection. Therefore the user has to install a certificate when CRM.launcher is first started.

For more information, see the article “CRM.Launcher and Security” at <https://support.aurea.com>.

To enable the communication between CRM.launcher and Aurea CRM.Web, both products need to know which port to use (default: 18188).

For Aurea CRM.Web the port is configured via the `System.LauncherPort` Web Configuration parameter. For further details, see [System.LauncherPort](#) in the *Web Administrator's Guide*. This port needs to match the port configured for CRM.launcher:

- Users can configure the port in CRM.launcher's **Settings** dialog (**Connectivity** tab). The configured port is saved in the `user.config` file that is used for CRM.launcher. The `user.config` file is located in the root directory of the installation, i.e. the same location as `update.Launcher.exe` (default: `%localappdata%/update.CRM`).
- If you don't want users to be able to change this setting, specify the port in the `update.launcher.exe.config` file as follows:

```
<update.launcher>
  [...]
  <!-- The list of plug-ins to be loaded by CRM.launcher -->
  <PlugIns>
    <!--
    <PlugIn type="update.Launcher.Offline.PlugIn,update.Launcher.Offline"
required="true" />
    -->
    <PlugIn type="update.Launcher.WebBrowser.PlugIn,update.Launcher.WebBrowser"
required="false" />
    <Url>wss://localhost:11771</Url>
    <PlugIn
type="update.Launcher.RemoteControl.PlugIn,update.Launcher.RemoteControl"
required="false" />
    <PlugIn type="update.Launcher.Documents.PlugIn,update.Launcher.Documents"
required="false" />
    <PlugIn type="update.Launcher.Phone.PlugIn,update.Launcher.Phone"
required="false" />
```

```
</PlugIns>  
</update.launcher>
```

## CRM.launcher communication with CRM.phone plugin

CRM.launcher allows the CRM.phone plugin to work on with a single user or with shared user environments.

For each user session of the CRM.phone plugin, the CRM.launcher stores the user's session information in user's local `AppData` folder. The CRM.phone plugin user session data include the logging files and call events like ringing event etc. The CRM.launcher performs the following tasks:

- The CRM.launcher allows the CRM.phone plugin to work and store its data and configuration files.
- For a shared environment, the CRM.launcher also allows the CRM.phone plugin to store and work with data and configuration files stored in the `Program Files` folder.
- The CRM.launcher helps the CRM.phone plugin to distinguish between the different user sessions. The CRM.phone creates events for each user session and these events need to be linked to the correct user session, when multiple users are using the CRM.phone plugin. For example, an incoming call events need to be redirected to the correct user's CRM.phone session by the CRM.launcher. Similarly, it would associate a user's outgoing call events to the correct CRM.phone session of the user.

## CRM.launcher communication with Document plugin

CRM.launcher allows the CRM Document plugin to allow multiple user sessions to work with the Document plugin.

When a user opens a document to work with it, CRM.launcher associates the file with the specific user's session and locks the file and closes the file associated with the user's session. CRM.launcher registers the correct document event, opens the document with the correct application and maintains the document events with the correct user's document session. CRM.launcher locks the files being opened and prevents conflicts between users opening and updating the same files.

---

**Note:** CRM.Launcher cannot work with files whose filename (including extension) along with the filepath exceeds 256 characters.

---

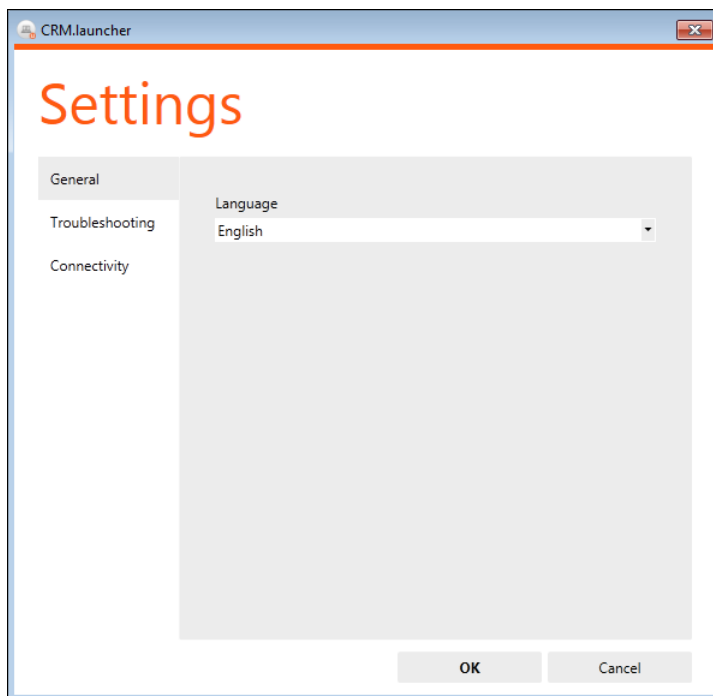
For a document session, CRM.launcher performs the following:

- CRM.launcher allows the Document plugin to work and store its document event data and configuration files.
- For a shared environment, the CRM.launcher also allows the Document plugin to store and work with event data and configuration files stored in the `Program Files` folder.
- CRM.launcher helps the Document plugin to distinguish between the different user sessions. The Document plugin creates events for each user session and these events need to be linked to the correct user session, when multiple users are using the Document plugin.

## CRM.launcher Languages

By default, CRM.launcher uses the same language as the user's operating system provided that this is a supported language, otherwise English is used.

User can change the language used by CRM.launcher via **Settings > General > Language**.



The user's language selection is saved per user in `%LocalAppData%`.

The UI strings and messages displayed by CRM.launcher are read from XML files located at `<launcher_InstallDir>\Resources\Texts`.

To add a language, create an `update.launcher.Texts.<language>.xml` file in this folder. You need to enter the ISO 639-2 three-letter code as `<language>`, e.g. `update.launcher.Texts.fra.xml`.

Use the SDL Passolo XML format:

```
<?xml version="1.0" encoding="utf-8" ?>
<pslXMLExportImport>
  <pslTransList Id="" Source-Language="en" Target-Language="en">
    <Entry Id="ButtonCancel" Origin="" Comment="" ><![CDATA[Cancel]]></Entry>
    <Entry Id="ButtonClose" Origin="" Comment="" ><![CDATA[Close]]></Entry>
    <Entry Id="ButtonOk" Origin="" Comment="" ><![CDATA[OK]]></Entry>
    [...]
  </pslTransList>
</pslXMLExportImport>
```

Additional CRM.launcher languages need to be deployed as global CRM.launcher patch. For further details, see [Updating CRM.launcher](#).

## CRM.Phone

Learn about the CRM.Phone.

For detailed information on CRM.Phone, see [CRM.Phone User Guide](#) and [CRM.Phone Administrator Guide](#).

Fields are defined as telephone fields in the data model. For example, **Telephone** column, Type "Telephone" and "Cell phone". For further details, see [Data Model](#) in the [CRM.Core Administrator Guide](#).

Aurea CRM.Web works with CRM.Phone using CRM.launcher. See [CRM.launcher\(Enterprise\)](#) for details.

If CRM.Phone is not installed, Aurea CRM.Web uses the `tel:` protocol for phone numbers.

## Displaying a Record's History

Track changes to records in the history info area of Aurea CRM.

In the **History** info area (H0) Aurea CRM tracks who made which changes to a record, and when. This provides an overview of the history of the record. For general information, see [History](#) in the *CRM.Core Administrator Guide*.

In Aurea CRM.Web the history can be displayed in the **Related Data** area of a record.



ID	Date & Time	Field	Changed by	Old Value	New Value	Application	Windows Comput...	Windows User ...
2	15.11.2012 16:12	Item name	Kevin Veith	web BTB MS-SQL	web BTB MS-SQLi	web	ATPC2131.update.c...	u8.0
1	20.01.2012 10:42	Item No.	Irmgard Sabet...		565656	web	updevweb02.update...	v8.sp0.bb
1	20.01.2012 10:42	Product Group	Irmgard Sabet...	Briefmarken	Software	web	updevweb02.update...	v8.sp0.bb
1	20.01.2012 10:42	Business Area	Irmgard Sabet...		Software	web	updevweb02.update...	v8.sp0.bb

To reduce the columns displayed in the history sub-list enable the `Record.History.Slim` Web Configuration parameter, see [Record.History.Slim](#).

Date & Time	Field	Changed by	Old Value	New Value
15.11.2012 16:12	Item name	Kevin Veith	web BTB MS-SQL	web BTB MS-SQLi
20.01.2012 10:42	Item No.	Irmgard Sabet-Wasinger		565656
20.01.2012 10:42	Product Group	Irmgard Sabet-Wasinger	Briefmarken	Software
20.01.2012 10:42	Business Area	Irmgard Sabet-Wasinger		Software

UPDATE\_DEFAULT contains sample implementations of the history for the following info areas:

- BTB: **Installed Base** (IB),
- FS: **Product Overview** (PU)
- OTC: **Sample** (MU)

To enable the history for an info area:

1. Aurea CRM win Rights module: Enable the history for the desired info areas in the **Rights** module by checking the **History** column. For further details, see [Enabling the History](#) in the *CRM.Core Administrator Guide*.

To display the history for Expand and Search&List views, adjust the header configuration of the info areas for which you want to display the history:

1. Add the info area **History** (H0) to the respective `ExpandChildren` and `SearchSubList` headers. For further details, see [Related Data area \(Header Info Areas\)](#) on page 329.
2. Specify the `RecordHistory` form in the **Form** column.

Configuration of the SearchSubList-Header of Header Group 'IB' ✓ Save

Text: **Related Data: {SUBINFOS} of {RECORD}**

Image:

Show buttons with text  Show buttons in extra line  Use plain style (no borders)  Use page header style  Fixed tab width (SublistTabWidth)

Description:

**Buttons**

**Infoareas**

Order	InfoArea	Link	CRM Proc	Fieldgroup	Header	Text	Filter	Tab Style	SubListParams	Menu	Search	Form Name
✗	Installed Base Item (KO)	<input type="checkbox"/>	all	no value	no value	Default	no value				no value	
✗	Participant (RO)	<input type="checkbox"/>	all	RO_IB	ROSublistNew	Default	no value		RefreshOnParent!		no value	
✗	Activity (MA)	<input type="checkbox"/>	all	no value	no value	Default	no value				no value	
✗	Installed Base (IB)	<input type="checkbox"/>	all	no value	IB-NoNew	Sub-Installed Bases	no value				no value	
✗	Service Contract Item (WP)	<input type="checkbox"/>	all	WPIBSublis	WPSublistNoNe	Default	no value				no value	
✗	Ticket (KM)	<input type="checkbox"/>	all	KM_IB	KMIBSublistNes	Default	no value				no value	
✗	Document link (D3)	<input checked="" type="checkbox"/>	all	no value	no value	Default	no value				no value	
✗	History (H0)	<input type="checkbox"/>	all	no value	no value	Default	no value				no value	RecordHistory

To display the history for a Tree view, add a new node for the tree's root info area:

The special header `RecordHistory` contains the buttons for changing the history view (**Most Recent Changes | All Changes**). You can specify the default in the `RecordHistory` form's definition: `view=<OverviewChanges | OverviewAllChanged-Fields>` (default = `OverviewChanges`, i.e. **Most Recent Changes**). For further details, see [Forms](#) on page 621.

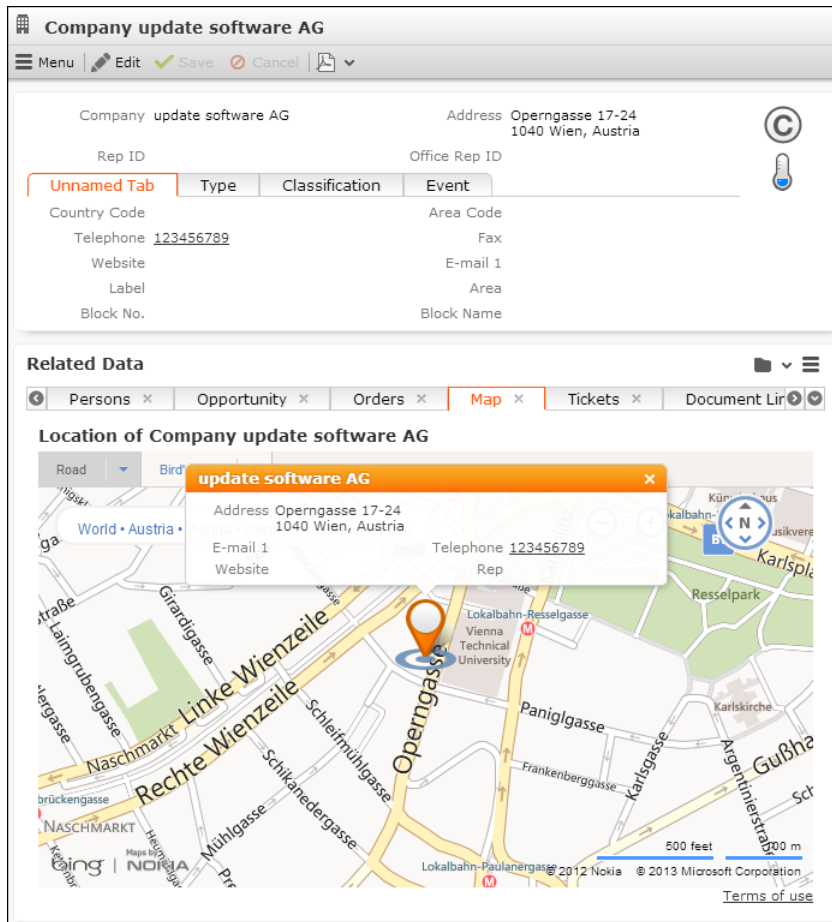
## Displaying Bing Maps in Related Areas and Forms

It is possible to display records from Aurea CRM in Bing Maps by defining a map panel (using the `RecordMap` widget) for the Related Area and in forms.

`UPDATE_DEFAULT` contains sample configurations for FI, KP, ZA and OJ (BTB only). for further details, see [Configuring a Map Panel](#) on page 828.

**Note:** The CRM process 1.5.2.5.2 "Geography" (CRM process ID 70000130) to which the map panels configured in `UPDATE_DEFAULT` are assigned, is disabled by default (since accessing a map panel without a valid Bing Maps license results in an error).

For information on the configuration steps required in CRM.Core (license key, query string format etc.).



You can configure map panels for all info areas that contain geographic coordinate fields and/or (default) address fields.

If a record contains geographic coordinates, the map uses them. Otherwise, the address fields of the record are used. Aurea CRM.Web does not automatically calculate geographic coordinates based on address fields.

To define other than the default address fields (Country, ZipCode, City, Street) to be used for the map, use the **Field Mapping** configuration parameter (**Geo Coordinates** category) in the **Configuration** info area (MC), see [Configuration Info Area](#) in the *CRM.Core Administrator Guide*. Use the **Query format** parameter to define the format of the query using these custom fields.

If a map is configured in a child info area (that does not contain address fields), the geographic coordinates or address to be used is determined in the following order: FI > KP > PE. Example: A map in MA uses the data from PE only if no KP/CP exists for that person.

You can change this default fallback behavior for child info areas via the `Geolocation.FindAddressStrategy` Web Configuration parameter, see [Geolocation.FindAddressStrategy](#).

Displaying a map for MA: If a rep of type "Resource" (i.e. a meeting room etc.) is added as **Participant** to a MA record, the address of this resource is used for the map, if available. If more than one resource is added as **Participant**, the first resource with valid address data is used.

## Configuring a Map Panel

Learn to configure a Map panel.

The configuration of a map panel consists of

- the integration of the `RecordMap` widget in a `SearchSublist` or `ExpandChildren` header or in a form,
- a `MiniDetails` control definition in field group `<InfoAreaID>.Address` for the respective info area (fallback: `<InfoAreaID>`).

