Crm Customer Relationship Management

CRM Pad Administrator Guide

Version 4.20

Aurea
Unlimited Possibility

Notices

Table of Contents

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.
`Bold Fixed-width`	**`Bold Fixed-width`** font is used to indicate user input or to emphasize certain lines of code.
Italic Fixed-width	*Italic Fixed-width* font indicates a placeholder for which you must supply a value.
Bold Sans serif	**Bold sans serif** typeface indicates the names of graphic user interface elements such as dialog boxes, buttons, and fields.
Italic serif	In text, *italic serif* 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`-LOGFILE` parameter in a command, you only need to enter `-LOGF`.
[ ]	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.
Menu > Choice	An angle bracket between two menu items indicates that you should choose an item from a menu. For example, the notation **File** > > **Exit** means: "Open the **File** menu and choose **Exit**."
>>	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 athttps://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

Introduction

CRM.pad is an Aurea CRM application, enabling users to view and edit CRM data with Apple's iPad.

CRM.pad supports all available verticals and all industry solutions. It is configured with CRM.designer using the `update.tablet` configuration as default configuration.

Note: CRM.pad shares its default configuration (`update.tablet`) with CRM.mobile. Therefore, keep in mind that changes to configuration units (e.g. a Details control or a placeholder definition) may affect the appearance and/or functionality of both applications.

Users can work with CRM.pad online as well as offline, therefore, data must be regularly synchronized.

This document describes how to configure CRM.pad and covers topics such as:

- Server and client architecture
- Client user interface
- Data model and datasets
- Synchronization
- Setup
- Customization using CRM.designer
- CRM.pad-specific configuration units

Additional information is provided in the following manuals:

- For details on installing Aurea CRM web and CRM.designer, see Aurea CRM web Technical & Installation Guide.
- For general information on using CRM.designer and customizing web-based Aurea CRM applications, see Aurea CRM web Administrator Guide.
- The administration and configuration of core functions are described in the CRM.core Administrator Guide.
- For information on the CRM.mobile configuration, see CRM.mobile 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.

System Requirements

Learn about the system requirements for CRM.pad.

See https://support.aurea.com for details on the system requirements and a list of supported devices and browsers.

Documentation PDFs

Download the pdf help CRM Pad Admin guide here.

Document	Description
CRM Pad Administrator Guide	An admin manual for the CRM Pad.

Prerequisites

CRM.pad only works in combination with Aurea CRM web.

An Aurea CRM web installation must exist as the Mobile Services server module uses the same process. Mobile Services server debug information can be found in the `web.log` file.

The following Aurea CRM web features also apply to CRM.pad:

- Users and configurations must already exist in Aurea CRM web. For information on how to create new users and sub-configurations, see the Aurea CRM web Administrator Guide.
- Access rights are based on assigned roles, see the Aurea CRM web Administrator Guide.

Rights

Required module and access right for using CRM.pad.

Check that the required module and access rights were assigned.

Module Rights

The users need the following module rights assigned in order to access CRM.pad:

- Mobile Services (`mmri.exe` or `mmcfg.exe`)

Access Rights

Rights defined in the Aurea CRM win Rights module also apply to Mobile Services:

- View rights (unconditional & conditional) are applied when the application synchronizes data.
- Edit rights (New, Update, Delete) are applied on the server when the user executes an operation.

The user is informed about rights violations when online and about synchronization conflicts, see Synchronization Conflicts on page 380.

The following might help to avoid synchronization conflicts:

- Allow offline changes only for processes if absolutely necessary (even if a feature is not available without a server connection).
- Apply server rights for processes that are only available in online mode.
- Unconditional Must Field (Mandatory Field) rights are applied on the client.
- Apply unconditional Table rights by assigning processes that result in designer units not being shown if the user does not own that process.
- You can revoke rights on the client for New and Edit by defining a Rights filter for an operation, e. g. when the user tries to add a record. In this case, the operation can only be carried out if the parent link record for New meets the filter criteria.

Restrictions

CRM.pad (and also Apple iPad itself) does not support all of Safari's browser communication features.

For example, you cannot use `https` in combination with a certificate based on a root certificate that is **not** from a trusted authority (like Verisign).

Languages and Texts

All available languages are included in the `update.tablet` configuration.

Some texts are displayed before a server connection can be established. If you use another language than English, German or French, these texts are displayed in English.

When users log on, they can select the desired language.

Languages and texts are handled by CRM.designer. Languages are installed with language patches, as for every other update application.

Data Stored on the Device

The following data is stored on the device:

- A database containing CRM data (records and catalogs) in the user's language
- A database containing the application's configuration in the user's language
- A database containing the uncommitted data (records saved by the user but not yet synchronized with the server)
- Resource files transferred with the configuration
- Documents from the CRM database

You need to define the offline data model and offline record and data sets, see Data Model on page 49 and Offline Record Set / Offline Data Set on page 51.

2

Installation and Setup

Learn how to install and setup CRM.pad.

To be able to use CRM.pad, you need:

- The CRM.pad app for the user, available in the Apple App Store free of charge.
- Mobile Services (the name of the server application enabling the use of CRM.pad), included in the installation package of Aurea CRM web and in Aurea CRM web patches.

When an Aurea CRM web version is installed, Mobile Services is automatically installed as well, if it is included in your license. The `update.Mobile.Services.dll` is installed in the `\bin` directory and the `settings.xml` and `web.config` files are changed accordingly, see Mobile Services and IIS 7.0 on page 13.

You need to enable the CGI restrictions for .net Framework (.net Framework is included in your Aurea CRM web installation). Start Aurea CRM web with .net Framework 4.0 and enable the CGI restrictions.

Mobile Services and IIS 7.0

If Mobile Services is installed on an IIS 7.0 server, HTTPHandlers are ignored and only handlers are used.

Therefore, you need not change the `httpHandlers`section of the `web.config` file, but must add the following entry underneath the `</system.web>` (closed system web) tag:

```
<system.webServer>
 < handlers >
 <add name="mobileServices" verb="POST ,GET " path="mobile.axd"
type="update.web.Services.mobileServices,update.web" />
</handlers>
</system.webServer>
```

If you want to install a hotfix, see the "Installing Hotfixes" document that you can download from https://support.aurea.com.

Registering the Server

You have to register on the Aurea CRM server to be able to use CRM.pad.

You cannot use CRM.pad unless you register your Aurea CRM web server. The server can be registered either by entering the following URL into Safari, or - if the URL was sent to you by e-mail - by clicking on the link:

```
crmpad ://configureserver?identification=<IDENT>&name=<NAME>&url=<URL>

&authenticationType=username&useragent=ipad
```

where

- <IDENT> and <NAME> can be the same. <IDENT> is CRM.pad's internal ID of the server and <NAME> contains the name displayed in the login mask.
- <URL> contains the server's base URL, usually identical with the Aurea CRM web installation.
- <AuthenticationType> defines how the Mobile Services server connects to the Aurea CRM web server (ssocredentials, sso, username or revolution), see Authentication on page 16.
- You can also specify the user agent when registering your server. Add the userAgent parameter to the URL, e. g. userAgent=ipad.
- If you are using MobileIron Sentry, then set the enablemobileiron parameter to true in the URL to ensure that you are able to connect to the CRM Server. For example, enablemobileiron=true. For more details, see MobileIron integration with CRM.pad application.

Defining a Default Registration

If no server has been registered yet, the server https://my.update.com/ is registered automatically with "update" as <NAME>. This is applied in SaaS environments where you do not need to register a server.

If you want to set a default registration that automatically accesses your (in-house) server, you can edit the info.plist file (stored in the Products/Applications/CRMpad.app directory) using XCode. You must sign this file afterwards anew.

Note: You can only define a default registration if the app itself has been signed, for information see see CRM.pad Deployment document that you can download from https://support.aurea.com.

To define the default server:

1. In the info.plist file, the com.update.CRMpad.defaultServer property is set to:

   ```
   crmpad://configureserver?identification=update&name=update&url=https://my.update.com/log
   &networkUsername=&networkPassword=&defaultErrorReportingEmailAddress=&userAgent=&authent
   ```

   Change this property accordingly to your server.