To configure a maps panel for the **Related Area** (e.g. the `ExpandChildren` header for FI as configured in `UPDATE_DEFAULT`):

1. Add the info area of the records you want to display on the map to the `ExpandChildren` header, see [Related Data area \(Header Info Areas\)](#) on page 329.

Configuration of the ExpandChildren-Header of Header Group 'FI'

Text **Related Data: {SUBINFOS} of {RECORD}**

Image  Show buttons with text  Show buttons in extra line  Use plain style (no borders)  Use page header style  Fixed tab

Description Enter description here.

**Buttons**

**Infoareas**

Order	InfoArea	Link	CRM Proc	Fieldgroup	Header	Text	Form Name
	Person in Company (KP)	Default	all	Q	KPSublistNew		
	Activity (MA)	Default	all	Q	MASublistNew		
	[...]						
	Company (FI)	Default	all	Q		Map	{ type: "RecordMap", options: { size: { height: 400 } } }

2. Enter a name for the tab (or panel) in the **Text** field.
3. Define the map panel in the **Form Name** field using the `RecordMap` widget:

```
{
  type: "RecordMap"
}
```

4. Create a field group `<InfoAreaID>.Address` for the info area and define the `MiniDetails` control (e.g. field group `FI.Address` in `UPDATE_DEFAULT`). This control is used for the popup in the Bing Map. The header of the popup displays the info area's default table caption.

To use the `RecordMap` widget in a form:

The `RecordMap` widget is available in the Form Designer (under CRM-specific widgets). It accepts one or more record uids as `value`. For further details, see [Data Binding](#) on page 628.

**Examples:**

---

```
$component(url).@uid
```

This binds the `RecordMap` widget to a record selected e.g. in a `RecordListBox`.

```
$widget(listView1).@selection(rowsUid)
```

This binds the widget to multiple records selected in a `ListView`.

---

**Note:** For information on all available options for the `RecordMap` widget, see the “How to Use Maps” at <https://support.aurea.com>.

---

## Documents

Learn to work with documents in Aurea CRM.

The default info area for uploading documents is the **Documents** info area (D1), i.e. record for newly uploaded documents are created in D1. The **Customer Documents** info area (D2) is only used for existing D2 records. No new D2 records can be created.

If uploaded from the Related Data area of a parent record (or from within a document field), the document is created in D1 and linked to its parent via D3.

For general information on documents in Aurea CRM, see [Aurea CRM win User Manual](#).


For information on [Document Management](#), see *Aurea CRM.Web Business Logic User Guide*.

The following topics provide information on working with documents in Aurea CRM.Web:

## Access Rights

Learn to define rights for document info areas.

Defining access rights for **Document** info areas (D1/D2/D3):

- Users need **Update** rights for D1 for the record selector icon  to be displayed, i.e. if **Deny Update** is defined for D1, users can neither edit existing nor upload new documents in document fields or the **Document** info area itself.
- If **Deny Access** is defined for D1, document fields display "Access denied" and D3 records are not displayed.
- To hide all D3 SubLists, define **Deny Access** for the D3 info area.

For further details, see [Rights](#) in the *CRM.Core Administrator Guide*.

## Configuring MIME Types

Learn to configure MIME type.

For users to be able to work with different document types (DOC, DOCX, RTF, PDF, etc.) in Aurea CRM.Web the file extensions must be registered on the web server: To do this, you need to configure the corresponding MIME types.

There are two possibilities to configure the MIME types:

- The required MIME types must be added to the Windows registry on the web server (in `HKLM\HKEY_CLASSES_ROOT\MIME\Database\Content Type` and `HKLM\HKEY_CLASSES_ROOT\[.extension]`).
- In addition, you can define MIME types in the `MimeTypes.xml` which is referenced via the `settings.xml` file's `<update.web>` section. (Both files are located at `..\system\settings`).

The most common MIME types are already part of the delivered `MimeTypes.xml`.

## Uploading Documents

Learn to upload on Aurea CRM.Web.

Users can upload local documents into Aurea CRM.Web via **Collaboration > Documents > New** or from the Related Data area of a record (**Document Links** tab > **New > Upload new document**).

For further details, see [Saving Documents in the Database](#) in the *Web Business Logic User Guide*.

By default, only one file can be uploaded at a time. You can enable uploading multiple files in the **Configuration** info area (MC) (**Documents** category, **Allow multiple file uploads/drag&drop** option). For further details, see [Configuration Info Area](#) in the *CRM.Core Administrator Guide*. If this option is enabled, users can also link multiple existing documents to a record (**Document Links** tab > **New > Existing update.CRM document**).

---

**Note:** Document fields never support multiple selection.

---

To modify the **Upload Document** dialog, see [Special Field Groups Naming Conventions](#) on page 190 and [Document.UploadFieldGroup.D1/D2](#). The values you enter e.g. as **Keyword** in the **Upload Document** dialog are applied to all uploaded files.

The type of documents that can be uploaded to Aurea CRM.Web can be configured in the `settings.xml` file. Add the list of file mime types to the parameter `<UploadFileMimeTypes>` under the `<update.crm.base>` element. The mime type names should be separated by a semicolon. If this parameter is not defined or is empty then there are no restrictions applied to the file types that can be uploaded. To get the complete list of mime types, see <https://www.iana.org/assignments/mediatypes/media-types.xhtml>.

The following example, shows the mime types for files that can be uploaded to Aurea CRM.Web. The mime types `mword`, `pdf`, `onenote`, `rtf`, `mp4`, `mpeg`, and `xml` are listed in the `<UploadFileMimeTypes>` parameter.

```
<update.crm.base>
  <UploadFileMimeTypes>
    application/mword;application/pdf;application/onenote;
application/rar;application/rtf;video/mp4;video/mpeg;application/xml;
  </UploadFileMimeTypes>
</update.crm.base>
```

The size of documents users may upload to Aurea CRM.Web is determined by the upload size limit configured for the server, i.e. the `maxRequestLength` attribute of element `<httpRuntime/>` in the `web.config` file (default: 4 MB). If the limit is exceeded the upload process is aborted with "Internal Server Error".


To show an explicit error message on the client, you need to configure the IIS to return detailed errors.


## Document fields

Learn to define documents fields.

Document fields can be defined:

- in the data model ("Document" property checked, see [Data Model](#) in the *CRM.Core Administrator Guide*)
- by setting the `Document` field attribute for any text field, see [Document](#) on page 224.

In New and Edit mode a record selector icon  is displayed next to the field. Clicking on it displays the `M_SelectDocument` context menu which contains options for selecting an existing or uploading a new document or image, see [M\\_SelectDocument](#).

Users need Update rights for D1 (and D2) for the record selector icon  to be displayed, see [Access Rights](#) on page 829. When uploading a new document, it is created in D1 and linked to its parent via D3.

In View mode a link to the document is displayed.

For image files you can use the `Image` field attribute to display a preview of the image instead. The maximum image size is 2048x2048px. For further details, see [Image](#) on page 229.

In lists, document fields display the document's key instead of its title except D3 lists.

To display the titles of the D3 SubList entries as links, check the `Extended Options` attribute and define `UseRootRecord=true`. For further details, see [Extended Options](#) on page 228.

## Document Check-Out/-In

To prevent multiple users from editing the same document at once, users can check out a document and thus lock it for other users.

This functionality is handled by a plug-in, see [CRM.launcher\(Enterprise\)](#) on page 793.


Documents are checked out using the `D_Document_Edit` menu action (in `M_Documents` for D1 and D2 and in `M_ExpandLinked_Delete` for D3 records), see [M\\_Documents](#).

The application used for opening documents is determined by the Edit association of the file type (defined in the registry).

To display a list of currently checked out documents:

- Users click **Collaboration > Checked Out Documents** (`ID_NET_CHECKEDOUT_DOCUMENTS_MY`), which lists the documents checked out by the current user.
- Administrators click **Administration > Manage Checked Out Documents** (`ID_NET_CHECKEDOUT_DOCUMENTS_ALL`), which lists all currently checked out documents of all users.

In both cases the result list displays the Process Tracking records (T3) that have been created on checking out documents (`T3.Documents` and `T3.Documents.Admin`).

- In addition, the navigation pane item  (**Checked Out Documents**) is displayed as long as at least one document is checked out. (The navigation pane item and popup is controlled by the plug-in and can not be customized.) The popup lists the documents checked out by the current user in a popup. When a new session is started, a system message containing a link to `ID_NET_CHECKEDOUT_DOCUMENTS_MY` is displayed instead of the **Checked Out Documents** menu item.

A document can be checked in/a check out can be undone as follows:

- upon closing the application in which the document is opened.
- selecting **Upload & Check-In** (`A_Document_CheckIn`) or **Undo Check Out** (`D_UndoCheckout`) from the context menu (`M_Documents_Edited`) in the popup, see [M\\_Documents\\_Edited](#).
- selecting **Undo Check Out** (`D_UndoCheckout`) from the context menus for D1, D2, D3 and from the checked out documents lists.
- upon logging out and selecting the option **Continue logout and undo check out**.

The administrator can undo the check out for all checked out documents. The respective user is informed by a message.



The `Document.AutoUpload` Web Configuration parameter defines whether users are prompted for automatically uploading a non-MS Office document (e.g. PDF, TXT etc.) after editing or need to manually upload such documents, see [Document.AutoUpload](#).

## Context Menus and Menu Actions

Learn about document related context menus and menu actions.

The following document-related context menus and menu actions are available in UPDATE\_DEFAULT.

### M\_Documents

The context menu used for the document info areas (D1, D2). For D3 the `M_ExpandLinked_Delete` context menu is used instead.

`M_Documents` contains the following menu actions:

- [A\\_Expand](#)
- [D\\_Document\\_Edit](#)
- [D\\_UndoCheckout](#)
- [A\\_DeleteDocumentEx](#)
- [A\\_Document\\_Open](#)

### M\_NewDocumentLink

The context menu that is used in D3 sub-list headers, i.e. for the **Document Links** tab in the Related Data area of e.g. FI. `M_NewDocumentLink` is also configured e.g. as sub-menu of `M_Appointments`.

It contains the following menu actions:


- `A_NewDocumentLink_Local`: Allows users to upload one or more local documents linked to the current record. For further details, see [Uploading Documents](#) on page 830.
- `A_NewDocumentLink_Existing`: Allows users to link one or more existing documents to the current record.

---

**Note:** Uploading/creating multiple files is enabled in the **Configuration** info area (MC) (**Documents** category, **Allow multiple file uploads / drag&drop** option).

---

### M\_SelectDocument

The context menu that is displayed when clicking on the record selector icon  displayed next to a document field, >> [Document fields](#) on page 831.

It contains the following menu actions:

- `A_SelectDocumentOpen`: Opens a document saved in the Aurea CRM.Web database without check out.
- `A_SelectDocumentRecord`: Adds an existing D1 document to the document field.

- `D_SelectCustomerDocumentRecord`: Adds an existing D2 document to the document field. This menu action is only active if the record has a KP or FI record as parent. It is not available when in New mode since no parent record is available at that time.
- `A_SelectDocumentUpload`: Uploads a local document to the Aurea CRM.Web database.

### **M\_Documents\_CheckedOut**

The context menu used for the T3 records used to list checked out documents. For further details, see [Document Check-Out/-In](#) on page 832.

`M_Documents_CheckedOut` contains the following menu actions:

- [D\\_UndoCheckout](#)
- `A_FromT3ToDocument`: Navigates to document record of the checked out document.

If more than one T3 record is selected, the `M_Documents_CheckedOut.MultiSelect` menu is used. It only contains `D_UndoCheckout`.

### **M\_Documents\_Edited**

The context menu used for the **Checked Out Documents** popup.

It contains the following menu actions:

- [A\\_Document\\_CheckIn](#)
- [D\\_UndoCheckout](#)

### **D\_Document\_Edit**

Checks out the selected document (D1, D2, D3) and opens it in its default application. For further details, see [Document Check-Out/-In](#) on page 832.

The application used for opening documents is determined by the Edit association of the file type (defined in the registry).

### **D\_UndoCheckout**

Unlocks the document without uploading a new version.

### **A\_DeleteDocumentEx**

This is the standard menu action for deleting documents (D1,D2) and document links (D3) as defined in UPDATE\_DEFAULT: It provides assistance to users when they are about to delete documents still used in other places.

### **A\_Document\_Open**

Opens the document without check out.

### **A\_Document\_CheckIn**



Checks in a new version of the document and removes the lock.

---

## emails

Learn how to send email from Aurea CRM.Web.

Users can send emails from within Aurea CRM.Web:

- by clicking on a field containing an email address (View mode) or on the **Send email** icon  beside it (Edit mode). For further details, see [Mail](#) on page 225.
- In the **Activity** info area the  **Send as E-Mail** button (**Send as email**) allows users to send **Activity** records via email. For further details, see [Send As email](#) on page 835.
- The **Notify Rep** menu item allows users to send the current record via email. For further details, see [Notify Rep](#) on page 837.
- reply to activities of type "email". For further details, see [email Reply](#) on page 839.

For information about configuring the sending of emails from within Aurea CRM refer to the [mail server](#) in *Aurea CRM.Web Installation Guide* and [SMTP settings](#) in *CRM.Core Administrator Guide*.

For a description of all email-related Web Configuration parameters, see [Email Parameters](#) on page 107.

When sending an email from within Aurea CRM, all email fields (i.e. fields with the property "email" in the data model) are taken into consideration: The first email field that is not empty is actually used. The fields are checked in the order of their unique field IDs (starting with the lowest ID).

---

**Note:** To send an email in a language other than the default languages, you should use UTF-16 encoding for the template document. For example, to send an email in Cyrillic, encode the template document in UTF-16. You can use any text editor that supports UTF-16, to encode the template document.

---

## Send As email

Learn how Send As email works.

The `OpenAppointmentEmail` button  **Send as E-Mail** in the MA `Expand` header in `UPDATE_DEFAULT` allows users to send a **Activity** record as an email.

**Send as email** works as follows:

- You can configure which internal recipients (reps) are added to the email. For further details, see [Internal Recipients](#).
- You can configure which external recipients (companies, persons) are added to the email. For further details, see [External Recipients](#). On the emails recipients list external recipients are always listed before the internal ones ("customer first").
- The content of the **Subject** field (MA/103 in the BTB version) is automatically entered as the subject of the email.
- The content of the **Text** field (MA/23 in the BTB version) is automatically entered as the text of the email. To specify another field, use the Web Configuration parameter [EMail.BodyFieldID](#).

To send HTML-formatted emails, create an HTML field in MA and specify it as `EMail.BodyFieldID` (server-side sending only). For further details, see [HTML Fields](#) on page 219.

Client-side emails do not support HTML-formatted text. For further details, see [EMail.SendingMethod](#).

- All documents linked to the record are offered as attachments for the server-side sending of emails. (For client-side emails no attachments can be added automatically.)

To limit the attachments offered for **Send as email** to certain content types and/or document classes use the Web Configuration parameters `EMail.AttachmentsExcludedContentTypes` and `EMail.AttachmentsExcludedDocumentClasses`. For further details, see [EMail.AttachmentsExcludedContentTypes](#) and [EMail.AttachmentsExcludedDocumentClasses](#).

---

## Internal Recipients

The internal recipients for appointments (MA) sent as emails are determined as follows:

- If a field group called `MA.EmailReps` exists for MA, the internal recipients are determined using the fields contained in the Details control of this field group:
  - Only rep fields and participant fields (i.e. fields with `type="uid"` in the data model) from the current info area may be added to the Details control. All other fields are ignored.
  - No internal reps are offered if the Details control is empty.
  - All reps that were determined using `MA.EmailReps` are added to the email in the **To:** field. Participants are divided into **To:**, **CC:** and **BCC:** according to their participation type.
- If the field group `MA.EmailReps` exists but is empty, no reps are used as recipients in the email (i.e. it is only addressed to the person or company of the appointment). For further details, see [External Recipients](#).
- If the field group `MA.EmailReps` does not exist, internal recipients are determined using the participant fields (if available). Rep fields are not taken into account.

The user can define in his settings if he (i.e. the currently logged on user) should be added to the internal recipients or if he should be ignored: The Web Configuration parameter `EMail.ExcludeCurrentUser`, which is included in the UserConfig in `UPDATE_DEFAULT`, controls whether the current user is included when sending emails.

Default = not checked.

For further details, see [EMail.ExcludeCurrentUser](#).

## External Recipients

External recipients are determined from the **Activity Participant** info area (MB). External recipients are always listed before internal ones in the recipient list of the email ("customer first").

The Web Configuration parameter `EMail.FallbackToCompany` determines if the company's email address is used as a fallback when a person's email is empty. For further details, see [EMail.FallbackToCompany](#).

## Notify Rep

The **Notify Rep** menu item (`A_NotifyRep`) allows users to send the current record via email.

This functionality is mainly intended to notify co-workers about records where their action is required.

**Notify Rep** is configured in the context menus of **Activities**, **Offers** and **To-Dos** in `UPDATE_DEFAULT` (BTB version).

**Notify Rep** works as follows:

- You can configure which internal recipients (reps) are added to the email. For further details, see [Internal Recipients](#).
- You can configure which external recipients (e.g. companies, persons) are added to the email. For further details, see [External Recipients](#). On the emails recipients list external recipients are always listed before the internal ones ("customer first").
- The subject of the email is automatically entered if a field with the XML name "Subject" exists in the respective info area. If no such field is available, the default table caption is used instead.
- The body text of the email is automatically entered if a field with the XML name "Text" exists in the respective info area.

To use another field for the email body text, specify the field in the `EMail.NotifyRep.BodyFieldIDs` Web Configuration parameter, e.g. enter "PR:OffText" to use the **Text** field from the **Offer** info area. For further details, see [EMail.NotifyRep.BodyFieldIDs](#). One field per info area is supported.

- If a signature is defined for a user (in the **Rep** info area), an **Insert Signature** button is available, allowing the user to insert his signature into the **Body** field.
- All documents linked to the record are offered as attachment for server-side sending of emails. For client-side emails no attachments can be added. For further details, see [EMail.SendingMethod](#).
- Set the **useHtmlInput** argument to `true` to enable rich text content in the email message editor.
- To limit the attachments offered for **Notify Rep** to certain content types and/or document classes use the Web Configuration parameters `EMail.AttachmentsExcludedContentTypes` and `EMail.AttachmentsExcludedDocumentClasses`. For further details, see [EMail.AttachmentsExcludedContentTypes](#) and [EMail.AttachmentsExcludedDocumentClasses](#).

## Internal Recipients

The internal recipients for mails sent via Notify Rep are determined as follows:

- If a field group called `<InfoAreaId>.NotifyReps` exists for the respective info area, the internal recipients are determined using the fields contained in the Details control of this field group:
  - Only rep fields and participant fields (type "uid" in the data model) from the current info area may be added to the Details control. All other fields are ignored.
  - No internal reps are offered if the Details control is empty.
  - All reps that were determined using `<InfoAreaId>.NotifyReps` are added to the email in the **To:** field. Participants are divided into **To:**, **CC:** and **BCC:** according to their participation type.
- If the field group `<InfoAreaId>.NotifyReps` does not exist for an info area, internal recipients are determined using the participant fields (if available). Rep fields are not taken into account.

The user can define in his settings if he (i.e. the currently logged on user) should be added to the internal recipients or if he should be ignored: The Web Configuration parameter `EEmail.ExcludeCurrentUser` controls whether the current user is included when sending emails (default = not checked). For further details, see [EEmail.Exclude-CurrentUser](#).

Internal reps are checked by default in the **Notify Rep** dialog.

## External Recipients

When sending **Activity** records, external recipients are determined from the **Activity Participant** info area (MB).

External recipients are always listed before internal ones in the recipient list of the email ("customer first").


The **Notify Rep** dialog also offers external recipients for info areas containing email addresses (e.g. **Company**, **Person**, and **Additional Address**).

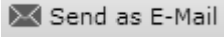
By default, the Email recipients are unchecked in the **Notify Rep** dialog, specify `{activateEmailFieldsRecipients: true}` in the `CommandArguments` of `Email/SendRecordNotification` command mentioned under `A_NotifyRep` menu action will mark recipients from e-mail fields as active.

To exclude external recipients from the notify rep functionality (e.g. if Notify Rep is used solely for company-internal communication), specify `{excludeExternalRecipients: true}` in the `CommandArguments` of the `A_NotifyRep` menu action.

## email Reply

Learn how to reply to email from Aurea CRM.Web.

The `EmailReply` button  allows users to reply to email activities from within Aurea CRM.Web. For further details, see [EmailReply](#) on page 310.

By clicking the `EmailReply` button or pressing Ctrl+R a new email activity is created. Users can write the email and then send it by using the  button. For further details, see [Send As email](#) on page 835.

When replying to an email activity the following fields are copied from the initial email activity to the new one:

- **Subject and Text**

Text no. 33 ("RE: ") of the Text Group `email_text` is inserted before the **Subject**, text no. 35 ("{1}----- Original Message {0} -----{1}") is inserted before **Text**. Text no. 35 supports special placeholders: {0} for the current date, {1} for a line break.

- **Activity Type, Purpose, Priority and Private**

- for each activity participant (internal participants (i.e. reps), companies and persons) a new **Activity Participant** record (MB) is created and linked to the new activity record:

- The following fields are copied to the new MB records: **Co.Gr.**, **Co.No.**, **Pe.Grp.**, **Pe.No.** and **Part. Type** keeping the same participation types (required, optional and for information).
- The **Part. Status** field is reset to "open".

**Starts on** and **Starts at** are set to the current date and time, the **Rep** field is set to the currently logged in user.

The new activity is linked to same parent record (**Company** or **Person**) as the original activity. The original activity is linked to the new activity via the **Previous/Main Activity** field.

The fields copied to the new MA and/or MB records can be extended by defining a Details control for a `MA.ReplyFields` and/or `MB.ReplyFields` field group. All fields contained in the(se) control(s) are copied to the new records.

## External Address Check

The external address check allows users to validate addresses against address data provided by an external system.

To activate the external address check:

- Define the required entries in the **Configuration** info area (MC).
- Create and define the `Expand.AddressCheck.InfoAreas` Web Configuration parameter to determine for which info areas the address check should be performed, see [Expand.AddressCheck.InfoAreas](#).



The address check is called when creating new records and when editing a field defined as `upfields` in the **Configuration** info area (MC) (**External Address** category, **Parameters** option).

The address check is called from both Expand and List views.

When saving a record, a popup dialog displays the address entered by the user and a list of addresses suggested by the external system (sorted by score):

**Address Check**

You have entered the following address information:

Country Austria      Postal Code 1040  
City Wien              Street

**Results (5 record(s))**

Score	Street	Postal Code	City	Country
98	Operngasse 25	1040	Vienna	Austria
97	Operngasse 50	1040	Vienna	Austria
96	Opernheide 17-24	1050	Vienna	Austria
95	Opernheide 17-24	1040	Vienna	Austria
94	Opernheide 17-24	1040	Vienna	Belgium

■ Identical value   
 ■ Different value   
 ■ Empty record value   
 ■ Invalid value

OK      Cancel

The fields displayed on the **Address Check** dialog are configured by the special field group `<InfoAreaId>.AddressCheck` (Details and List control). If no `<InfoAreaId>.AddressCheck` is defined, the `upfields` defined in the MC entry (**External Address** category, **Parameters** option) are used as a fallback. If no `upfields` are defined, the fields actually updated in the record are displayed.

If the external address check returns only one result with a score above the defined threshold (MC, **External Address** category, **Threshold** option), the record is automatically updated with this address, and no **Address Check** dialog is displayed. The user receives a notification that the address data was checked and corrected.

If **Threshold** is defined, and multiple results are returned, the address with the highest score is automatically used.

Clicking **Cancel** saves the user's original values. Clicking **OK** updates the record with the currently selected address. (If the match-up is activated as well, the match-up dialog is displayed after the address check dialog.)

All values returned by the external address check are written to the record. Empty values overwrite existing values.

---

**Note:** You can not return empty values for Boolean fields (e.g. `<FreeL1></FreeL1>`); Boolean fields must have a value.

---

Address suggestions containing a catalog value that does not exist in the database are displayed in gray (= invalid). If the user selects an address containing an invalid value, a warning is displayed.

For information on calling the external address check via add-on programming, see the article "How to configure the address check" at <https://support.aurea.com>.

The color-coding of the results is defined in the `addresscheck.css` file (`.u8-addresscheck-cell-identical` etc.) located in `..\styles\update\crm\widgets`.

## Favorites

Learn to configure info areas where records can be added as favourite.

To configure the info areas for which users can add records as favorites:

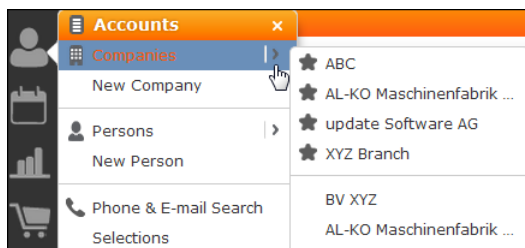
1. Add the info area to the `Favorites.InfoAreas` Web Configuration parameter, see [Favorites.InfoAreas](#).
2. Add the menu action `D_Favorites` (**Add to favorites**) to the info area's menu (e.g. `M_Company` in `UPDATE_DEFAULT`), see [Configuring a Menu](#) on page 292.
3. To display the favorites as Quick Links in the application menu, enter the info area's ID as **QuickLink Context** (e.g. `ID_NET_COMPANY_SEARCH` in `UPDATE_DEFAULT`), see [Menu Actions](#) on page 295.

In the Favorites popup the `SimpleList` header (displaying "`{INFOAREAIMAGE} {INFOS}`") by default) is used for the section headers, see [SimpleList](#).

For the favorite records, the respective info area's default table caption is used.



For the Quick Links you can define special table captions `<InfoAreaId>QuickLinks`. If undefined, the info area's default table caption is used.



# Letters & Mail Merge

Learn to send single mail or create mass mailing in Aurea CRM Web.

You can create single letters and mass mailings in Aurea CRM.Web:

- Single letters can be sent based on any record that matches the settings in the Word Processor format. For further details, see [Mail Merge Configuration](#) on page 844
- Mass mailings ("Serial letters") can be used for both quick campaigns and marketing activities. For further details, see [Quick Campaigns](#) on page 845 and [Mail Merge Letters Based on Marketing Activities](#) on page 849.

Prerequisites for creating letters and mass mailings in Aurea CRM.Web:

- Define the required formats: Word Processor formats in Aurea CRM win. For further details, see **Defining Transfer Formats** in the *Aurea CRM win User Manual* ([https://help.aurea.com/crm/crm-win-admin-guide/crm.win.admin.guide/#topics/c\\_win\\_adm-1\\_1-Intoduction\\_crm.html](https://help.aurea.com/crm/crm-win-admin-guide/crm.win.admin.guide/#topics/c_win_adm-1_1-Intoduction_crm.html)).

These formats contain the following information:

- the name of the document template
- a read engine format (or query) that defines the available fields as placeholders
- a label format containing the letterhead and salutation
- settings for creating, saving and signing the resulting letter

---

**Note:** Word Processor formats using queries cannot be used for the creation of single and serial letters in Aurea CRM win.

---

---

**Note:** Queries used for Word Processor formats must not contain parameters (`$parValue`), i.e. may not require user input.

---

- Define the required document templates as Microsoft Word documents.

---

**Note:** The document templates must be owned by the administrator.

---

- Microsoft Word must be installed on the clients. For information on supported versions of Microsoft Word, >> System Requirements available from <https://support.aurea.com>.
- Select the option "Used by Mail Merge" to make the query appear on the template document used for Mail merge.

## Mail Merge Configuration

This topic describes how to configure the mail merge functionality for creating letters from single records in Aurea CRM.Web.

When the user selects the menu action `A_MailMerge`, Aurea CRM.Web connects to Microsoft Word to display a document template from the Aurea CRM database. Data from the current record is merged with the document template according to the Word Processor format definition.

The mail merge is executed by a plug-in. For further details, see [CRM.launcher\(Enterprise\)](#) on page 793.

The user can modify the letter and then upload it to the Aurea CRM database as a D1 document linked to the current record via a document link (D3).

To configure the single letter functionality:

1. Create menu entries/buttons using the `A_MailMerge` menu action for the info areas you want your users to be able to create letters from (e.g. MA, PR).

This calls the `MailMerge` action template. For further details, see [MailMerge](#) on page 430.

2. Define any additional settings (e.g. for filtering the available formats) as arguments in the `MailMerge` action template. For further details, see [MailMerge Action Parameters](#) on page 430.

Clicking the menu item/button results in the creation of a letter. If your configuration of the `MailMerge` action template results in multiple Word Processor/transfer formats, a dialog is displayed where the user can select a format.

---

**Note:** The single letter feature first tries to load the format with the given name and the logged-in user as owner, and - if it does not find such a format - it loads the administrator format with that name.

---

When the letter is created, its default name is generated as `letter_{tablecaption}_{rep}{extension}` (e.g. `letter_Aurea Software_Joe Smith.docx`).

You can define customized file names using variables (**Save document as** field in the word processor format). For further details, see [File Name Variables](#) on page 850.

Characters that are not valid in file names (\*,' etc.) are automatically replaced with "\_".

If the document name does not already contain a valid file extension, an extension is added automatically. Valid extensions are those that are registered on the web server (via Windows registry or `settings.xml`). For further details, see [Configuring MIME Types](#) on page 830.

The file extension (e.g. ".pdf") is determined by the MIME type (e.g. "application/pdf") of the document. If no mime type is defined, the extension of the document template is used. If the document template has no extension, ".docx" is used as a fallback.

## Quick Campaigns

A quick campaign allows the user to select several records from a list and create mail merge letters for the associated persons or companies.

In addition, users can carry out a quick campaign for queries that are either explicitly called using the **Queries** menu item, embedded in a form (e.g. a start page) or carried out using a `QueryRun` action.

Quick campaigns can be launched

- from any list that supports selecting multiple records: Companies, persons and records that have either a person or a company as parent.

The screenshot shows the 'Search Companies' interface. At the top, there are search filters for 'Company' (with a dropdown menu), 'Country', 'From PC', 'To PC', and 'With me' (checked). A 'Start Search' button is on the right. Below the filters is a 'Search Results' section with a table. The table has columns: Company, Lead Status, Country, Pos..., City, and Street. A context menu is open over the row for 'Fratelli Pasquini', with 'Quick Campaign' highlighted. The 'Quick Campaign' button in the top right of the table area is highlighted in green. The table data is as follows:

Company	Lead Status	Country	Pos...	City	Street
ABC AG	Interest	Austria	1030	Wien	Paniglgasse 3
Firma 3	Cold	Austria			
Fratelli Pasquini	Lead	Italy	00100	Roma	
	Cold	Austria			
	Cold	Italy	43121	Parma	
	Cold	M_QuickCampaigns in M_Company.MultiSelect		Washington	
	Lead	Spain			
Query Test	Cold	Italy			
update sales GmbH	Cold	Austria			
Vienna City Hotel	Cold	Austria	1030	Wien	
ZZZ & Comp.	Cold	Austria	1030	Wien	Löwengasse 35

**Note:** If a quick campaign is configured for an info area linked to multiple companies and/or persons (e.g. KM), the company/person with the first link (i.e. the default link) is used as the addressee of the letter.