2. You can also define that a user cannot log on to any other server by setting the com.update.CRMpad.defaultServer.force to YES (default value: NO).

Including User Credentials in the Registration

You can also add the user's credentials to the registration URL, e.g. if you want your users to access CRM.pad directly from another app.

If you allow access to CRM.pad directly, the user credentials must always be provided when the user logs in for the first time. The credentials are stored on the device.

The following options are available:

- `username`: the user's login name. Must be given if the `loginMode` option is used.
- `loginMode`: Defines how the login is carried out.
  - `savepasswordlogin`: The password is stored locally and the user is logged in automatically.
  - `savepassword`: The password is stored locally, but the login screen is displayed. The credentials are pre-filled, but the user must login manually

Example: `crmpad://configureserver?identification=CRMpad&name=CRMpad&url=http://myserver/updatecrm&authenticationType=username&username=Seller2&loginMode=savepasswordlogin`

Note: Ensure that the `Login.SavePasswordAllowed` Web Configuration parameter is set, see Login.SavePasswordAllowed.

Displaying the Server Before the First Login

By default, servers to which you can log on to are only available on the login page after your first login.

Add the following to your registration link to register the default server automatically:

```
com.update.CRMpad.defaultServer.registerFirst=yes
```

Removing the Server

You can remove a registered server by entering the following URL into the browser:

```
CRMpad :/ /removeServer?name=<NAME>
```

Replace `<NAME>` with the server name that you have entered in the `name=<NAME>` parameter when you registered the server.

Note: `removeServer` does **not** remove the local database on the client. If you re-register the server using the same `<IDENT>` and `<NAME>` as before, CRM.pad connects to this existing database.

Demo Servers for Industry Solutions

By default, if you did not yet register your own server yet, CRM.pad is automatically registered to Aurea CRM's demo servers of the available industry solutions.

This default setting is replaced by the server you register.

You can use the `Login.ShowDemoServer` Web Configuration parameter if you want to display Aurea CRM's demo servers in addition to your server, see Login.ShowDemoServer.

Authentication

After starting CRM.pad, the user must enter the Aurea CRM web server (URL) s/he wants to connect to, the user name and the corresponding password.

Depending on the server type, the following access types are available:

- Anonymous Server Access: Authentication to the CRM application is handled with the Aurea CRM credentials (user name and password, authentication type `username`).
- Authenticated Server Access: Authentication to the CRM application with the Windows user name and password. Single Sign-on must be enabled for the CRM application (authentication type `ssocredentials)`.

  If the user is already authenticated in the WLAN (e. g. VPN), no user name and password is needed (authentication type `sso`).

  Note: A user logs on to CRM.pad in offline mode. If s/he changes the credentials online (e. g. the Windows password on the server) while being in offline mode, the user must log on again to CRM.pad with the new credentials when s/he switches to online mode.

  Note: If a user does not enter a user name, depending on the configuration either Single Sign-on is applied or the login fails.

- Revolution Server Access: Authenticate with e-mail address and password (authentication type `revolution`).

  If login failed, you can check whether authentication was successful by entering one of the following URLs into Safari:

- Anonymous access:

  `<myServer>/mobile.axd?Service=Authenticate&Username=<name>&Password=<password>&ClientVersion=<AppStoreVersionNumber>` (AppStore version number: e.g. 2.x)

- Authenticated access:

  `<myServer>/mobile.axd?Service=Authenticate&ClientVersion=<AppStoreVersionNumber>` (AppStore version number: e.g. 2.x)

If content looking similar to the following is displayed, authentication was successful:


```
{"authenticated":true,"sessioninfo":["100",[["repName","Alexander Holzer"],["repId","100000035"],
["repGroupId","010000003"],["repDeputyId","100000067"],["repSuperiorId","100000015"],["repGroupId","010000003"]
["tenantNo","0"],["tenantName","0"],["datamodelVersion","1.0"],["clientRequestTimeout","30"]]]}
```


Note: Unfortunately, no check is available for revolution server access.

Performance Considerations

The performance of synchronizing and searching data on the client depends on the size and complexity of the data model and the offline dataset.

If in addition large data volumes are involved, synchronizing can be very time consuming.

Each dataset is synchronized synchronously in one server request and must be transferred before a timeout occurs:

- The default client timeout is set to 60 seconds and can be changed, see ClientRequestTimeout.
- The default ASP.NET server timeout is set to 90 seconds. You can change it with the Aurea CRM web installation. For further information refer to the Aurea CRM web Installation Guide.

Note: Bear in mind that setting these values higher might affect the application's performance.

For a thin connection (e.g. GSM) the timeout hardly ever occurs. A timeout only occurs if time runs out before the first response package is transferred. A full synchronization on a thin connection probably takes a long time.

Do not use link fields ("Z fields") in your offline data set. Each offline dataset package is requested with one SQL statement, but the server might execute additional read operations for each record containing link fields.

When a dataset is requested, the following is applied and can result in complex SQL statements:

- The defined filters of each info area
- The current user's rights
- Dependencies to other info areas including defined filters and rights

Note: Complex SQL statements might need a long run time resulting in a timeout on the database server. This could result in incomplete record sets being returned to the client. For more information, see Troubleshooting on page 374.

Note: You can set the `RequestMode` option to `BEST` when assigning the page call for a Details control, then the record and its link are loaded from the server if a server connection is available.

To limit the duration of synchronization, you can:

- Define more than one offline data set for one info area, each containing only one filter or one dependency (the full timeout limit applies to each dataset, a record can be contained in more than one dataset).
- Define filters in a less complex way, even if this results in more records being transferred.
- Consider reducing your offline data set.
- Consider reducing the complexity of your rights definition, e.g. by replacing them with triggered information (easier to query).

Monitoring CRM.Pad Synchronization and Performance

Learn how to monitor synchronization tasks performed by users.

You can capture the following detailed information for synchronization tasks performed by each user:

- Synchronization type
- Time taken for synchronization
- Total number of synchronizations performed by the user including the average time, max time etc
- Information per infoarea: Last synchronization date, start and end time, total time, and number of records synchronized.

To capture the log information you have to use the `update.Mobile.Services.Performance` filter in the `settings.xml` file for CRM.Web. Specify the path to the log file using the filePath attribute of the `<Writer>` element. Configure the attributes in the `<Channel>`, `<Writer>` and `<Formatter>` elements as required.

See the example configuration below for details:

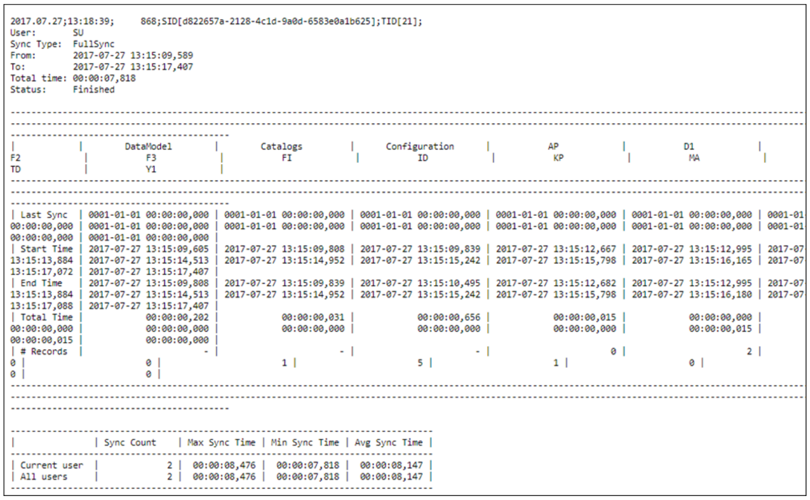
```
<Channel facilityFilter="update.Mobile.Services.Performance"
logLevel="Debug"
logLevelDebug="Debug"
writerName="fileLogWriter4"
```

```
formatterName="webLogFormatter4"/>

<Writer filePath="log\performance.log"
name="fileLogWriter4"
maxSizeKB="8192"
closeOnIdleTimeout="5"
type="update.Lib.Logging.Writers.FileLogWriter,update.Lib" />

<Formatter name="webLogFormatter4"
logDate="true"
logTicks="false"
logFacility="false"
logLevelFormat="None"
logException="false"
logMemory="false"
type="update.Lib.Logging.Formatters.WebLogFormatter,update.Lib" />
```