- from query results (but not query snapshots)

The screenshot shows a software interface for a query titled 'Query FI/KP\_DACH'. The header of the 'Query Result' table includes several action buttons: 'Export (MS Excel)', 'Save as Query Snapshot', and 'Quick Campaign' (which is highlighted with a green box). The table itself has columns for 'Company', 'Synonym', 'Country', 'Post...', 'F...', 'First Name', and 'Last Name'. The data rows are as follows:

Company	Synonym	Country	Post...	F...	First Name	Last Name
ABC AG	ABC	Austria	1030	Mr.	Self	Admin
ABC AG	ABC	Austria	1030		Franz	Bauer
ABC AG	ABC	Austria	1030		Karl	Popper
BemComp		Austria	1010	Ms.	Berta	Banner
Froberger & Comp.	Froberger	Austria		Mr.	Johann Jakob	Froberger

Result rows containing multiple uids (FI/KP/CP/PE records) are handled as follows:

- The **last** FI/KP/CP/PE uid in the result set row is used.

---

**Note:** This has nothing to do with the order of the output fields; the "last" uid is the last one determined by the query statement.

---



- If no such uid exist, the last uid whose info area contains a default link to a FI, KP, CP or PE is used to find a parent: The first actually existing parent uid (in the order KP/CP/PE/FI) is used.

---

**Note:** For quick campaigns launched from a query only fields from FI/KP/CP/PE records are supported, i.e. you can not create a mail merge letter containing e.g. data from a linked offer in this context.

---

In UPDATE\_DEFAULT the quick campaign functionality is available as follows:

- The `Quick Campaign` button  is configured in the `FI.SearchResults`, `FITree.SearchResults`, `PE.SearchResults`, `KP.SearchResults` and `KPTree.SearchResults` headers. It is active as soon as at least one record is selected from the list.
- the `Quick Campaign Query` button  is configured in the `QueryResult` header. It is always active. Therefore, when executing a quick campaign for a query without selecting any result rows, the quick campaign is executed for **all** records found in the database that match the query criteria.

If the query result exceeds 1000 result records, a warning is displayed allowing the user to cancel creating the quick campaign.

If one or more result row(s) are selected, the quick campaign is executed for these records.

- The menu action `A_QuickCampaign_Selection` (in the menu `M_QuickCampaigns`) is configured in the `M_Company.MultiSelect` and `M_Person.MultiSelect` menus.

It is available as soon as the user selects more than one record from a search or query result list that contains FI, KP or PE records.

When selecting a **Quick Campaign** button/menu entry, the **Format selection** dialog displays the available Word Processor formats. For further details, see [Specifying a Word Processor Format](#) section below. Selecting a format creates a Word document containing the letters for the quick campaign. All relevant records for the quick campaign are created automatically, see [Quick Campaign Records](#).

---

**Note:** Make sure all users working with quick campaigns have Microsoft Word installed. If a quick campaign is executed on a client without Microsoft Word being installed, an error message is displayed. However, the quick campaign records are still created in this case.

---

To offer quick campaigns for other info areas (that have either company or person as parent):

1. Create a context menu named `XXX.Multiselect` (XXX being the name of the context menu used for the Search&List configuration of that info area, e.g. `M_Company.Multiselect`). The easiest way is to inherit the new menu from `M_MultiSelect`.

This context menu is then used for multi-selection of the respective info area.

2. Add the `M_QuickCampaigns` menu to `XXX.Multiselect`.

### Quick Campaign Records

The following records are automatically created when running a quick campaign:

- **Campaign** (CM)
- **Marketing Activity** (AK) linked to the campaign
- Depending on the Web Configuration parameter `Marketing.QuickCampaign.Record-sType` **Mailings** (BR) or **Activities** (MA) linked to the marketing activity. For further details, see [Marketing.QuickCampaign.RecordsType](#).
- **Document** (D1) linked via D3 to the marketing activity

The following fields are automatically entered during a quick campaign:

Info Area	Field No.	Field Name	Value
CM	2	Campaign Type	catalog filled according to text nr. 50 of text group <code>net_campaign_mgmt</code>
CM	3	Name	text nr. 48 of text group <code>net_campaign_mgmt</code> Default: Quick Campaign '<name of document>'
CM	4	Responsible Rep ID	current rep
CM	9	Campaign Status	set to "completed"

Info Area	Field No.	Field Name	Value
CM	5, 6	Starts on, Ends on	current date
AK	2	MarketingActivity	catalog filled according to text nr. 51 of text group <code>net_campaign_mgmt</code>
AK	7	Name	text nr. 49 of text group <code>net_campaign_mgmt</code> Default: <name of document> This text is limited to 20 characters.
AK	24	Responsible Rep ID	current rep
AK	9	Status	set to "completed"
AK	5	Starts on	current date
MA	0	Activity Type	set to "letter"
MA	2, 3	Starts on, Starts at	current date and time
MA	8	Rep ID	current rep
BR	0	Date	current date
BR	37	Channel	set to "letter"

**Note:** Texts number 48 to 51 from the text group `net_campaign_mgmt` are used when creating a quick campaign. The placeholder {0} refers to the name of the single letter. Texts number 50 and 51 contain the catalog values for campaign and activity type. update suggests that the administrator creates these catalog values - using the catalog maintenance of the Aurea CRM win **Maintenance** module - before the first quick campaign is created to avoid the automatic creation of multiple catalog values in multi-language environments.



## Specifying a Word Processor Format

To restrict the selection of document templates in a certain context, you can configure menu items for specific Word Processor formats:

1. Configure a `XXX.Multiselect` menu (e.g. for the **Offer** info area).
2. Create a menu action with the following parameters:
  - **Action:** `Command`
  - **Command:** `CampaignManagement.CreateQuickCampaign`
  - **RecordSet:** `selected`
  - **FormatName:** `<Name of desired the Word Processor format>`

When selecting the menu action, the quick campaign is launched and the serial letter is created using the specified format.

PR-SerNo	Date
2070000158	04.12.2007
2080000022	23.04.2008
2080000028	04.06.2008
2080000029	04.06.2008
2893432601	03.05.2012

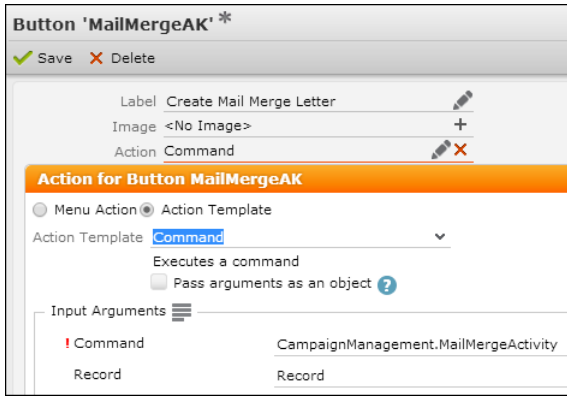
## Mail Merge Letters Based on Marketing Activities

Learn to configure mail merge letters based on marketing activities.

For information on configuring and generating marketing activities, see [Marketing Activities](#) in the *Web Business Logic User Guide*.

To create mail merge letters from within a marketing activity, the target group's addresses etc. need to be merged with the document template. The recipients of the mail merge letter are determined based on the control file created by CRM.server when generating the marketing activity. The control file is a D1 document, linked to the activity execution (AD).

To configure this functionality in Aurea CRM.Web, you need to define a button (or context menu action) with action template `Command` for the **Marketing Activity** info area (AK), e.g. in the `Expand` header of header group `AK`:



This button executes the command `CampaignManagement.MailMergeActivity` for the current marketing activity (`Record`). It merges the document template stored in the field **Letter: Template Document** (AD/78) and the control file (`Brief.txt`) linked to the activity execution thus creating the mail merge letter in Microsoft Word.

**Note:** If more documents are linked to the AD record, the first file is used. The template document is filtered automatically.

**Note:** To send an email in a language other than the default languages, you should use UTF-16 encoding for the template document. For example, to send an email in Cyrillic, encode the template document in UTF-16. You can use any text editor that supports UTF-16, to encode the template document.

## File Name Variables

You can define variable document names using placeholders enclosed in `{}`.

These placeholders are replaced with the actual values when creating the letter.

The following placeholders are available:

Placeholder	Description
<code>documentclass</code>	The document class as defined in the word processor format.
<code>extension</code>	The extension (e.g. ".pdf") for the MIME type (e.g. "application/pdf") of the document. If no mime type is defined the extension of the document template is used. If the document template has no extension ".docx" is used as a fallback.  The placeholder <code>{extension}</code> includes a leading ".".
<code>formatname</code>	The name of the word processor format.

Placeholder	Description
infoareaid	The info area ID of the record the letter is created for.
keyword	The keyword as defined in the word processor format.
now	The current time (HHMMSS).
recordid	The record ID of the record the letter is created for.
rep	The rep name of the current user.
repid	The rep ID of the current user.
tablecaption[:name]	<p>The table caption of the record the letter is created for.</p> <p>Optionally the name of a custom table caption can be specified (separated by a colon): <code>tablecaption:&lt;TableCaption_Name&gt;</code></p> <hr/> <p><b>Note:</b> This placeholder is supported in Aurea CRM.Web only.</p> <hr/>
template	The file name of the template document (without extension).
today	The current date (YYYYMMDD).
user	The login name of the current user.

### Examples:

Definition	Result
<code>{template}_{tablecaption}{extension}</code>	Invitation_Mr. John Doe.pdf
<code>{formatname}{repid}</code>	Offer100026.docx
<code>order_{today}</code>	order_20120531.docx

## Document Caching & Performance Considerations

To improve performance, Aurea CRM.Web allows web browsers to cache the documents they download.

By default, browsers are allowed to cache documents for 10 seconds. You can change this duration by setting entry `<DocumentCacheDurationInSeconds />` in the `settings.xml` file.

**Note:** Setting `<DocumentCacheDurationInSeconds/>` to a value `<= 0` disables the cache. This has a serious impact on performance and might also facilitate a DoS (Denials of Service) attack. Disabling the cache is therefore **not** recommended for productive environments.

Furthermore Aurea CRM.Web caches template documents for mail merge letters in the file system. This cache is updated automatically based on the last modification date of the document record. Although not recommended, you can disable this cache by unchecking the `Document.MailMergeCache` Web Configuration parameter. For further details, see [Document.MailMergeCache](#).

The location of this cache is defined by the `<MailMergeDownloadPath/>` entry in the `settings.xml` file.

## Merging Companies and Persons

Two company or person records can be merged into a single record in Aurea CRM.Web to remove duplicates.

**Merge**  
Source: ABC 12  
Target: update Software AG

Prepare Resolve 0% Execute

**Overview of records to be merged**

Source Record (will be deleted)	Target Record (will be kept)
Company: ABC 12	Company: update Software AG
Postal Code: 1010	Postal Code: 1040
City: Wien	City: Wien
Street:	Street: Opengasse 17-21
Country: Austria	Country: Austria

Switch records

**Merge method**

When a conflict occurs:

- User decides conflict resolution.
- Source record always wins.
- Target record always wins.
- In case of conflicts the newer record wins.

Cancel Continue

The merge process (`MergeRecords`, see [Processes](#) on page 547) is called using the menu actions `A_WebMerge_Companies` and `A_WebMerge_Persons` (and `D_WebMerge_Multiple_Companies` und `D_WebMerge_Multiple_Persons` for multi-select).

The process can be started from a single source record (the user then selects the target record in the 1st step) or with both records in that case, the user starts with the conflict resolution step. If more than two records are selected, the menu action is disabled.

The `MergeRecords` process supports the following parameters which you can specify as input arguments for the menu action:

- `fixedFilterName: "<InfoAreaId>.<FilterName>"` adds a fixed filter for the search.
- `copyFields: <field names or numbers>` defines the fields to copy from the source record into the search.
- `mergeOptions: ["userDecides", "sourceWins", "targetWins", "mostRecentWins"]` array defining which conflict resolution options are displayed beneath the merge overview under **Merge method**.
- `defaultMergeOption` sets the default merge option.
- `showEqualValues: <true|false>` defines whether fields with identical content should be displayed in the merge overview in case of a conflict. (It some case it may be helpful to display fields with identical content, because e.g. the correct postal code is hard to judge if you do not see the corresponding city; on the other hand that means that users sees a lot of fields, including system fields and fields which they probably haven't seen before.)
- `useTransaction: <true|false>` If `true` (default), the merge is executed using transactions. For further details, see [Support for Database Transactions](#) on page 855.
- `listParams: <Name>,<Value>;<Name>,<Value>;...` defines `listParams` for the search results list. For further details, see [ListParams](#) on page 374.

The merge overview (of source and target record) uses the Field Group `FIMergeOverview / KPMergeOverview / PEMergeOverview / CPMergeOverview`. (The Details controls in these Field Groups only support one column.)

The actual merge process may take some time depending on the amount of child records.

The merge is performed with the rights of the currently logged-in user. Therefore conflicts may occur during the merge process which the user cannot solve due to his rights definition (e.g. because he cannot access this data). Since merging records where the user cannot see the contents makes little sense and can lead to inconsistencies, the merge process is stopped in this case. The merge then has to be performed by a person with adequate rights (e.g. an administrator).

---

**Note:** Since the source record is deleted during the merge process, users need "Delete" rights to be able to merge records. To prevent users from deleting records, but enable them to perform a merge: Grant the users "Delete" rights, but hide all **Delete** options from the user interface using the **Hide button/Visible** check boxes

of the corresponding buttons and menu actions, see [Configuring a Button](#) on page 305 and [Menu Actions](#) on page 295.

---

## Merging Properties (BTB only)

Learn about merging properties available for BTB customers.

Analogous to the company and person merge, the BTB version allows users to merge property records (OJ) using the same `MergeRecords` process. For further details, see [Merging Companies and Persons](#) on page 852.

When merging properties, linked **Property Competitor** (OM) records are merged as well (match-up by **Competitor**, **Product Group** and **Product**). For more information on [Properties](#), see *BTB Web Business Logic BTB User Guide*.

The `MergeRecords` process is called by the following menu actions:

- `A_WebMerge_Properties`
- `D_WebMerge_Multiple_Properties`

These menu items are assigned to CRM process 1.2.3.2.9. "Merge Properties" (CRM process ID 70000290).

## Refreshing Data after C(R)UD Events

Learn how data is refreshed after CRUD events.

Aurea CRM.Web's CRUD service takes care of automatically refreshing data in lists, query results, details views and headers as well as in the calendar, daily & weekly planning, resource view, dispatching dashboard and in timelines. When the user creates, updates or deletes data on another tab or in a popup, these changes are reflected in the other open tabs once the user switches back to them.

---

**Note:** This behavior only applies to changes the user made in his current browser session. Changes to data on the server e.g. by another user are not immediately reflected on the client.

---

**Note:** Due to performance issues query results are only refreshed on `QueryRun` pages, i.e. not in the Query Designer.

---

All CRUD events are handled in the `ResultSet` model, which has a Boolean property `autoRefresh` that activates the monitoring of the CRUD events. When displaying lists and query results, this property is automatically set to `true`.

---

CRUD events are handled as follows:

- `create`: The result set is re-loaded. The newly created record(s) are only visible if the sort order/filter is set in a way that the new record(s) appear on the current page.
- `update`: Only the values in those rows that contain the updated record are updated. The result set is not reloaded entirely.
- `delete`: The rows containing the deleted record are marked as deleted and displayed in a disabled state (grayed out). If the currently selected row is disabled, the row below this row becomes the new selected row. If there is no row below the selected row, no row is selected.

## Support for Database Transactions

Data integrity is insured during database transactions.