The following screenshot shows the synchronization log information captured for a user:



```
2017.07.27;13:18:39;     868;SID[d822657a-2128-4c1d-9a0d-6583e0a1b625];TID[21];
User:        SU
Sync Type:   FullSync
From:        2017-07-27 13:15:09,589
To:          2017-07-27 13:15:17,407
Total time:  00:00:07,818
Status:      Finished

------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
--------------------------------------
|            |        DataModel        |        Catalogs         |      Configuration      |           AP            |           D1            |
F2            |          F3             |           FI            |           ID            |           KP            |           MA            |
TD            |          Y1             |
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
--------------------------------------
| Last Sync  | 0001-01-01 00:00:00,000 | 0001-01-01 00:00:00,000 | 0001-01-01 00:00:00,000 | 0001-01-01 00:00:00,000 | 0001-01-01 00:00:00,000 | 0001-01
00:00:00,000 | 0001-01-01 00:00:00,000 | 0001-01-01 00:00:00,000 | 0001-01-01 00:00:00,000 | 0001-01-01 00:00:00,000 | 0001-01-01 00:00:00,000 | 0001-01
00:00:00,000 | 0001-01-01 00:00:00,000 |
| Start Time | 2017-07-27 13:15:09,605 | 2017-07-27 13:15:09,808 | 2017-07-27 13:15:09,839 | 2017-07-27 13:15:12,667 | 2017-07-27 13:15:12,995 | 2017-07
13:15:13,884 | 2017-07-27 13:15:14,513 | 2017-07-27 13:15:14,952 | 2017-07-27 13:15:15,242 | 2017-07-27 13:15:15,798 | 2017-07-27 13:15:16,165 | 2017-07
13:15:17,072 | 2017-07-27 13:15:17,407 |
| End Time   | 2017-07-27 13:15:09,808 | 2017-07-27 13:15:09,839 | 2017-07-27 13:15:10,495 | 2017-07-27 13:15:12,682 | 2017-07-27 13:15:12,995 | 2017-07
13:15:13,884 | 2017-07-27 13:15:14,513 | 2017-07-27 13:15:14,952 | 2017-07-27 13:15:15,242 | 2017-07-27 13:15:15,798 | 2017-07-27 13:15:16,180 | 2017-07
13:15:17,088 | 2017-07-27 13:15:17,407 |
| Total Time |            00:00:00,202 |            00:00:00,031 |            00:00:00,656 |            00:00:00,015 |            00:00:00,000 |
00:00:00,000 |            00:00:00,000 |            00:00:00,000 |            00:00:00,000 |            00:00:00,000 |            00:00:00,015 |
00:00:00,015 |            00:00:00,000 |
| # Records  |                       - |                       - |                       - |                       0 |                       2 |
0 |                       0 |                       1 |                       5 |                       1 |                       0 |
0 |                       0 |
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
--------------------------------------

-------------------------------------------------------------------------------
|              | Sync Count     | Max Sync Time | Min Sync Time | Avg Sync Time |
-------------------------------------------------------------------------------
| Current user |              2 |  00:00:08,476 |  00:00:07,818 |  00:00:08,147 |
| All users    |              2 |  00:00:08,476 |  00:00:07,818 |  00:00:08,147 |
-------------------------------------------------------------------------------
```

You can use the below, optional, configuration for any time-consuming info area to speedup sync by adding additional caching. Update the file performanceSettings.xml in the `..\sys` directory in the following way:

Note: This change needs to be made in *performanceSettings.xml* in *Aurea CRM core Administration Guide*.

```
-<PerformanceSettings>
  <RefreshCycle min="30"/>
   <DumpCycle min="30"/>
    <RecordCache>
     <CacheTypes>
      <CacheType checkInMin="5" reloadInMin="5" maxRecs="500" maxSize="10000"
cache="high" flags="9" id="1"/>
     </CacheTypes>
  <GlobalCache>
    <Table cacheNo="1" cacheType="1" code="FI"/>
  </GlobalCache>
<SessionCache></SessionCache>
<not> </not>
</RecordCache>
</PerformanceSettings>
```

3

Configuring CRM.pad

Learn to configure CRM.pad to provide custom configuration to meet and facilitate your internal processes.

The following topics are available in CRM.pad but are not described in this document. For further details, see Aurea CRM web Administrator Guide for information:

- Defining alternatives, i. e. if you want different icons or pages containing different fields according to the content of a field. For example, appointments can be displayed with a different icon according to its status (scheduled, completed, missed) or an Edit page according to the appointment type (e-mail, letter, etc.).
- Virtual info areas

CRM.designer

The configuration tool to configure CRM.pad to your needs is CRM.designer.

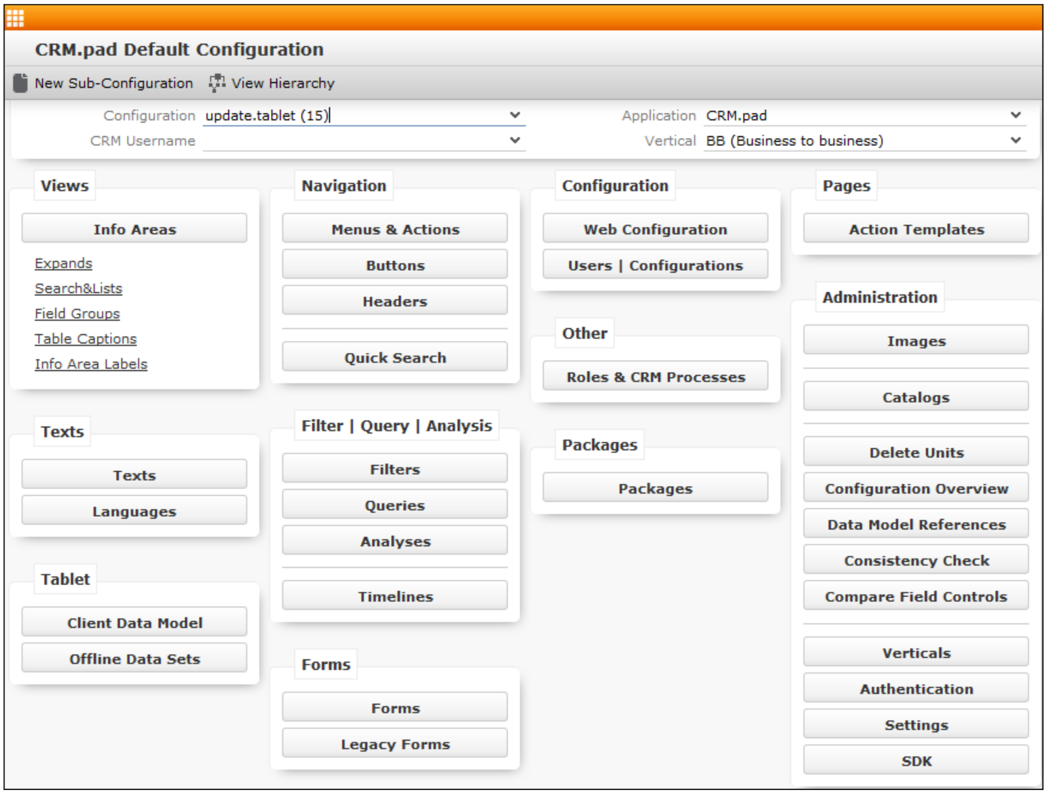
This tool is used to configure all web-based Aurea CRM applications.

Aurea software supplies default configuration called `update.tablet`. This configuration is the base for your configurations. The default configuration is read-only, you cannot change it. You only have access to sub-configurations that you must create.

If you are not used to working with CRM.designer, see the Using CRM.designer chapter in the Aurea CRM web Administrator guide.