To ensure data integrity Aurea CRM.Web automatically uses transactions when:

- creating/updating records in Expand view
- updating records List view
- editing multiple records
- merging records (companies, persons).

Although not recommended, transactions can be deactivated (via the `useTransaction` input argument) when merging records. For further details, see [Merging Companies and Persons](#) on page 852.

---

**Note:** If records were created or changed with the option `AddNewCatalogValues = true`, new catalog values are created by design even if the transaction is rolled back.

---

Sometimes when saving records, a failed insert operation might roll back the entire transaction. To prevent such situations you might want to disable the database transaction mode. To prevent a save operation in an expand page, from being in transaction scope set the url parameter `disableTransaction=true` and append it to an expand url. This prevents rollback on a database save operations in case of any failures. See the following example:

```
crm/expand?uid=FI%3Bx0000000100000001&mode=Show&infoAreaId=FI
&_action=A_Expand&disableTransaction=true
```

In addition, methods for server-side add-on programming (`BeginTransaction`, `CommitTransaction`, `RollbackTransaction` and `IsActive`) are available.

## Tuning the Performance of the Notification Engine

CRM.Web server administrators can limit the number of concurrent database sessions started by the Notification engine.

This helps improve the performance of the Notification engine when a number of users are working with CRM.Web at the same time.

The following configuration parameters can be set in the `settings.xml` file under the `NHibernateSettings` node of the `update.NotificationEngine` plugin configuration in the `update.server` section:

- `SessionPoolLimit` (default value is 16) - set the maximum number of NHibernate sessions in the Notification Engine. Increasing this value does allow Notification Engine to perform faster when many users are working with CRM.web at the same time. The minimum valid value is 1.
- `SessionTimeout` (default value is 30000 milliseconds) - set the time for which the NHibernate session can be idle.

The following sample configuration shows the concurrency settings for the Notification engine in the `settings.xml` file:

```
<!-- settings of update.server -->
<update.server>
  <!--
  PlugIns:
  A list plugin assembly that will be loaded by update.sever.
  The order of this list matters!
  -->
  <PlugIns>
    <... skipped ...>
    <PlugIn type="update.Web.NotificationsEnginePlugIn,update.Web" >
    <... skipped ...>
      <NHibernateSettings>
        <SessionPoolLimit>10</SessionPoolLimit>
        <SessionTimeout>30000</SessionTimeout>
      </NHibernateSettings>
    </PlugIn>
  </PlugIns>
</update.server>
```

## XML Exports & Reporting

This topic explains how reports are generated in update.CRM web.

Aurea CRM features a newly built general-purpose reporting engine. It operates on XML data, but by using output transformations it can generate PDF or HTML reports.

The Aurea CRM reporting engine operates on XML data. By using output transformations it can generate e.g. PDF or HTML reports. For details, see the article “Reports” at <https://support.aurea.com>.



Aurea CRM.Web knows three different types of XML exports each with its own XML schema:

- **List:** A list of records usually the result of a search or a query
- **Expand:** Details about a single record based on a field group
- **Tree:** A hierarchy containing the table captions for a set of linked records based on a TreeView configuration

Lists can be exported using two different XML schemata:

- Aurea CRM.Web schema (default)
- Aurea CRM win schema

Exports and reports are executed via buttons. For further details, see [Export/Report Buttons](#) on page 310.

For information on executing reports via JavaScript, see the “How to Execute a Report from Script” at <https://support.aurea.com>.

For information on customizing PDF reports using Aurea CRM's reporting plug-in, see the article “How to use the reporting PDF plug-in” at <https://support.aurea.com>.

Use the **SDK** page to explore the reporting functionality and test your reports: **SDK > Reporting**.

For tips on how to improve report performance, see “How to Improve Report Performance” at <https://support.aurea.com>.

## XML Output

This topic explains XML output.

For detailed information on the XML output, see [Reports](#) in the *Aurea CRM win User Manual*.

The tag `<LocaleVersion/>` at the beginning of the XML output contains the current regional settings (like date format, decimal separator, time zone etc.) to make processing and formatting of values easier.

The values of date, time, Boolean, short, long and float fields are output in invariant format (in the attribute value) as well as in display format.

If `EmitNativeFormat` is specified, date fields do not contain a time portion, time fields do not contain a date portion, and Boolean fields are output as "t" and "f" instead of "true" and "false" .

Empty fields are output as `<field.../>`.

Catalog fields contain the numerical value (in the `catval` attribute) as well as the text value e.g. `<field tableshort="FI" fieldno="5" fid="5" catval="20">Austria</field>`

## Export Options

You can specify export options to customize the XML output.

Export options are specified as `CommandArguments` (`report` parameter) for Export/Report buttons and in the `reportName` argument for `QueryReport` actions. For information on the syntax.

The following matrix shows which export options are available (Y) for the various export types:

	Fat Client Compatible	Encoding	Emit Configuration	Emit Native Format	Emit Invariant Format	Emit Catalogs	Emit Conditions	Emit Links
List	FALSE	Y	Y	Y	Y	Y	Y	Y
Details	FALSE	Y	Y	Y	Y	Y	N	Y
Tree	FALSE	Y	Y	N	N	N	N	N
List	TRUE	Y	Y	Y	Y	Y	Y	Y
Details	TRUE	N	N	N	N	N	N	N
Tree	TRUE	N	N	N	N	N	N	N

**Note:** Not every combination of the options listed here is actually meaningful, e.g. exporting catalogs for a tree view does not make sense, because a tree export contains table captions only and no catalog fields.

### Export.ContentDisposition

`inline` ... the browser tries to display the report results in the current browser window

`attachment` (default) ... the browser asks the user to save the report results or open them in a separate window.

### Export.EmitRepNames

Determines what is exported from rep ID fields in queries and lists:

`true` (default) ... rep names are exported.

`false` ... rep IDs are exported.

## Export.Encoding

`UTF-8` ... the export is encoded using UTF-8.

`UTF-16` ... the export is encoded using UTF-16.

## Export.Xml.EmitCatalogs

`true` ... the XML elements `<Catalogues/>` and `<Languages />` are included in the XML export. `<Catalogues>` contains all values of all catalogs used in the exported data. The catalog values are included in two languages: the user's language and the catalog base language. The core catalog IDs are also output.

Information about the fixed catalogs used in a report is output to an XML node analogous to variable catalogs (with the difference that values of fixed catalogs are output only in the current language).

`false` (default) ... no catalogs are included with the XML export.

## Export.Xml.EmitConditions

`true` ... when exporting query results, this option exports the conditions (=filters) used when executing the query. The filters are output to the XML element `<Conditions/>`. Variable conditions like `$curDay`, `$curRep` etc. are replaced with their actual value.

`false` (default) ... no conditions are included with the XML export.

## Export.Xml.EmitConfiguration

`true` ... the XML element `<Configuration/>` containing the configuration (e.g. of the field group or the tree view) is included in the XML export. This allows you to format a report's output e.g. based on field attributes.

`false` (default) ... no configuration data is included in the XML export.

## Export.Xml.EmitBaseFieldIds

`true` ... an attribute `fidcore` (short notation: `fic`) containing the base field number is output for the `<row/>` and `<fields/>` tags. The base catalog IDs are output as well. The base info area shortcuts `tableshortcore` are also output to allow the creation of base reports.

## Export.Xml.EmitInvariantFormat

`true` ... all exported field values are formatted using Aurea CRM's internal culture invariant representation:

Field type	Formatting
Date	YYYYMMDD
Time	HHMM

Time (with seconds)	HHMMSS
Time (with milliseconds)	HHMMSSTTT
Boolean	1 or 0
Other field types	No special formatting is applied.

`false` (default) ... all exported field values are formatted according to the users locale settings stored as Web Configuration parameters in the users configuration. For further details, see [Locale Parameters](#) on page 125.

### Export.Xml.EmitLineBreaks

`true` (default)...Line breaks are exported.

`false`...Line breaks are removed from the XML output.

### Export.Xml.EmitLinks

`true` ... for every record, a hyperlink and a title for this link is included in the XML export. The titles are formatted using the default table caption of the record's info area, e.g.

```
<field_link tableshort="FI">http://.../ </field_link> <field_ref tableshort="FI">Aurea Software</field_ref>
```

`false` (default)... no hyperlinks are included in the XML export.

### Export.Xml.EmitLocaleSettings

`true` (default) ... the XML element `<LocaleSettings/>` containing the users locale settings is included in the XML export.

`false` ... no locale setting data is included with the XML export.

### Export.Xml.EmitNativeFormat

`true` ... all exported field values are formatted using an XML-compliant format:

Field type	Formatting
Date	YYYY-MM-DD
Time	HH:MM
Time (with seconds)	HH:MM:SS
Time (with milliseconds)	HH:MM:SS.TTT

Boolean	t or f
Other field types	No special formatting is applied.

`false` (default) ... all exported field values are formatted according to the users locale settings (stored as Web Configuration parameters in the user's configuration). For further details, see [Locale Parameters](#) on page 125.

### Export.Xml.EmitOnlyBaseFields

`true` ... only fields having a base ID are included in the XML output. The base info area shortcuts `tableshortcore` are also output to allow the creation of base reports.

### Export.Xml.EmitVariables

`true` ... the XML element `<Variables/>` is included in the XML export. `<Variables>` contains all the variables that are valid for the currently logged-in user.

`false` (default) ... no variables are included in the XML export.

### Export.Xml.ExportPhysicalInfoArealds

Relevant when using virtual info areas. For further details, see [Virtual Info Areas](#) on page 275.

`true` ... the exported XML always contains the physical info area ID.

`false` (default) ... the exported XML contains the actual info area ID, i.e. the virtual info area ID, if applicable.

Example:

```
<field tableshort="VDOC" tablename="Doctor" fieldno="2" fieldname="Company"
...><field tableshort="FI" tablename="Company" fieldno="2" fieldname="Company"
...>
```

### Export.Xml.FatClientCompatible

`true` ... XML Export uses a Aurea CRM win-compatible schema.

`false` (default) ... XML Export uses the Aurea CRM.Web schema.

### Export.Xml.NamesOption

`Short` ... the XML output contains shorter XML attribute and tag names. If your reports are very large, this options helps to reduce the file size (but has the disadvantage that the XML is much less readable).

Default Name	Short Name	Default Name	Short Name	DefaultName	Short Name
catalog	c	fieldno	fn	recid	ri

catalogs	cs	fieldplaceholder	fph	sortnr	sn
catNo	cn	fields	fs	table	t
catNoCore	cnc	fieldtype	ft	tableCaption	tc
catval	cv	field_link	fl	tablelabel	tlb
extKey	e	field_ref	fr	tablename	tn
fid	fid	fixedCatalogues	fcs	tables	ts
fidcore	fic	langid	li	tableshort	tsr
field	f	mnr	mnr	tableshort-core	tsrc
fieldlabel	flb	name	nm	type	ty
fieldlength	fl	parCatNo	pcn	value	v
fieldmandatory	fmt	parCatVal	pcv	Variable	var
fieldname	fnm	prefix	p	Variables	vars

Every other value than `short` results in the default names being used.

## XSL Transformations

Learn about XSL transformation for CRM.Web.

XSL transformation scripts and document functions should be enabled for CRM.Web to be able to use them. Set the `EnableScript` and `EnableDocumentFunction` options to true, under the XSLTransformation section in the `settings.xml` file. See the sample below:

```
<update.lib>
  <XslTransformation>
    <EnableScript>FALSE</EnableScript>
    <EnableDocumentFunction>FALSE</EnableDocumentFunction>
  </XslTransformation>
</update.lib>
```

---

**Attention:** Enabling XSL transformations exposes a severe security threat, potentially revealing the remote code execution path to an attacker. You should

ensure that people with potential malicious intent do not have access to the system. if you enable these options.

By default, the XSL transformations used by Aurea CRM.Web are stored in `..\data\reports\xslt`.

To use another than the default location, add the following elements to the `settings.xml` file:

```
<update.web>
  ...
  <mmReports>
    <XSLTPath>Data/Reports/XSLT</XSLTPath>
  </mmReports>
```

By default, the XSL transformations are loaded on first use and cached thereafter. While developing new XSL transformations this can become cumbersome, because Aurea CRM.Web has to be restarted for each change to the XSL transformation. For information on disabling the caching of XSL transformations and debugging reports, see the article “How to Debug Reports” at <https://support.aurea.com>.

There are no constraints on the XSL transformations you create for XSL reporting, as long as they are valid transformations according to the W3C XSL Transformations (XSLT) Version 1.0 recommendation located at <http://www.w3.org/TR/xslt>.

However, you should add a `media-type` attribute to your `<xsl:output>` element.

Aurea CRM.Web looks for this attribute and modifies the content-type of the HTTP-Stream sent to the client (browser). Without setting the media-type the OS on the client computer cannot determine the correct application for opening the report.

```
<xsl:output method="xml" encoding="UTF-8" media-type="application/msword" />
```

**Note:** The media-type is also used to alter the file extension of the file name transferred to the client. For security reasons the client application (browser) uses the file extension rather than the HTTP content-type to determine the application used to open a file. As a result all content-types and file extensions must be known to your Aurea CRM.Web server, i.e. defined in the registry.

## Creating Aurea CRM win Reports in Aurea CRM.Web

This topic explains how to create reports on Aurea CRM Win.

In Aurea CRM.Web you can create reports based on report formats defined in Aurea CRM win by using the `ExecuteReport` action. For further details, see [ExecuteReport](#) on page 423.

For information on Aurea CRM win reports, see [Defining Report Formats](#) in the *Aurea CRM win User Manual*.

The corresponding XML is generated by the core, but all transformations with style sheets are done by Aurea CRM.Web. Therefore it is necessary that all required files (xslt, css and images) are available in the respective directories on the web server (i.e. you need to manually copy the required files to these locations):

```
..\data\reports\crm\xslt
..\data\reports\crm\xslt\images
```

The report format to be used is specified by the `name` parameter of the `ExecuteReport` action. The `uid` parameter can be used to pass the uid of a linked record for the report. The `parameters` parameter allows you to specify additional settings (as JSON):

- XSL transformations: Enter `{XslTransformation: "<transformation name>"}` to apply an additional transformation at the end (i.e. after a transformation defined in the format was performed).
- target and contentDisposition settings, e.g. `{target: "CreateNewTab"}` or `{target: 'myIFrame', contentDisposition: 'inline'}`

For more information on these parameters, see [targetFrame](#) and [contentDisposition](#).

---

**Note:** The `parameters` parameter does not support any other export options like `Export.Encoding`, `Export.Xml.EmitConditions` etc. You need to set them directly in the format definition in Aurea CRM win.

---

- The parameters listed below as defaults, e.g. `{stylesheet: " myreport.css "}`

By default, the following parameters are passed from Aurea CRM.Web to the XSL:

- `style`: name of the style to be used, e.g. "update"
- `stylesheet`: file name or path (URL or UNC) to the CSS file, e.g. " update.css " or "http://myweb.com/styles/custom.theme/report.css"

---

**Note:** The CSS file must be located as defined in the XSLT file. Aurea CRM.Web uses `../styles/update/report.css` as default.

---

- `imagepath`: relative or full path (URL or UNC) to Aurea CRM.Web's images, e.g. " ../images/update/ " or "http://myweb.com/images/update/"
- `imagefolder`: full path to the `..\xslt\images` folder, e.g. " c:\my\_web\_installation\web\data\reports\crm\xslt\images "

---

**Note:** These defaults cannot be modified directly in the XSL, but can be overridden by the format definition or by parameters of the action definition, e.g. by `{stylesheet: "<style sheet name>"}`.

---



---

**Note:** In order to be able to override these defaults via the action definition (`parameters` parameter), the corresponding parameters need to be defined as `param` in the XSLT file, i.e. you cannot use e.g. `imagefolder` if the XSLT only contains `imagepath`.

---

**Note:** Settings provided via the action definition always take precedence over parameters defined in the XSL or in the format definition.

---

If the report format has the **Output Link** option checked, make sure you have configured the **web URL** entry in the **Configuration** info area (MC, category **Link**). For further details, see [Configuration Info Area](#) in the *CRM.Core Administrator Guide*.

The following **restrictions** apply to Aurea CRM win reports executed in Aurea CRM.Web:

- The XSLT files must be located in `..\data\reports\crm\xslt`.
- If the `XslTransformation` parameter is set in the action definition, the provided transformation is always applied last.
- The placeholders `$userdir` and `$installDir` cannot be used.
- You cannot use variable conditions in the transfer field format.
- The **Save as file** option is not supported: Reports in Aurea CRM.Web are always executed synchronously and then displayed.
- The **Save first/second transformation to file** options are not supported.
- The **Transfer to spreadsheet** option is not supported.
- The creation of PDF reports as described in the *Aurea CRM win User manual* (chapter **PDF Reports**) is not supported.

Use the `Export.WinReportAlwaysInMemory` and `Export.WinReportMemoryThreshold` Web Configuration parameters to control whether a report is generated as a file (in `web.data\session\report.xml`) or only in the memory:

If `Export.WinReportAlwaysInMemory` is checked, the report is created in-memory (instead of as a file).


`Export.WinReportMemoryThreshold` defines the max. file size (in KB) used for in-memory creation (default: 256 KB)

If `Export.WinReportAlwaysInMemory` is checked and the defined `Export.WinReportMemoryThreshold` is exceeded, the report creation is aborted with an error message.

## Customer Admin Mode

Learn about accessing customer admin mode on web.

This feature is primarily intended for customer administrators in an SaaS environment but can also be used e.g. for country admins in large in-house projects.

You can set up a special "customer admin mode" enabling (administrative) users to access CRM.Designer from Aurea CRM.Web and customize field controls using the  (**Adjust Layout & Fields**) button.

To set up customer admin mode:

1. Enable **authentication** for CRM.Designer, i.e. disable anonymous access. For further details, see [Settings](#) on page 750.

All users with the \*SelfAdmin configuration (s. below) in their configuration hierarchy are able to access CRM.Designer. The respective CRM.Designer logins are automatically created (s. below).

2. Set up the **connection between Aurea CRM.Web and CRM.Designer**.

---

**Note:** Aurea CRM.Web and CRM.Designer need to run on the same server.

---

3. Create the required **configuration hierarchy** with a \*SelfAdmin and \*SelfAdminDev configuration between a common parent and the users. For further details, see [Creating SelfAdmin Configurations](#) on page 868. This ensures that configurations delivered and maintained by update (or a partner) can not be changed by customer admins.

The customer admin customizes field controls in the \*SelfAdminDev configuration and then publishes the finalized customization to the \*SelfAdmin configuration, thus making it available to all users.

Other CRM.Designer pages and other configurations are not accessible.

4. Define a dedicated **customer admin role** and assign it to the \*SelfAdminDev configuration and to the users that should perform the customizations.


Customer admins must log on to Aurea CRM.Web with this role.

5. Configure the EditDetailsControl **button** in the desired `Expand` headers. For further details, see [EditDetailsControl](#) on page 871.





---


**Note:** The `EditDetailsControl` button is available for the SU and users who have access to the CRM process 1.5.1.3.106 "Edit Mask" (CRM process ID M\_173). This CRM process is enabled by default. Make sure you disable this CRM process for all roles except the customer admin's role.

---

If a user who has the \*SelfAdmin configuration in his configuration hierarchy logs on to Aurea CRM.Web with the customer admin role and clicks on  (**Adjust Layout & Fields**) the following happens:

- CRM.Designer is opened in a new browser window.
- When logging on for the first time a CRM.Designer login "crm:<update.CRM login name>" is automatically created:

CRM.designer Logins						
✓ Save						
	Name	Password	Configuration	Configuration Mode	Vertical	User Group
	Admin1	yes	customized_parent_config	this configuration	BB	User
	Admin2	yes	customized_parent_config	with children	all	Power User
	SU	yes	all	with children	all	Administrator
	crm:SELF	yes	customized_parent_config_SelfAdminDev	this configuration	BB	Customer

The login is created in **User Group** "Customer" and for the \*SelfAdminDev **Configuration** only. This login is then automatically used every time the user clicks on  (**Adjust Layout & Fields**).

- Changes to field controls are saved in the \*SelfAdminDev configuration.
- The administrative user can publish these changes to the \*SelfAdmin configuration, thus making it available to all users.

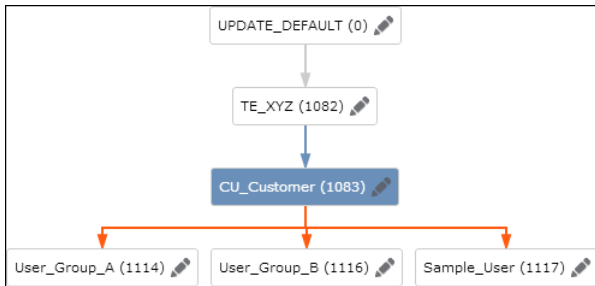
**Note:** The customer admin mode feature is intended to be used only in the productive environment. There is no (automatic) transport back to a development (or testing) environment. You would have to use CRM.bulkloader to transport the \*SelfAdmin configuration to your development environment.

**Note:** Use the **Compare Field Controls** page in CRM.Designer to compare changes made by a customer with a parent configuration as provided by update or a partner to determine possible additions for the next release of your parent configuration. For further details, see [Compare Field Controls](#) on page 746.

**Note:** No automatic merge of field control definitions is supported.

## Creating SelfAdmin Configurations

This topic explains creation of SelfAdmin configurations.



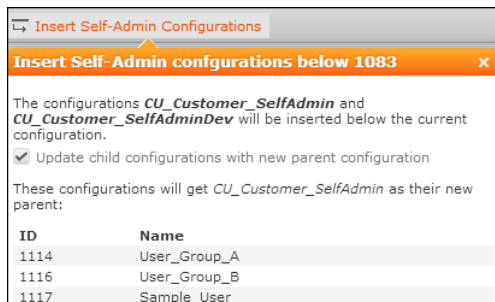
For customer admin mode to work you need to insert "SelfAdmin" configurations between the lowest configuration maintained and deployed by Aurea or partners and the customer's user configurations. This SelfAdmin configuration can then be maintained by the customer; changes in this configuration apply to all users.

**Note:** It is strongly recommended to back up the CRM.Designer database before creating Selfadmin configurations.

**Note:** The SelfAdmin configurations are identified by their suffixes: \*\_SelfAdmin and \*\_SelfAdminDev. Do not use these names in any other context.

To create/adapt the required configuration hierarchy:

1. Switch to the configuration below which you want to insert the SelfAdmin configuration:
  - a. Click **Users | Configurations**.
  - b. Select **View > Configurations**, see [Configurations View](#) on page 83.
  - c. Beside the desired configuration (CU\_Customer in this example) click on **Edit**.
2. Click on **Insert Self-Admin Configurations**.

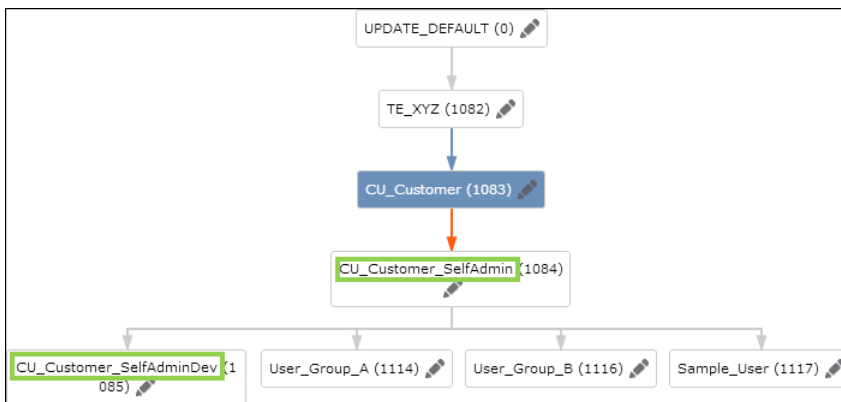


A popup displays which configurations is added and which child configurations is affected by the change. (Uncheck **Update child configurations with new parent configuration** if you do not want to change the configuration hierarchy for existing child configurations. In that case the child configurations does not see any changes done by the customer admin.)

**Note:** If the IDs required for inserting the new configurations are not available, a message is displayed asking you to click on **Change ID** instead to free up the required IDs, see [Changing Configuration IDs](#) on page 85.

### 3. Click on **OK**.

Two new configurations are inserted according to your settings: <parent\_config>\_SelfAdmin and <parent\_config>\_SelfAdminDev.



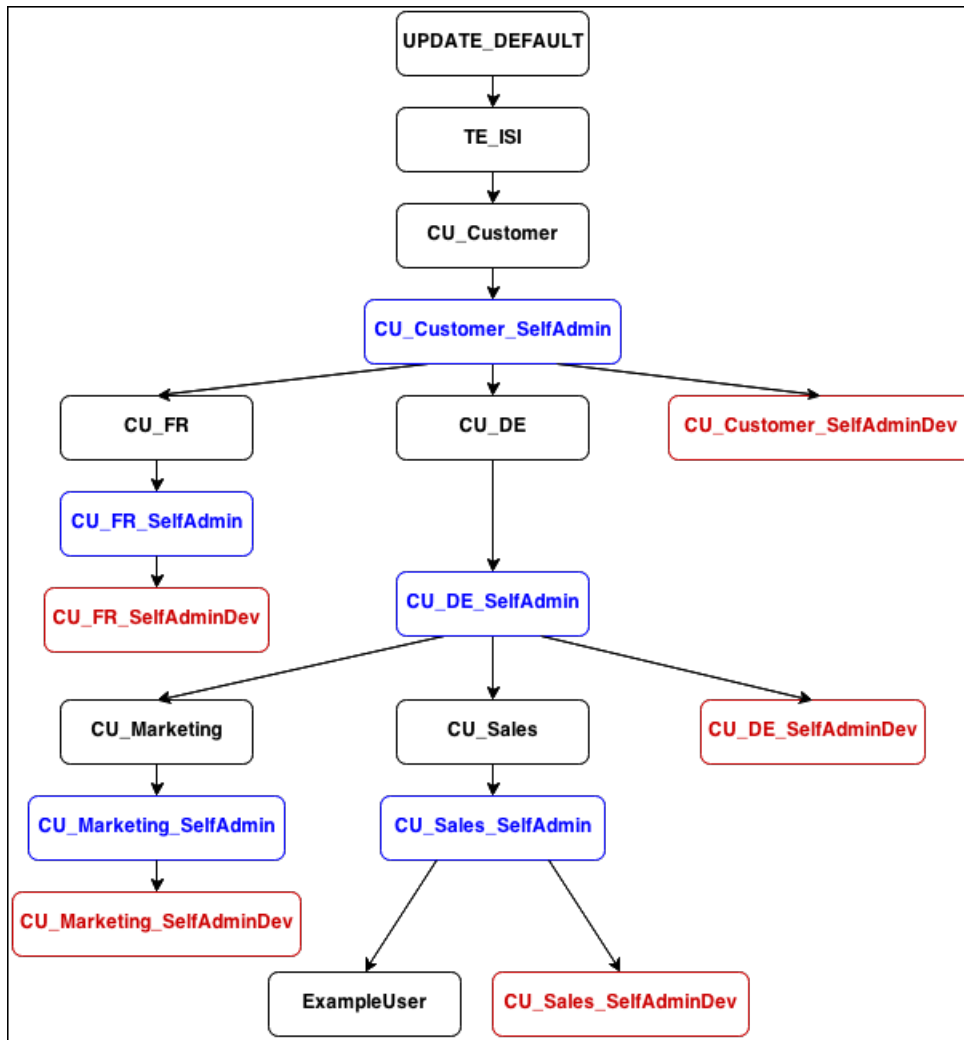
The **\*SelfAdmin** configuration is the new parent of your user (and user group) configurations.

Assign the **\*SelfAdmin** configuration to the DefaultUser to make sure new Aurea CRM.Web users get this new parent configuration. For further details, see [Automatically Creating User Configurations](#) on page 82.

The **\*SelfAdminDev** configuration is the configuration that is edited by the customer admins when using the `EditDetailsControl` button.

Assign the customer admin role to the **\*SelfAdminDev** configuration. For further details, see [Defining and Assigning a Customer Admin Role](#) on page 870

## Multiple SelfAdmin Configurations



In such a scenario the customer admin needs to log on to Aurea CRM.Web with e.g. the "Marketing" role **and** the "SelfAdmin" role to access the CU\_Marketing\_SelfAdminDev configuration.

## Defining and Assigning a Customer Admin Role

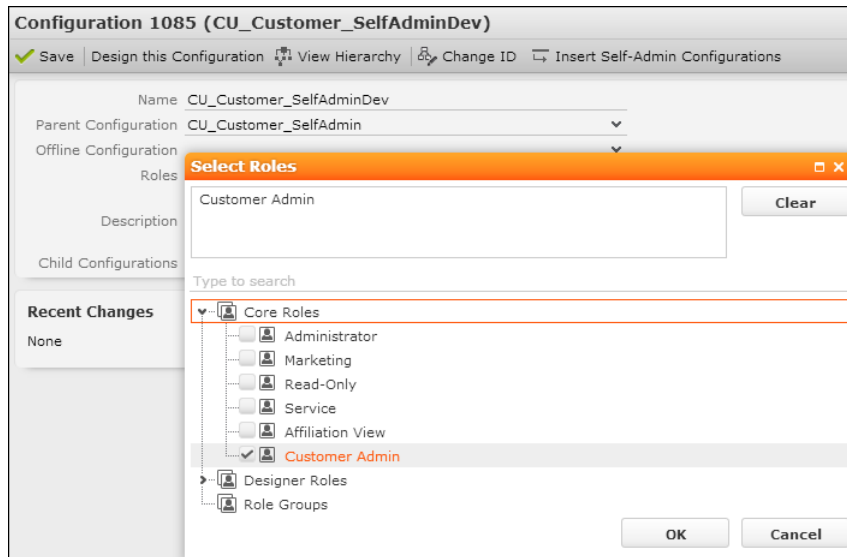
Learn to configure Customer admins and assign role to them.

Customer admin mode requires a dedicated customer admin role:

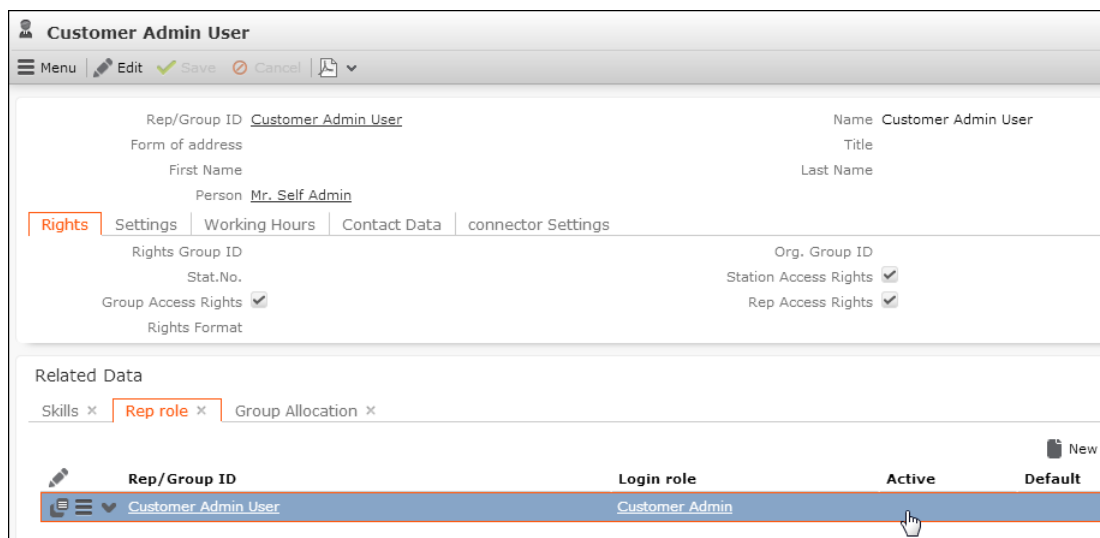
1. Define a login role, e.g. "Customer Admin", see [Login Roles](#) in the *CRM.Core Administrator Guide*.