To call the `update.tablet` configuration:

1. Start CRM.designer. Aurea CRM web's default configuration is displayed on CRM.designer's main page
2. Select `CRM.pad` from the **Application** drop-down list. The default configuration called `update.tablet` is automatically selected from the **Configuration** drop-down list:



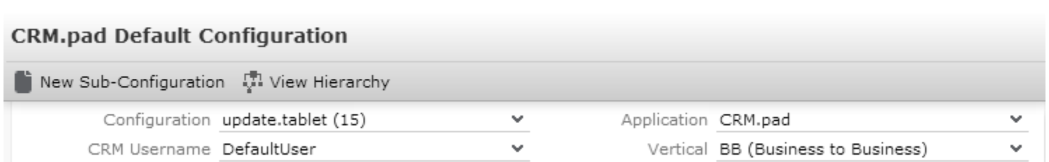
The available options are displayed.

Note: You must create a sub-configuration, as the default configuration is read-only, see Users and Configurations on page 21.

Users and Configurations

Learn how to control the behaviour of CRM.pad.

The behavior and functionality of CRM.pad is controlled by the current user's configuration and its parent configurations.



The default configuration used by CRM.pad and CRM.mobile is called `update.tablet`. This is a root configuration, i.e. it is in **no way** connected to `UPDATE_DEFAULT` used for Aurea CRM web.

You cannot change or delete the `update.tablet` configuration.

To create a custom configuration:

1. On the CRM.designer main page select "CRM.pad" from the **Application** drop-down list.

   The **Configuration** field is automatically updated to `update.tablet`.

2. Click **New Sub-Configuration**.
3. Enter a name and an (optional) description.
4. Click **Save**.

Since it is not possible for a user to change configuration units in CRM.pad and CRM.mobile (define lists, store queries, etc.), the automatic creation of a user configuration is not supported.

All user configurations have to be created and configured by the administrator.

1. If you want to use one custom configuration for all users, assign this configuration to the "DefaultUser".

To create a user:

1. On the CRM.designer start page in the **Configuration** area, click **Users | Configurations**.
2. In the **User** area enter the user's login name as **User Name**.
3. Select a **Configuration**.

   (The **create new config** flag is not supported by CRM.pad and CRM.mobile)

4. Select "CRM.pad" as **Application**.

5. Click + (**Add**).

When the user logs on to CRM.pad or CRM.mobile, the assigned configuration is used.

Web Configuration Parameters and Layouts

Learn to configure web parameters to control the layout and functioning of CRM.pad.

Web Configuration parameters influence CRM.pad. Some of the parameters described in the topics below are not included in the default configuration. If needed, you can add them.

Information on how to set parameter values or add new parameters you find in the Web Configuration topic in the *Aurea CRM web Administrator Guide*.

Calendar Parameters

Learn how to use the calendar parameters for CRM.pad.

CalendarFirstWorkingHour

Overrides the default value for the first working hour of the day displayed in the calendar.

Only integer numbers are currently supported, therefore, you can only define full hours.

Type: Character

Default: 8

CalendarIncludeSystemCalendar

Defines that CRM.pad's calendar also loads records from iPad's system calendar.

Type: Checkbox

Calendar.StartDateLimit

This parameter defines the iPad calendar items displayed in the CRM.pad activities list.

Default value: `$curfdMonth-1`

If you add an exclamation mark (!) before the given value, it is applied to all views (i.e. also in the calendar views).

CalendarNumberOfWorkingHours

Overrides the default value of working hours of a day displayed in the calendar.

Only integer numbers are currently supported, therefore, you can only define full hours.

Type: Character

Default: 10

CalendarView.RememberViewType

Setting this parameter to `true` opens the calender in the view type last used by the user.

Type: Checkbox

Client Timeout Parameters

Learn about client request and client session timeout.

ClientRequestTimeout

This parameter is ignored under iOS6. Requests are aborted after 60 seconds by the client.

This timeout is often set to several minutes to ensure that e. g. the full synchronization request can be carried out without encountering a timeout. This timeout is also the time span until a "Server not available" error is generated if a request is not responded by the server.

Type: Character

Default: 60 seconds

ClientSessionTimeout

Defines the time span in minutes after which the user has to log on anew. This can occur if the user works with other applications and CRM.pad is not on the screen or iPad switches off automatically. However, the operating system can also switch off the application (this can happen before the timeout set in this parameter occurs).

Type: Character

Edit Parameters

Learn about the edit parameters.

Edit.BestAvailable

If enabled (default), the `RequestOption` is set to `Best` for all New or Edit views.

Type: Checkbox

Default: enabled.

Edit.DefaultRequestMode

Defines whether a user can only add or edit records in online mode.

Possible values: `Best` or `online`

Type: Character

Default: `Best`

Format Parameters

Learn about the format parameters.

Format.EmptyForFalse

If enabled, Boolean parameters that are not explicitly set to `false` (i.e. their default value is `false`) are not displayed in the app.

Type: Checkbox

Default value: `false`

Format.EmptyForMidnight

Defines that the value "0:00" is displayed for empty time fields.

Type: Checkbox

Format.Show0InLists

If enabled, the value "0" is displayed in lists for number fields.

Type: Checkbox

Format.Show0InListsForFloat

If enabled, the value "0" is displayed in lists for float fields.

Type: Checkbox

Format.ShowFieldNameForTrueValue

If enabled, the field label is displayed for Boolean fields (instead of YES/NO).

No value is available for FALSE (NO). If you generate a client report (i. e. an xml export), the whole xml element is omitted. To export the xml element, set the `Format.ShowYesNoInReports` parameter.

Type: Checkbox

Format.ShowYesNoInReports

Enable this parameter if you do not want the `Format.ShowFieldNameForTrueValue` to be applied in client reports (i. e. xml export).

Type: Checkbox

Google Analytics Parameters

Learn about the google analytics parameters.

GoogleAnalytics.MainTrackerEnabled

Set this parameter to `true`, to enable tracking of the CRM.pad application by the Google Analytics tracker provided by Aurea. It is disabled by default.

Type: Checkbox

Default: Disabled

GoogleAnalytics.CustomerTrackerId

Set the id value of your custom google analytics tracker to track the CRM.pad application.

Type: Character

GoogleAnalytics.ScreenWhiteList

Enables tracking of the listed CRM.pad application screens you want to track.

Type: List

GoogleAnalytics.ScreenBlackList

Disables tracking of the listed CRM.pad application screens you do not want to track.

Type: List

GoogleAnalytics.ButtonWhiteList

Enables tracking of the listed CRM.pad application buttons you want to track.

Type: List

GoogleAnalytics. ButtonBlackList

Disables tracking of the listed CRM.pad application buttons you do not want to track.

Type: List

GoogleAnalytics.MenuWhiteList

Enables tracking of the listed CRM.pad application menus you want to track.

Type: List

GoogleAnalytics.MenuBlackList

Disables tracking of the listed CRM.pad application menus you want to track.

Type: List

Note: You can use wild cards for the WhiteList and BlackList parameters. For example, adding FI.* to the `GoogleAnalytics.ScreenWhiteList` enables tracking of all screens related to the Company info area.

ListGroup Parameters

Learn about the listGroup parameters.

List.ClassicLayout

Defines whether the "classic" 1.x layout is used. Possible values can be given in JSON syntax:

- empty: The CRM.pad 2.x design is applied.
- `false` (default value): The CRM.pad 2.x design is applied throughout the application.
- `true`: The CRM.pad 1.x design is applied throughout the application.

You can limit the validity for the given setting by entering a JSON array containing specific field groups:

- `Includes`: The layout is only applied to the given field groups.
- `Excludes`: The layout is applied to all the field groups except the given ones.

Example:

```
{"default " :  "true", "includes" : <"fieldGroupConfigName">, "excludes" :
<"fieldGroupConfigName">}
```

ListGroup.NoColor

If enabled, no color coding is applied in lists, i.e. in lists, the colored bar left to an item is not displayed.

Type: Checkbox

ListGroup.NoVirtualInfoAreas

If enabled, color codes and images defined for virtual info areas are not applied in lists.