Make sure the CRM process 1.5.1.3.106 "Edit Mask" (CRM process ID M\_173) is enabled in this role's CRM process configuration. This CRM process determines the availability of the `EditDetailsControl` button.

2. Assign the role to the \*SelfAdminDev configuration, see [Editing Configurations](#) on page 84 and [Dynamic Configuration Assignment Based on Roles](#).



- Assign the role to the rep records of those users that should perform the customizations, see [Role-Based Access](#).



If a user logs on to Aurea CRM.Web with this role, the \*SelfAdminDev configuration is loaded as his parent configuration. If he logs on without the role the \*SelfAdmin configuration is used.

## EditDetailsControl

Learn to customize fields control using EditDetailsControl button.

The `EditDetailsControl` button  (**Adjust Layout & Fields**) allows (administrative) users to customize field controls.

**Note:** The `EditDetailsControl` button is available for the SU and users who have access to the CRM process 1.5.1.3.106 "Edit Mask" (CRM process ID M\_173). This

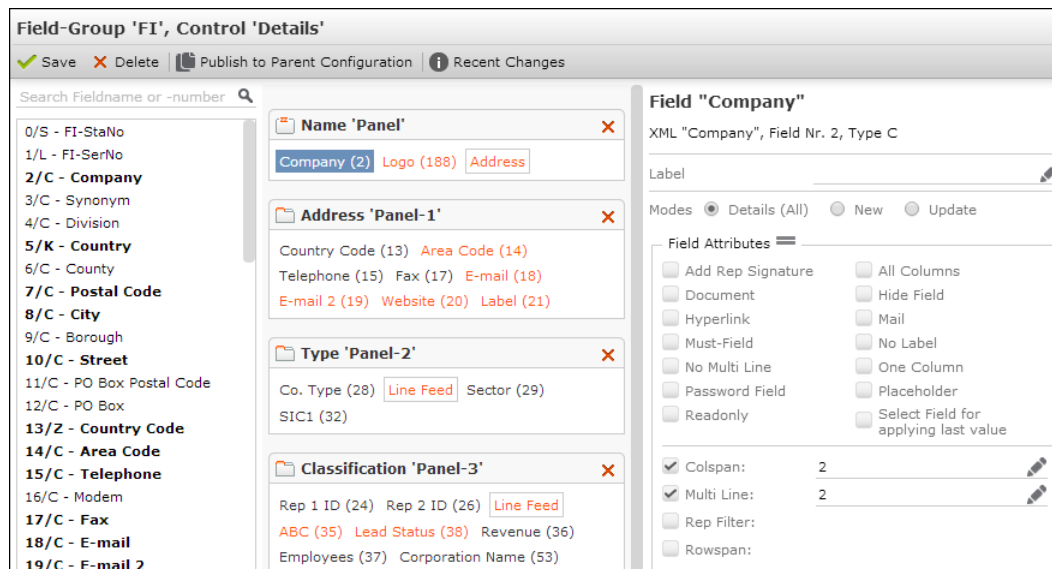
CRM process is enabled by default. Make sure you disable this CRM process for all roles except the customer admin's role.

You can configure the `EditDetailsControl` button into `Expand` headers.

Clicking on the `EditDetailsControl` button opens CRM.Designer in a new browser window displaying the Details control of the current info area. Users can customize (copy/edit/delete) the Details control. In addition, they can access all available field groups (via the start page) and thus customize field controls for other info areas as well (Details, Search, List, MiniDetails controls). Other CRM.Designer pages are not accessible.

**Note:** If Aurea CRM.Web was started in debug mode, clicking the `EditDetailsControl` button starts CRM.Designer in debug mode as well.

## Field Control Editor



The Field Control Editor displays only a sub-set of the available attributes and settings that require no or little training and no knowledge of JSON and JavaScript. Hidden attributes are retained if a field is removed from a control and re-inserted at later time.

## Publish to Parent

Changes are immediately displayed in Aurea CRM.Web after saving, i.e. it is not necessary to invalidate the cache. Changes are saved to the `*SelfAdminDev` configuration. Clicking on **Publish to Parent Configuration** copies the changed configuration units from the `*SelfAdminDev` configuration to the `*SelfAdmin` configuration; this makes changed field control(s) available for all users having the `*SelfAdmin` configuration in their configuration hierarchy. For further details, see [Creating SelfAdmin Configurations](#) on page 868.



---

**Note:** The **Publish to Parent Configuration** button is only available for \*SelfAdminDev configurations.

---

## Texts and Text Groups

When defining new labels (for fields or panels) the texts are saved in a `SelfAdmin_FieldControl` text group that is automatically created as soon as the first text is defined. The text group is automatically **moved** to the \*SelfAdmin configuration when clicking on **Publish to Parent Configuration**.

Users can select texts from existing text groups but can not change existing texts or define new texts for any text group other than the `SelfAdmin_FieldControl` text group.

---

**Note:** The `SelfAdmin_FieldControl` text group must not be used by any other than the \*SelfAdminDev configuration.

---

## Deleting Field Controls

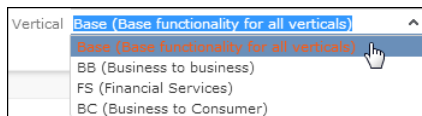
If a field control is deleted while in customer admin mode, a Remote Deletion entry is automatically created. For further details, see [Remote Deletion](#).

# Base Customizing

This feature is only relevant for Aurea software and for partners who want to create a designer Base configuration for multiple verticals.

The goal of this feature is that processes of a similar type involving "base" info areas should only need to be configured once when developing a template. These configurations can then be distributed using the individual vertical templates where they can be integrated in the specific configuration for the target industry. Defining Base `Infoarea` designer units is, of course, only possible for info areas, fields and catalogs that are common to all (or at least two) verticals. In addition, designer units independent of info areas (like some menu actions, forms etc.) can also be customized as "base".


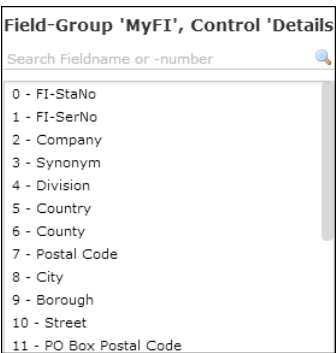
See the *CRM.Core Administrator Guide* for details concerning the [data model](#) (e.g. which info areas, fields, and other designer units are "base") and customization in Aurea CRM win.



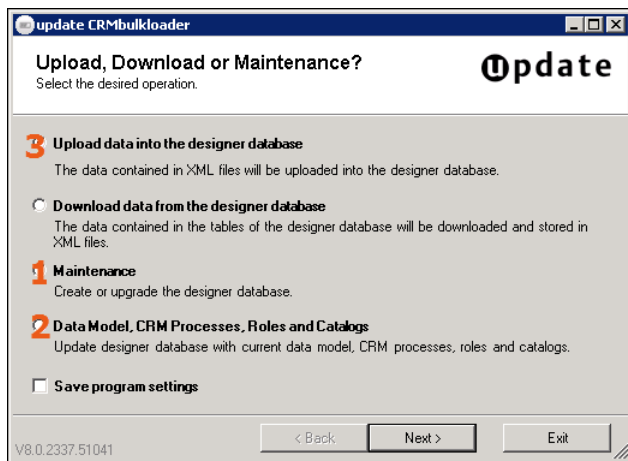
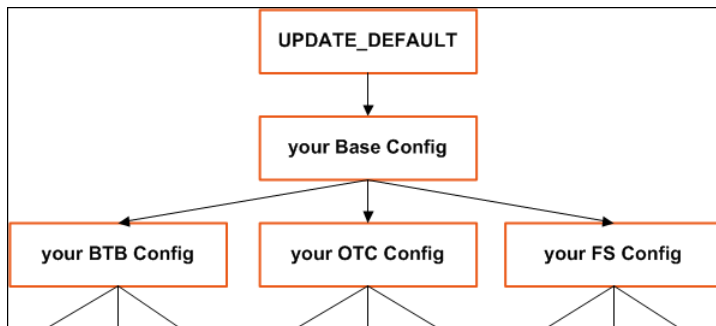
The Base vertical in CRM.Designer contains a data model that consists of "base" info areas and their fields, vertical info areas if they have "base" fields only these fields are visible, and "base" catalogs.

**Note:** Do not mix base and vertical in one designer!

In the Base vertical, "base" info area IDs, field IDs and catalog numbers are displayed; in the BB/FS/BC verticals you still see the vertical-specific IDs. In the example for the Person (KP) info area below, note the different field IDs for Base and BTB, as well as the fact that the "Function" catalog is only visible for the BTB vertical since it is not a "base" catalog.

Infoarea FI – vertical Base	Infoarea FI – vertical BTB
 <p>Field-Group 'FI', Control 'Details'</p> <p>Search Fieldname or -number</p> <ul style="list-style-type: none"> <li>1000 - FI-StaNo</li> <li>1001 - FI-SerNo</li> <li>1002 - Company</li> <li>1003 - Synonym</li> <li>1004 - Division</li> <li>1005 - Country</li> <li>1006 - County</li> <li>1007 - Postal Code</li> <li>1008 - City</li> <li>1009 - Borough</li> <li>1010 - Street</li> <li>1011 - PO Box Postal Code</li> </ul>	 <p>Field-Group 'MyFI', Control 'Details'</p> <p>Search Fieldname or -number</p> <ul style="list-style-type: none"> <li>0 - FI-StaNo</li> <li>1 - FI-SerNo</li> <li>2 - Company</li> <li>3 - Synonym</li> <li>4 - Division</li> <li>5 - Country</li> <li>6 - County</li> <li>7 - Postal Code</li> <li>8 - City</li> <li>9 - Borough</li> <li>10 - Street</li> <li>11 - PO Box Postal Code</li> </ul>

When using Base customizing, update suggests the following designer configuration structure:



---

**Note:** To be able to see the Base vertical in CRM.Designer, the `<EnableVerticalBase />` entry must be active (i.e. not commented out) in the `settings.xml` file of CRM.Designer.

---

## Extending Field/Header Groups Defined in Base

Learn to extend header groups configured in base.

Header groups are used to organize different types of headers - predefined, standard headers (e.g. Search, Expand, Tree) and custom special headers. When a header is referenced, a header group is configured in which it should be searched for. First, a suitable group is found in the hierarchy of the designer configurations and verticals. Then, a header with the matching name / type is searched for in that group.

In Aurea.CRM a new feature was implemented that allows an on-demand extension of header groups in child configurations.

This has been possible before in the same vertical, it is now even possible to switch from Base to a different vertical. Below is the detailed explanation of this logic:

- If a header group A is defined in the Base vertical of some configuration X, it is not possible to extend it in a specific vertical. One can make a copy of it in a specific vertical, however this copy is fully independent from the Base's A and won't inherit the headers. There are technical reasons for it in the way update.CRM web handles configurations and verticals - once a header group is found in a particular vertical, Base is no longer looked at, so one can not extend it horizontally.
- In a sub-configuration Y of X, in the Base vertical, one can add/override headers to the header group A, thereby logically extending it. These headers are still a part of the original header group A but is only visible in the configuration Y.
- Additionally, one can add/override headers to the header group A in the sub-configuration Y in another (e.g. BB) vertical. As soon as he does that, an extension of the group A is created in the BB vertical. An extension is a header group with a special flag (called `extendsparent` in the DB table) that can be used to define vertical-specific headers while still inheriting those from the X's Base. As a result, in the vertical BB of Y, headers defined both in X's Base and Y's Base is available in addition to those created specifically for Y's BB. This is also reflected in the copy operation - if one decides to copy the header group into Y's BB, all the inherited headers is merged and cloned so that the copy can be worked with independently. Such a copy no longer inherits anything.
- An important property of the extension is that it extends the parent configuration (although it is used for extending a header group in Base), so if we were to create a copy of A in the X's BB, this takes precedence and the extension created in Y's BB only inherits headers from that one. In general, an extension always refers to a header group found in the nearest parent configuration, where first the current and then the Base vertical is searched through. Similarly, to the on-demand

extension done when the first header is created that requires the extension, the extension is deleted automatically as soon as all the headers it contained (except for those that are inherited, of course) are deleted. Thus, the extension only lives until it becomes superfluous.

## Virtual Info Areas & Core Customizing

This topic concerns the development of product templates and deriving multiple solutions from a single Base template.

Virtual Info Areas & Core Customizing is solely directed at internal update users and partners.

By default virtual info areas inherit the properties concerning Base info areas and fields from their physical info area. This may lead to problems e.g. if multiple verticals use FI to create a virtual info area called "VFI1". All these virtual info areas would be defined as "Base", because their physical reference (FI) is Base. Therefore the Base configuration would contain multiple VFI1 info areas.

To prevent that scenario, set the `coreMode` attribute to "vertical" (or "false") in the `settings.xml`. This defines the virtual info area and its fields as vertical-specific independent of the Base info area.

```
<VirtualSchema>
    <InfoArea infoAreaId='VHOS' physicalInfoAreaId='FI' fieldId='28'
fieldType='K' value='Spital' name='Hospital/Clinic' coreMode='base' />
    <InfoArea infoAreaId='VDOC' physicalInfoAreaId='FI' fieldId='28'
fieldType='K' value='Arzt' name='Doctor' coreMode='vertical' />
    <InfoArea infoAreaId='VPHA' physicalInfoAreaId='FI' fieldId='28'
fieldType='K' value='Apotheke' name='Pharmacy' coreMode='vertical' />
</VirtualSchema>
```

---

**Note:** If a virtual info area has been defined as "Base", you cannot revert it to "vertical" at any later time.

---

# A

## Data Types in Aurea CRM

---

In some dialog boxes, the field type is also displayed along with the field name and number.

The following data types are available:

- B = Boolean (logical)
- C = Character string
- D = Date
- F = Floating point number
- K = Variable catalog
- L = Long integer
- S = Short integer
- T = Time
- X = Fixed catalog
- Z = Decoded field. Values in these fields are automatically entered by the business logic. Z-fields do not exist in the database. (Examples: reference fields, fields that contain links to other records, rep fields, rep group fields, 2nd currency fields, external match-up fields etc.).

Z-fields cannot be added as search fields or used for defining filters and queries.

The content of a Z-field is not updated in lists if the target record is updated by another process. Workaround: Instead of the Z-field add a linked field of the target record to the list. Example: Instead of using the **Company** field (MA:76), add the **Company** field from FI as linked field to the MA List control.

# B

## Effectiveness of Rights and Configuration Changes

For changes made using the Aurea CRM win **Rights** or **Maintenance** modules to take effect, it depends on the type of change whether it is applied immediately, after the refresh cycle (i.e. after the time when rights, catalog values etc. cached for a certain time period are read again from the Aurea CRM database), after reloading Aurea CRM.Web (F5) or after recycling the application pool.

The refresh cycle is determined by the **Cat. refresh frq.** and **Format refresh frq.** settings in the **Station Configuration** (T0) info area (default: 30 minutes), >> [Station](#) on page 32 on page 13.

The following list gives an overview when which changes are effective:

Group	Action	Change effective for currently logged-in user	Change effective for newly logged-in user
Maintenance	Data model changes	application pool recycling	application pool recycling
Rights	Station right NEW	refresh cycle expired	refresh cycle expired *
Rights	Changes to station right	refresh cycle expired	refresh cycle expired *
Rights	Remove station right	refresh cycle expired	refresh cycle expired *
Rights	Group right NEW	refresh cycle expired	refresh cycle expired *
Rights	Changes to group right	refresh cycle expired	refresh cycle expired *
Rights	User right NEW	refresh cycle expired	refresh cycle expired *
Rights	Changes to user right	refresh cycle expired	refresh cycle expired *
Rights	Remove user right	refresh cycle expired	refresh cycle expired *

<b>Group</b>	<b>Action</b>	<b>Change effective for currently logged-in user</b>	<b>Change effective for newly logged-in user</b>
Rights	Changes to default value triggers	refresh cycle expired	refresh cycle expired *
Rights	Conditional field rights	refresh cycle expired	refresh cycle expired *
Roles	Assign to/remove for a user	refresh cycle expired + logout/login	immediately
Roles	Add, update, delete a role	every 60 sec.	immediately
Catalogs	Changes to catalog values	refresh cycle expired + logout/login or F5	refresh cycle expired
Catalogs	Creation of new catalog values	refresh cycle expired + logout/login or F5	immediately
Catalogs	Changes to fixed catalog values	application pool recycling	application pool recycling
Catalogs	Creation of new fixed catalog values	application pool recycling	application pool recycling
Rep	New rep	refresh cycle expired + logout/login or F5	refresh cycle expired
Password Policy	Minimum password length	application pool recycling	application pool recycling
Password Policy	Password must contain numbers, letters	application pool recycling	application pool recycling
System Lock	Set system lock	refresh cycle expired + logout/login or F5	immediately
System Lock	Remove system lock	refresh cycle expired	refresh cycle expired
Tenants	Assign a company to a new tenant	immediately	immediately
Tenants	Assign a user to a new tenant	application pool recycling	application pool recycling
Logging	Activate web trace	refresh cycle expired	refresh cycle expired
Logging	Deactivate web trace	refresh cycle expired	refresh cycle expired
Logging	Change log level in settings.xml	application pool recycling	application pool recycling

Group	Action	Change effective for currently logged-in user	Change effective for newly logged-in user
Tracking	Activate session tracking	refresh cycle expired + logout/login or F5	refresh cycle expired
Tracking	Deactivate session tracking	refresh cycle expired + logout/login or F5	refresh cycle expired
Protocol	Activate write protocol	application pool recycling	application pool recycling
Protocol	Deactivate write protocol	application pool recycling	application pool recycling
Protocol	Change write protocol	application pool recycling	application pool recycling
Configuration (MC)	Change configuration entries	refresh cycle expired	refresh cycle expired

**Note:** For the rows marked with \* the following applies:

- If e.g. "Deny Update" is set and the user logs on after the refresh cycle, the fields are inactive in Expand view. If the user was already logged on, he can still modify these fields, but when he tries to save the record a corresponding message is displayed.
- If a rights definition is removed which resulted in fields being displayed as inactive, the user must log off and log on again after the refresh cycle (otherwise the fields remain inactive).



# C

## AQL – Aurea Query Language

---

Similar to SQL which queries relational databases for records, AQL can be used to query the Aurea CRM database.

Use the Query Editor in Aurea CRM.Web to define and test your AQL statements:

The screenshot shows the Aurea CRM Query Editor interface. At the top, there are action buttons: Save (green checkmark), Delete (red X), Send to Reps... (person icon), Run Query in Background (double arrow), and Run Query (single arrow). Below these is the 'Query name' field containing 'MyQuery'. The main area is divided into two sections: 'UQL Syntax' and 'Formatted'. The 'UQL Syntax' section shows the following AQL query: 

```
select (ProblemDescription, FI.Company, KP1.LastName, KP2.LastName) from (KM)
where (Priority>"#1")
with (FI)
where (Revenue>"100000")
plus (KP using link 1 as KP1)
plus (KP using link 2 as KP2)
```

 The 'Formatted' section shows a visual representation of the query with expandable sections for 'Ticket', 'Priority > "High"', 'WITH Company', 'Revenue > "100000"', 'PLUS Person in Company (1. FI\_StaNo,FI\_SerNo,PE\_StaNo,PE\_SerNo -> FI\_1\_StaNo,FI\_1\_SerNo,KP\_1\_StaNo,KP\_1\_SerNo)', and 'PLUS Person in Company (2. FI\_StaNo,FI\_SerNo,PE\_StaNo,PE\_SerNo -> FI\_2\_StaNo,FI\_2\_SerNo,KP\_2\_StaNo,KP\_2\_SerNo)'. Each section has an 'Output Fields' field and a pencil icon for editing.

**Important:** A field with an AQL statement containing multiple conditions and using the same operator AND is interpreted as OR. Similarly, a multiple condition statement using only OR is interpreted as AND. This is the default behaviour of AQL in the context of a single field and applies only for a single field having multiple conditions defined with the same operator. The AND and OR operators behave as expected in all other cases.

For example, consider the following multiconditions on a fields and how they are interpreted:

1.  $F1 = 1 \text{ AND } F1 = 2 \text{ AND } F1 = 3$  was interpreted before as  $F1 = 1 \text{ OR } F1 = 2 \text{ OR } F1 = 3$   $\square$   $F1 \text{ IN } (1, 2, 3)$ .
2.  $F1 \neq 1 \text{ OR } F1 \neq 2 \text{ OR } F1 \neq 3$  was interpreted before as  $F1 \neq 1 \text{ AND } F1 \neq 2 \text{ AND } F1 \neq 3$   $\square$   $F1 \text{ NOT IN } (1, 2, 3)$ .

To configure the correct behaviour of the operators in the above context, set the option `Respect AND-OR condition operator during SQL IN-NOT IN conversion to true` in the Configuration (MC) info area. This will ensure that the AQL conditional operators AND and OR will behave as expected in the above single field context. If you enable this option, ensure that you change all the filter conditions already set based on the existing behavior.

For more information on setting the Configuration (MC) entry, see **Adding a Configuration Entry** in the *Aurea CRM.Core Administrator Guide*.

## AQL Syntax

Use the following syntax for AQL statements:

<code>&lt;AQLstatement&gt;</code>	<pre>SELECT [(&lt;outputfields&gt;)] FROM (&lt;from&gt;) [ WHERE (&lt;filter&gt;)] [ &lt;jointype&gt; (&lt;from&gt;) [WHERE (&lt;filter&gt;)]]* [ ORDERBY (&lt;sortfields&gt;)]</pre>
<code>&lt;outputfields&gt;</code>	<code>[&lt;alias&gt;.&lt;*&gt;   &lt;outputfield&gt;[,&lt;outputfield&gt;]*</code>
<code>&lt;outputfield&gt;</code>	<code>[&lt;alias&gt;.&lt;field&gt;</code>
<code>&lt;from&gt;</code>	<pre>[&lt;parentalias&gt;.&lt;infoareaid&gt; [AS &lt;alias&gt;] [LIMITED TO &lt;number&gt; ROWS] [FLAGS &lt;flags&gt;] [EXTENDED FLAGS &lt;extendedflags&gt;] [USING LINK &lt;linkid&gt;]</pre>
<code>&lt;filter&gt;</code>	<code>&lt;filterexpression&gt;[&lt;booleanoperator&gt; &lt;filterexpression&gt;]*</code>

<filterexpression>	<filterfield> <comparisonoperator> < filterfield >   < filterfield > <comparisonoperator> <filtervalue>   (<filterexpression>)
<filterfield>	F<fieldid>   <fieldname>
<filtervalue>	<quotedstring> (for more information see below)
<booleanoperator>	AND   OR
<comparisonoperator>	=   <   <=   <>   >=   >
<sortfields>	<sortfield>[,<sortfield>]
<sortfield>	[<alias>.] [<field>] [ <sortorder>] [ FLAGS <sortflags>]
<sortorder>	ascending descending
<jointype>	WITH   WITHOUT   PLUS   WITHSUMMED   PLUSSUMMED   WITHOPTIONAL   WITHOUTOPTIONAL   WITHSUMMEDOPTIONAL   HAVING   HAVINGOPTIONAL
<field>	<fieldname> <fieldid>
<fieldname>	<string> (the XML name)
<fieldid>	<number> (the unique field ID)
<infoareaid>	<string> (The info area ID in double quotes: "FI", "KP", etc.)
<alias>	<string> (for more information see below)
<parentalias>	<alias> (The alias of the parent info area, for joins only)
<linkid>	<number> (The link ID from the data model)
<quotedstring>	"<string>"

<flags>	<string>   <number> (for internal use ONLY)
<extendedflags>	<string>   <number> (for internal use ONLY)
<sortflags>	<string>   <number> (for internal use ONLY)

<alias> uniquely identifies an info area within the query. By default the alias and the info area ID ("FI", "KP", etc.) are the same. If an info area is used multiple times within a query, aliases can be defined using "AS <alias>" to keep those instances apart, >> [Example](#) on page 884 [Example](#) on page 519.

<filtervalue> has to be formatted as usual (hence with all other query APIs). Use the following prefixes:

- # for catalog codes, >> [Filtering by Catalog Values](#) on page 885 on page 519
- \$par for parameters, >> [Dynamic Filters](#) on page 509 on page 320
- \$cur for variable expressions, >> [Using Variable Expressions](#) on page 505 on page 318.

<sortflags> can currently not be saved in the CRM.Designer database.

---

**Note: Notes:** Line feeds in queries are for readability only, they have no semantic meaning.

---



---

**Note:** Keywords (SELECT, FROM, etc.) are case insensitive.

---

## Example

```
SELECT (KM.ProblemDescription,FI.Company,KP1.LastName,KP2.LastName)
FROM (KM) WHERE (Priority > "#1")
WITH (KM.FI) WHERE (Revenue > "100000")
PLUS (KM.KP AS KP1 USING LINK 1)
PLUS (KM.KP AS KP2 USING LINK 2)
```

Query Result: Tickets with Priority "High" for Companies with Revenue > 100000

Output fields: Problem Description, Company, End Customer Person and Contact Person

## Filtering by Catalog Values

You have the following possibilities when specifying **variable catalog** values as `<filtervalue>`:

- Search for **code**: `#<catalog code>`

**Example:** `select from (FI) where (Country=' #29 ')`

For hierarchical catalogs a valid filter value consists of the child and the parent code, separated by "~": `#<child>~#<parent>`

**Example:** `select from (AR) where (Product='#3~#1' OR Product='#3~#4')`

- Search for (part of) the catalog **text**: `<text>`

`<text>` may contain wildcards (\*).

---

**Note:** Note: You can use literal character values of '\*' and '?' using the escape sequences '\\*' and '\?'.  


---

Examples:

`Country=' Deutschland '`

`Country=' *land '`

- Search for (part of) a **description**: `#desc:<search string>`

`<search string>` may contain wildcards (\*).

**Example:** `select from (FI) where (Country=' #desc:my* ')`

- Search for **external key**: `#xk*:<external key>`

`<external key>` may contain wildcards (\*).

Examples:

`Country=' #xk:100* '` searches for records with an external key starting with "100"

`Country=' #xk*:100_2_28 '` searches for exactly this external key.

- For description and external key you can specify the **language** to be searched:

`Country=' #desc,2:my*'` searches for description "my\*" only in language no. 2.

`Country=' #xk,1*:100_2_28 '`

---

**Note: Note:** Only the specified language is searched, i.e. no parent languages are searched.  


---

---

**Note:** Note: You cannot use an external key filter inside a hierarchical catalog filter.

---

Notes on filtering by **fixed catalog** values:

- Search by code or text:

```
select from (FI) where (LeadStatus='#1')
```

or

```
select from (FI) where (LeadStatus='Interest')
```

- Specifying no value (e.g. `select from (FI) where (LeadStatus='')`) results in the value with code '0' ("Cold" in this example).

# D

## URL Parameters

---

You can pass various settings via URL parameters when starting Aurea CRM.Web or CRM.Designer.

Make sure to use URL encoding ("percent-encoding") for special characters.

Use `&` to specify more than one URL parameter: `http://<my web address>/crm/start?lang=ger&user=myloginname`

### Aurea CRM.Web

The following URL parameters are available for Aurea CRM.Web:

- **debug:** Starts Aurea CRM.Web in debug mode displaying the client log in a popup window, >> [Client Log](#) on page 22 on page 7.
- **user:** Specify the user's login name:

**password:** Specify the user's password.

`http://<my web address>/crm/start?user=<myloginname>&password=<mypwd>`

- **language:** Specify the login language (3-character code or language no.):

`http://<my web address>/crm/start?language=ger`

`http://<my web address>/crm/start?language=6`

- **phoneProfile:** Specify the phone profile to be loaded when the user logs on.

`http://<my web address>/crm/start?phoneProfile=abroad`

If the specified phone profile does not exist, the default profile is used instead.

- **touch:** Use this parameter to load CRM.Web in touch enabled layout (`touch=true`) or standard layout (`touch=false`) mode. For example:

`https://www.abc.com/crm.web/start?touch=false`

---

**Note:** Sometimes the Aurea Web application may respond incorrectly to inputs from pointer devices (for example, a mouse pointer). This is because your browser for some reason might be assuming that you have touch devices (for example, touch screen) installed on your laptop or computer. In such cases, you can enforce the presentation of Aurea CRM.Web application in the standard mode, add the flag "touch=false" to the login page URL, of the CRM.Web application. See the example above.

---

**Note:** You can also enforce the default layout for all users to `touch` or `standard` by setting the `DefaultLayout` parameter in the CRM.Designer. For more information, see [DefaultLayout](#).

---

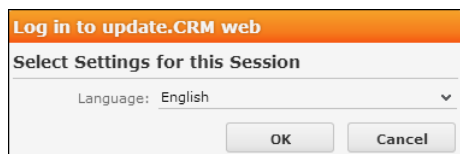
- **roles:** Specify one or more roles to be loaded when the user logs on. Separate multiple roles by comma.

`http://<my web address>/crm/start?roles=Admin,Sales`

For `language`, `phoneProfile` and `roles` you can use `*` to allow users to choose the language/phone profile/role after login.

**Example:**

`http://<my web address>/crm/start?user=<myloginname>&password=<mypwd>&language=*`



---

**Note:** The respective `<CanChange*>` tag in the `settings.xml` must be set to true for `*` to work, >> `settings.xml` (Aurea CRM.Web) in the Aurea CRM.Web Installation Guide.

---

- **theme:** Specify the theme (CSS) to be loaded when the user logs on to a device supporting both mouse and touch gestures, >> [Customizing Styles](#) on page 15 on page 3.

`http://<my web address>/crm/start?theme=update.touchable`

## CRM.Designer

The following URL parameters are available for CRM.Designer:

- **debug:** Starts CRM.Designer in debug mode displaying the client log in a popup window, >> [Client Log](#) on page 22 on page 7. In debug mode the CRM.Designer main page displays an additional legacy column containing deprecated configuration units. Use legacy items for reference only!
- **configurationid:** Specify the numeric ID of the configuration for which to start CRM.Designer:

`http://<my web address>/designer/designer.aspx?verticalid=1&ConfigId=1004`

- **verticalid:** Specify the vertical for which to start CRM.Designer:

`http://<my web address>/designer/designer.aspx?verticalid=1`

**Valid values:** 0 (Base) | 1 (BTB) | 2 (FS) | 3 (OTC)