Type: Checkbox

Log Parameters

Learn about the log parameters.

Log.Class

Name of the class to be logged. For development purposes only.

Type: Character

Log.Config

If enabled, configuration changes are logged.

Type: Checkbox

Log.Method

Name of the class to be logged. For development purposes only.

Type: Character

Log.Questionnaire

If enabled, Questionnaire actions are logged.

Type: Checkbox

Log.Requests

If enabled, all requests are logged.

Type: Checkbox

Log.Result

If enabled, all requests are logged.

Type: Checkbox

Log.SerialEntry

If enabled, all quick add activities are logged.

Type: Checkbox

Log.Statements

If enabled, SQL log changes are logged.

Type: Checkbox

Log.UpSync

If enabled, all UpSync activities are logged.

Type: Checkbox

LogReport.EmailAddress

Defines the e-mail address of the person that receives the client log files.

Login Parameters

Learn about the login parameters.

Login.AllowOnlineLoginOnly

Use this parameter to force a user to work only in online mode. By default, CRM.pad users can work in the offline and online mode.

If set to `true`:

- In the login screen, user is not be able to switch to offline mode using login options.
- If user is not able to connect to server/internet, error is shown to the user.

If set to `false`:

- The CRM.pad user can work in both online and offline mode.

Note: For this parameter to be activated on CRM.pad, the user has to login into CRM.pad in the online mode, during this the web config parameters are synchronised and the parameter activated.

Default: `false`

Type: Checkbox

Login.CaseInsensitive

By default, the password is checked by the server case-sensitively. However, some servers always require lower case passwords. If set to `true`, the password is always submitted in lower case.

Default: `false`

Type: Checkbox

Login.CustomBackgroundImage

Defines the background image that is displayed on the login screen.

Recommended size: 1024 x 680 px.

Type: Character

The image must exist in the **Images** area of CRM.designer, see Images on page 220.

Login.CustomImage

You can add a logo to the login screen using this parameter.

Fixed size: 650 x 50 px.

Type: Character

The image must exist in the Images area of CRM.designer, see Images on page 220.

Login.DisableBlurredBackground

If enabled, the background image is displayed full screen with a value of `Login.FullscreenAlpha` = 50.

Default: `false`

Type: Checkbox

Login.Fullscreen

If enabled, the background image is displayed full screen with a value of `Login.FullscreenAlpha` = 10.

Type: Checkbox

Default: `false`

Login.FullscreenAlpha

If the background image has a high white content, color display may be distorted. With this parameter you can influence the image display.

Possible values: 0 to 100, defining the opacity of the image (0 = transparent, 100 = opaque).

Default: no value

Type: Numeric

Login.InsightBoard

You can enter the name of an existing Insight Board configuration (i.e. the context menu name). This Insight Board must have the `contextMenu` argument given, see Insight Board on page 263.

Login.SavePasswordAllowed

Defines that the password is stored on the device for future comparison.

This parameter must be set if you want to define that users can log on automatically to the application, see Including User Credentials in the Registration.

Login.ShowDemoServer

If enabled, CRM.pad the user can also log in to Aurea CRM's demo server (even after registering a specific server).

Type: Checkbox

Resource Parameters

Learn about the resource parameters.

Resource.2XExtensions

You can add this parameter if you want to define for which extensions @2x files are loaded. Only original files are loaded for all file types that are not given.

Type: Character (Comma-separated list)

Default value (if this parameter does not exist): `png , gif , jpg , jpeg`

Resource.CachePolicy

If you change resource files (in the `tabletResources` directory), the old file is not always replaced on the device after the synchronization due to IIS expiry settings.

Set this parameter to `true` (or to `5`) to ensure that the latest file is always applied (This parameter overrules the default expiry settings in IIS).

Resource.Ignore2.x

If enabled, additional resources needed for iPad3's retina display are not synchronized to the device.

Type: Checkbox

ResourcePath

Defines the path to the configured server resources.

Type: Character

Default: `tabletResources`

Search Parameters

Learn about the search parameters.

Search.AutoSwitchToOffline

If enabled, changes to the search criteria switch to offline mode.

Example: A user carries out an online search ("Cloud" button) and then changes his search criteria. If this parameter is set, this new criteria is searched offline.

Find-as-you-type is only available offline.

Type: Checkbox

Search.MaxResults

By default, a max. of 10.000 records are searched (then the search is interrupted). You can change the limit for your searches by setting this parameter.

Type: Character

Default: max. 10,000 records

Search.MinimumSectionsForIndex

Defines, when an index is displayed for the company and person info areas. For example, if this parameter is set to "5" (default value), then records that start with at least five different letters must exist.

Add this parameter if needed.

The index does not list all special characters explicitly. For example, the letter Ö is listed under O, the letter Ø is listed under its own section.

An index is displayed is defined in the assigned action if the `Section` parameter was set, see RecordListView on page 117.

Type: Character

Default: 5

Search.NoColorOnDefault

If enabled, no color coding is applied for info areas if the default Expand configuration name is the same as the info area ID.

Type: Checkbox

Search.NoColorOnVirtualInfoAreaIds

If enabled, no color coding is applied for virtual info areas.

Type: Checkbox

Search.OfflineDelayTime

Defines the delay time for the find-as-you-type function. If the user does not edit the search field's content, after 0.3 seconds the search is started. Increase the value of this parameter if you encounter performance problems ("slow searches").

Type: Character

Default: `0.3` (seconds)

Search.OnlineDelayTime

Defines the delay time for the find-as-you-type function. If the user does not edit the search field's content, after 0.3 seconds the search is started. ("slow searches")

Type: Character

Default: `0.3` (seconds)

Search.ReplaceCaseSensitiveCharacters

By default, only case-sensitive ASCII characters are supported in the search.

If enabled, the user can replace a special character (e.g. ä or ç) with ? as a placeholder.

Type: Checkbox

Default: `false`

Quick Add (SerialEntry) Parameters

Learn about the serial entry parameters.

SerialEntry.DisableThirdListRow

Defines whether a third row is displayed in the quick add.

Type: Checkbox

SerialEntry.EnableAutoSave

If disabled, serial entry values are not auto-saved by CRM.Pad. The auto-save functionality is always enabled by default.

Type: Checkbox

Default: enabled

SerialEntry.MarkOrderPositionOnQuantity

If enabled, the blue bar is only shown for items where the Quantity is <> 0.

By default, the blue bar is also shown if a UP record exists.

Order.ShowSumLinInDetails

If enabled, the SerialEntry detail screen in landscape mode displays the total sum of the orders placed and the number of line items ordered.

Type: Checkbox

SerialEntry.RowLineShowEndPrice

If enabled, discounts are also shown in each row. By default, discounts are shown in the total line.

Type: Checkbox

SerialEntry.ShowFreeGoodsPrice

Defines that the value of free goods is also displayed in an order's total line.

Type: Checkbox

SerialEntry.SumLineShowEndPrice

If enabled, discounts are displayed in the total line.

This is the default behavior for CRM.pad. However, the `RowLineShowEndPrice` Web Configuration parameter can be applied to your implementation. You can use this parameter to override the `RowLineShowEndPrice` behavior.

Type: Checkbox

SerialEntry.SumLineWithAllRows

Amounts are summed up only for the displayed records by default in the quick add. If all the available records are displayed, the sum is formatted in bold to indicate that the total is displayed.

If enabled, the total sum (for all items) is always displayed.

Type: Checkbox.

SerialEntry.UnitPriceCheckPriceScaleFor1

If enabled, unit price scales defined for unit 1 are also checked if the base price is set to 0.

Type: Checkbox

SerialEntry.EnableAutoSave

Enable this option to immediately save and refresh the changes to data on a Serial Entry **Details** page. By default, Aurea CRM.Pad saves the changes when the user navigates away from the page or navigates to the next Serial Entry item.

Enabling this option is useful in situations where the user needs to see the results of a change immediately. For example, when the user increases or decreases the quantity of an item in an order, the page is refreshed immediately updating the total number of items ordered. This helps the user confirm the changes before he navigates to the next order.

Type: Checkbox

Order.ShowSumLinInDetails

If enabled, the SerialEntry detail screen Landscape mode displays the total sum of the orders placed and the number of line items ordered.

Type: Checkbox

StartPage Parameters

Learn about the serial entry parameters.

StartPage.LegacyMenuPosition

You can define that the "old" (i.e. from version 1.x) SMARTBOOK menu is displayed on the start page as an Insight Board.

Type: Numeric value

Possible values:

- `0` (default): Places this menu above the first row defined in the `StartPage` Legacy Form.
- `-1`: The menu is not displayed.
- Any number, e.g. `2`: If you enter `2`, the menu is placed between the second and third row defined in the `StartPage` Legacy Form

Sync Parameters

Learn about the sync parameters.

Sync.Config

Add this parameter if you want to define that an incremental synchronization is carried out at specific time intervals. Use JSON syntax to specify the values.

Type: Character

Example:

`{"hours" :[ 2,5,14,17],"weekdays":[4,6], "timezone":"America/New_York"}`

The following options are available:

- You must define one of the following options:
  - `hour`: The time of the day when data is synchronized. You can enter decimal values, e.g. 7,5 for 7:30 h.
  - `hours`: An array of hours when data is synchronized.
- You can define that data is only synchronized on specific days:
  - `weekday`: Number of the desired day (1= Sunday, 7 = Saturday)
  - `weekdays`: Array of day numbers (e.g. [2,3,4,5,6] for Monday to Friday)
- You can set a fixed time zone (i.e. the current time zone is ignored):

  `timezone`: Valid iPad time zone names, see System.iOSServerTimeZone.

  Note: By default, the time zone setting is inherited from the parent configuration.

- You can combine different settings (i.e. different synchronization times on different days):

  `alternates`: Array of different definitions in the same format

  Example: `{…, "alternates":[{"hour":5, "weekday":3} ,{ …}]}`

  Note: If this configuration is invalid or not defined, the default interval for incremental synchronization is applied.

Sync.ConfigOnOff

If enabled, the **Update Configuration** option is available on the control center's Synchronization page.

Default: `true`

Sync.ConflictEmailAddress

Add this parameter and assign the `Character` type. As soon as this parameter is not empty, a button is displayed in the application as soon as a synchronization conflict occurs. With this button the user can switch to a page that lists all the conflicts for which an XML file exists. The user can send these XML file containing the data that caused the conflict by e-mail either for all conflicts or for individual conflicts by e-mail to a desired recipient.

Possible values:

- E-mail address. e. g. of the administrator. This address is copied to the e-mail's **To** field.
- `#`: The user can enter an e-mail address.

Sync.DisableSeparateSession

Defines that the incremental synchronization is carried out in the same server session.

Type: Checkbox

Sync.DocumentSync

Defines, how documents are transferred:

- `OFF`: Documents are not automatically transferred.
- `WLAN/UMTS/G2`: Documents are only transferred, if the selected connection (or a better one) is available. If no connection is available, the documents are queued and the synchronization starts automatically as soon as a connection is available.

Default value: <empty>, resulting in the WLAN option being applied.

Type: Character

Note: This parameter does not influence manual document downloads.

Sync.DocumentUploadMaxSizeForWan

Maximum size in KB of the document to be uploaded.

Type: Character

Sync.DocumentUploadMimeTypesForWan

JSON-Array containing the MIME types that are included in the upload using non-WLAN connections

Example: `["application/pdf", "image/jpeg"]`

Type: Character

Sync.FullSyncBlockIntervalDays

Number of days after the last full synchronization when the user cannot access the application anymore.

This parameter and the `Sync.FullSyncIntervalDays` parameter are mutually exclusive.

Type: Character

Default: 28 (days after the last full synchronization)

Sync.FullSyncIntervalDays

Number of days after the last full synchronization when a full synchronization is carried out automatically.

This parameter and the `Sync.FullSyncBlocIntervalDays` parameter are mutually exclusive.

Type: Character

Default value: 21 (days after the last full synchronization)

Sync.FullSyncDate

Forces all users who login after the specified date and time to carry out a full synchronization of their CRM.pad application.

Type: DateTime

Sync.FullSyncWarnIntervalDays

Number of days after the last full synchronization when a warning is displayed.

Type: Character

Default: 7 (days after the last full synchronization)

Sync.IncludeLinkedData

Allows the CRM.pad application to override preventive filters set on linked records and ensure that linked records are also synchronized.

For example, an appointment record is a linked record to a Person record, filters set on the appointment record might prevents it from being synchronized along with the Person record. Set this parameter to true to override the filters and complete the synchronization of the linked Appointment record. For more information, see Filters on page 193.

Type: Checkbox

Default: False

Sync.IntervalMinutesUMTS

Defines the interval in minutes between automatic incremental synchronizations. Use this parameter if the device is connected by 3G or other means than WLAN.

Type: Character

Sync.IntervalMinutesWLAN

Defines the interval in minutes between automatic incremental synchronizations, if a WLAN is available.

Type: Character

Sync.LanguageOnOff

If enabled, the change language option is available on the synchronization page. If disabled, the user can only log on in his default language.

Default: `true`

Sync.TimestampBufferSeconds

The time span in seconds to correct the current time zone when synchronizing incrementally.

Default value: 10 seconds.

Type: Character

This parameter is not included in the `update.tablet` configuration.

Sync.HideOverlayDuringSync

Enable this to prevent the CRM.pad application from getting blocked when a background synchronization is in progress. This prevents the synchronization overlay screen from locking the application and users can continue to use other functions in the CRM.pad application.

Default: false (Unchecked).

Type: Checkbox

This parameter is part of the `update.tablet` configuration.

System Parameters

Learn about the system parameters.

System.AsyncRequestRetryTime

Interval in milliseconds after which CRM.pad asks the server, whether the request has finished processing.

Type: Character

System.AsyncRequestWaitTime

Interval in milliseconds a server waits whether a request has finished processing.

Set this parameter to a value > 0 enables asynchronous server requests.

Type: Character

System.ClientOptions

Switches the user interface to the "classic" mode (i.e. user interface of version 1.x).

Full synchronization is not required. After the cache was invalidated, the defined user interface is displayed for all users logging on after the change.

Syntax: `{"classicMode" :true }`

Note: This parameter is not applicable anymore as of version 2.1. If set, it is ignored.

System.DisableAsyncExecution

A comma-separated list of services that disables asynchronous processing.

Asynchronous server requests are available for the following services:

- Synchronization
- RecordData
- SaveRecords
- GlobalSearch

If left empty, all services are processed asynchronously.

System.DisableChangePassword

If enabled, the option allowing the user to change the password is not available any more. The user can only change the password after it expires and is, therefore, forced to do so.

System.DisableVirtualLinks

If enabled, virtual links are not applied anymore.

Type: Checkbox

System.DisplayFixCatBySortInfo

By default, fixed catalogs are sorted by code (= sort info + code). If enabled, fixed catalogs are sorted alphabetically by text. `System.SortFixCatBySortInfo` parameter is ignored, see System.SortFixCatBySortInfo.

Default: `false`

Type: Checkbox

System.EnableConnectivityIndicator

If enabled, the connected status indicator shows up on the top right corner of the CRM.pad application. The indicator shows the online or offline status of the CRM.pad application.

Type: Checkbox

System.ErrorTexts

List of field group control(s) (separated by semicolon) that are used to display user-friendly error texts.

Syntax: `[FieldGroupName] .[ ControlName]`, e.g. `FI_GlobalErrorTexts.List`

You can enter more than one field group separated by semicolon. They are processed in the given order, the first hit is displayed.

If no control is given, the List control is applied by default.

Type: Character

System.ExplicitCatalogTenantCheck

By default, catalog values are not checked for assigned tenants in CRM.pad (as this check occurred in values not being transferred from the server to the client).

If you set this parameter, catalog values are checked for tenant assignments.

System.iOSServerTimeZone

The session's server time zone may differ from the iPad time zone. This parameter converts the session's time zone to the iPad Time zone.

Note: The iPad's time zone is only applied after a full synchronization.

Apply the same format than for the TimeZone Web Configuration parameter. You can either use the time zone's short name or the iOS name.

Examples:

ADT = "America/Halifax";

AKDT = "America/Juneau";

AKST = "America/Juneau";

ART = "America/Argentina/Buenos_Aires";

AST = "America/Halifax";

BDT = "Asia/Dhaka";

BRST = "America/Sao_Paulo";

BRT = "America/Sao_Paulo";

BST = "Europe/London";

CAT = "Africa/Harare";

CDT = "America/Chicago";

CEST = "Europe/Paris";

CET = "Europe/Paris";

CLST = "America/Santiago";

CLT = "America/Santiago";

COT = "America/Bogota";

CST = "America/Chicago";

EAT = "Africa/Addis_Ababa";

EDT = "America/New_York";

EEST = "Europe/Istanbul";

EET = "Europe/Istanbul";

EST = "America/New_York";

GMT = GMT;

GST = "Asia/Dubai";

HKT = "Asia/Hong_Kong";

HST = "Pacific/Honolulu";

ICT = "Asia/Bangkok";

IRST = "Asia/Tehran";

IST = "Asia/Calcutta";

JST = "Asia/Tokyo";

KST = "Asia/Seoul";

MDT = "America/Denver";

MSD = "Europe/Moscow";

MSK = "Europe/Moscow";

MST = "America/Denver";

NZDT = "Pacific/Auckland";

NZST = "Pacific/Auckland";

PDT = "America/Los_Angeles";

PET = "America/Lima";

PHT = "Asia/Manila";

PKT = "Asia/Karachi";

PST = "America/Los_Angeles";

SGT = "Asia/Singapore";

UTC = UTC;

WAT = "Africa/Lagos";

WEST = "Europe/Lisbon";

WET = "Europe/Lisbon";

WIT = "Asia/Jakarta";

System.OfflineStationNumber

With this parameter you can define that if `StatNo` and `LNr` for a Link are given, a (dummy) link record ID is generated offline.

The link record ID is generated based on:

- StaNo is set to the value given in this parameter
- LNr is set to RequestNr<<16+RecordNr

Note: Do not use the server's station number to set this parameter.

If you copy this (offline) StaNo/LNr in combination with a Template filter, ensure that the Options input argument is set to {"ComputeLinks":true} in the assigned action call. These values are then only used locally in offline mode and the server calculates these values anew when in online mode.

System.ReachabilityRetry

If the user was working in flight mode (i.e. no server connection was available) with CRM.pad and then a server connection is available again, an error (stating that no server with the given host name could be found) can occur after the system tried to synchronize data (incremental synchronization or UpSync).

If you set this parameter (number of seconds), a request fails, the system retries to connect to the server after the given number of seconds.

Note: The system only retries to establish the connection if the error occurred in a network layer, therefore, restarting the application pool does not solve the problem.

Type: Character

System.RequestControl

Add this parameter if you want to enable the conflict handling for complex server requests and enter the name of the field group you configured for this purpose, see Conflict Handling for Complex Server Requests.

Type: Character

System.SortFixCatBySortInfo

Defines how fixed catalogs are sorted:

- `true` (default): Fixed catalogs are sorted by its texts.
- `false` : Fixed catalogs are sorted by catalog code.

Type: Checkbox

System.SortVarCatBySortInfo

Defines how variable catalogs are sorted:

- `true` : Variable catalogs are sorted by catalog code. This sort order differs from the server's sort order, therefore, data may be displayed inconsistently sorted when accessing data online.
- `false` (Default): Variable catalogs are sorted by its texts.

Type: Checkbox

Time-related Parameters

Learn about the time-related parameters.

TimeMinuteInterval

Defines the minute interval for time fields of the input device. The given number must be a factor of 60.

Possible values: min = 1, max = 30.

Type: Character

TimeZone

Sets the server's time zone.

Other Parameters

Learn about the other parameters for configuring CRM.pad.

Application.TerminateOnLogout

For authentication of type ssocredentials:

If enabled, the CRM.pad process terminates on logout.

For all other authentication types:

Ensure, this parameter is set to `false`.

Catalog.HideLockedInFilters

Defines that locked catalog values are not shown in filters according to the `Locked-CatalogMode` parameter set for Aurea CRM web.

Type: Character

Characteristics.HideAdditionalFields

Defines that additional fields from other info areas than `Interest` and `Interest Group` are not displayed.

Type: Checkbox

This parameter is not included in the `update.tablet` configuration.

CustomIndices

You can define that custom indices are created for the CRM tables on the device so that searches are carried out faster.

You can enter the following:

- As a JSON array, e. g.

```
[{"table": "FI", "columns": [ "F2", "F5" ] }, {"table": "KP", "columns":
["F9", "F10"] }]
```

- As a string with the syntax `<InfoAreaId>:<Field1Id>+<Field2Id>`

  Examples:

  - Index on the F2 field in the FI table:

    `FI:F2`

  - Index on the F2 an F5 field in the KP table:

    `KP:F2+F5`

  - Index on the F2 and F5 fields in the FI table, the F10 field in the KP table and the F10 and F8 fields in the UP table:

    `FI:F2+F5;KP:F10;UP:F10+F8`

  - To index a field in descending order, add `d` before the desired field:

    `FI:dF2+F1`

    Results in `INDEX ON CRM_FI (F2 DESC, F1)`

DashboardCalendarBackground

If your configuration includes a calendar on the dashboard, a background picture is displayed when this parameter is enabled.

Type: Checkbox

DatePickerShowWeeknumber

If enabled, the week number is displayed in the Date Picker in Edit view.

Default: `false`

Type: Checkbox

Delete.Ask

If enabled, a confirmation dialog box is displayed, when the user deletes a record.

Default: `false`

Type: Checkbox

DistanceFilter.MaxValue

In location filters a slider is displayed that the user can use to set the distance within which data is returned when a location filter was applied.

With this parameter you can define the maximum value of the slider.

Default: `10000` (meters).

Type: Character

DocumentInbox

Defines that the user can upload documents to CRM.pad's Document Inbox's and which document type's he can upload.

Syntax: JSON dictionary

Type: Character

The following options are available:

- `enabled`: If set to `false`, an error message is displayed stating that the Document Inbox is not available when the user tries to open a document.
- `excludes`: The given MIME types cannot be opend in CRM.pad.
- `includes`: The given MIME types that can be opened in CRM.pad.

Note: Only specify either the `excludes` or the `includes` parameter, but not both.

Example:

```
{"enabled":"true","excludes" :[ "application/x-pdf","application/pdf"]}
```

FixedCatalog.SortByCode

Parameter is obsolete, still supported for compatibility purposes. Use `System.DisplayFixCatBySortInfo` instead, see System.DisplayFixCatBySortInfo.

FullTextSearch

If enabled, wild cards are automatically added before and after the search criteria. This parameter does not affect the global search.

HttpErrorCodesForOffline

A (comma-separated) list of http error codes that the system does not handle as errors but as offline mode. For example, a synchronization is interrupted if an error occurs that was specified using this parameter.

Type: Character

InsightBoard.TitleEnabled

If enabled, this parameter displays a heading for the insight board. The displayed text is defined in the insight board's **Text** field of the calling Legacy Form, see InsightBoard Type.

Type: Checkbox

InvalidateCacheStrategy

Defines whether the cache is invalidated during synchronization.

Possible values:

- `ALL`: Cache is invalidated for all configurations
- `current` (default value): Cache is only invalidated for the current configurations
- If left empty: Cache is not invalidated

Type: Character

Default: `current`

LogReport.EmailAddress

Defines the e-mail address of the person that receives the client log files.

MaxCountMapView

The max. number of records displayed in Map view.

If you add a map to a search results list, you can add this parameter to limit the hits displayed in the map.

Type: Character.

PresentationMode

Defines, that when, for example, the device is connected to a beamer, a white dot is displayed where the user tapped to call a function.

Type: Checkbox

RepCopySearchAndLists

Defines the field group that is used to define rep filter tokens. If left empty, the `IDCopy` field group is used per default.

Type: Character

Report.EnableCoreReportInBrowser

If core reports are executed in Safari, errors occur. Therefore, the button calling core reports ("Open in Browser") was disabled.

Use this parameter to enable the button.

Type: Checkbox

SendErrorContextInfo

Defines whether context-related information such as screenshots or the navigation path is included in the sent e-mail.

Type: Checkbox

Default: `true`

SendErrorReport

Defines whether an error report can be sent (button in the header).

Type: Checkbox

Default: `true`

ServerIdentification

Name that identifies the server. If set, the client can identify a server even after its URL was changed.

Type: Character

SoftScannerEnabled

If enabled, the user can scan bar codes with the iPad's camera.

Note: You can add this Web Configuration parameter to the user settings in the user's Settings in the Control Center so that users can decide when they want to use the camera or an external device to read bar codes, see Global Actions and SettingsEditView and SettingsView on page 136.

Type: Checkbox

Default: `false`

SortLocale

Defines the collation for sorting.

Possible values:

- `Default`: Uses the `sqllite` BINARY collation
- `Device`: Uses the device's collation
- Country-specific collations are assigned by entering the corresponding ISO Code. You must enter ISO code as a parameter value first, currently only `ru_RU` is predefined.

Type: Character

Note: If you change the value of this parameter, users have to log on anew, otherwise the new sorting collation is not applied.

StartPage

Defines the page displayed after the user logs in or after he taps on **Reset All Data and Reinitialize**.

Version.Minimum

If enabled, the client version is checked (e.g. 2.x) and synchronization requests are rejected if the client application is not at least the set version.

Type: Character

SQLDebugging.Disable

This parameter enables or disables SQL access to the local databases CRM Db, Config Db and Offline Db. It is currently available via the **Control Panel** > **Log Panel** .

When this parameter is enabled, the CRM Db, Config Db and Offline Db buttons currently shown in the offline storage section of the log screen is hidden.

Setting `SQLDebugging.Disable` as true disables the SQL debugging capabilities in CRM Pad.

Check the checkbox to disable SQL debugging or uncheck to enable access.

Type: Checkbox

Default: Unchecked (i.e. SQL access to local DB's is enabled)

UseUserConfiguredEmailClient

Note: Available from CRM.pad 4.22.

This parameter enables or disables user ability to open mail client from CRM.pad:

- The iOS mail client (unchecked)
- The default user-selected mail client. This may be the iOS app or another mail client, such as Microsoft Outlook or Gmail (checked)

Type: Checkbox

Default: Unchecked

Deleted Parameters

List of the deleted parameters.

The following parameters have been deleted:

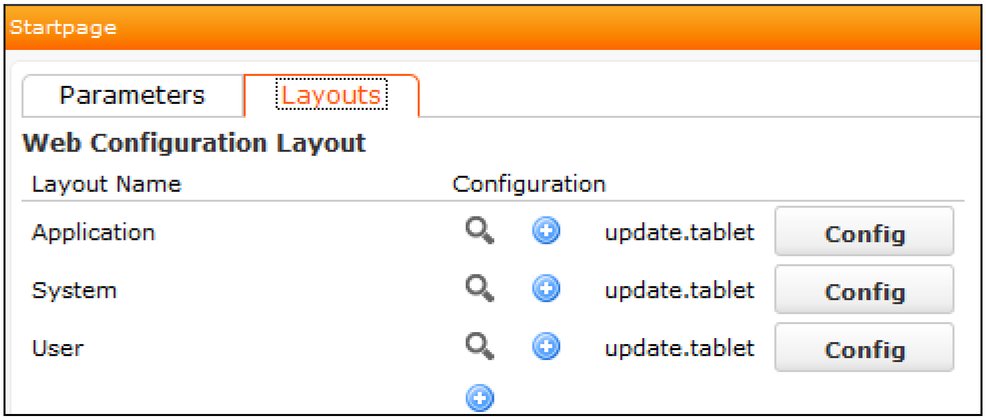
- ConnectivityMode
- History.PageSize
- Search.Mode
- SynchronizationMode

These parameters are still available for compatibility purposes. If they have been previously set, you can still alter their values.

Web Configuration Layouts

A Web Configuration Layout groups several Web Configuration parameters by topic.

The following three configuration layouts are included in CRM.pad by default:



See the Configuration Layouts chapter in the Aurea CRM web Administrator Guide for more information.

Note: Be careful when assigning these configuration layouts in Action Templates. For example, if you enter `System` in the `LayoutName` argument of a user configuration, you allow the user to change system-wide parameters.

Data Model

Users can access the Aurea CRM database in online mode or a local database in offline mode.

CRM.pad uses a subset of the Aurea CRM data model. You must define this subset in CRM.designer. The following picture shows the default BTB data model for CRM.pad:

Client Data Model

Update Data Model

Info Area	Fields	Relations	Unused Fields	Missing Fields
Rep (ID)	21	8	0	0
Company (FI)	33	28	0	0
Person in Company (KP)	16	19	0	0
Item Master (AR)	0	1	0	0
Questionnaire/Portfolio (F1)	2	9	0	0
Question (F2)	12	12	0	0
Answer (F3)	5	10	0	0
Survey (U1)	2	25	0	0
Survey Result (U2)	5	11	0	0
Document (D1)	10	6	0	0
Customer Document (D2)	7	7	0	0
Opportunity (Y1)	15	10	0	0
Activity Participant (MB)	5	4	0	0
Favorites (FV)	2	18	0	0
Document link (D3)	8	24	0	0
Activity (MA)	27	16	0	0
Offer (PR)	9	11	0	0
Offer Item (AP)	10	25	0	0
Order (AU)	0	8	0	0
Opportunity Product (Y5)	6	5	0	0

The data model must be updated at least once for each new sub-configuration. The data model displays all info areas used by the selected configuration (and its parent configurations).

To update the data model:

1. Click on **Data Model** on the CRM.designer start page, then click on **Update DataModel**.
2. After updating the data model, data is synchronized automatically when the user logs in.

Note: If you add (linked) fields to a field group, you **must** update the data model to include them on the user's device.

Offline Record Set / Offline Data Set

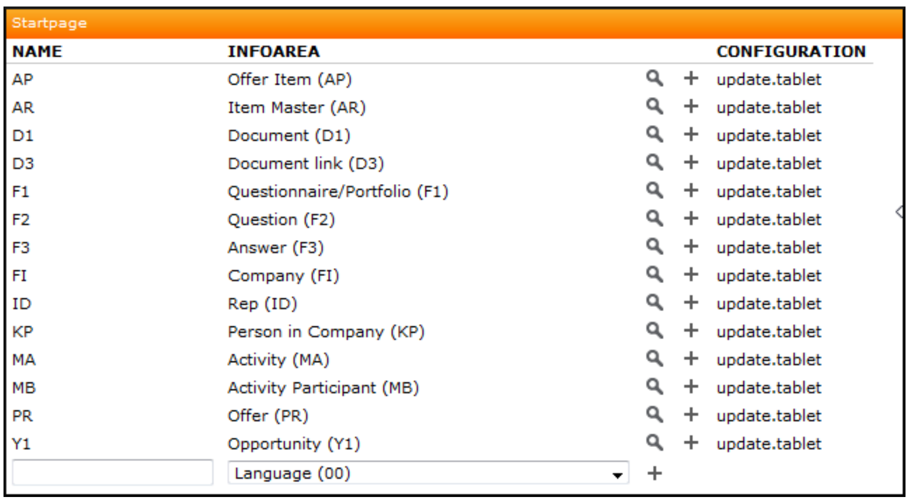
You must define an offline data set that contains all the records that are downloaded to the device. Without this offline data set the rep is not able to work offline.

A default offline data set is included in CRM.pad. You can adapt it to your needs by adding or removing info areas. You must add a dataset for each info area to your record set, otherwise data cannot be transferred to the rep's device.

Reps can still access records that were not downloaded to the device in online mode.

To define an offline data set:

1. Click **Offline Data Set** on the start page.

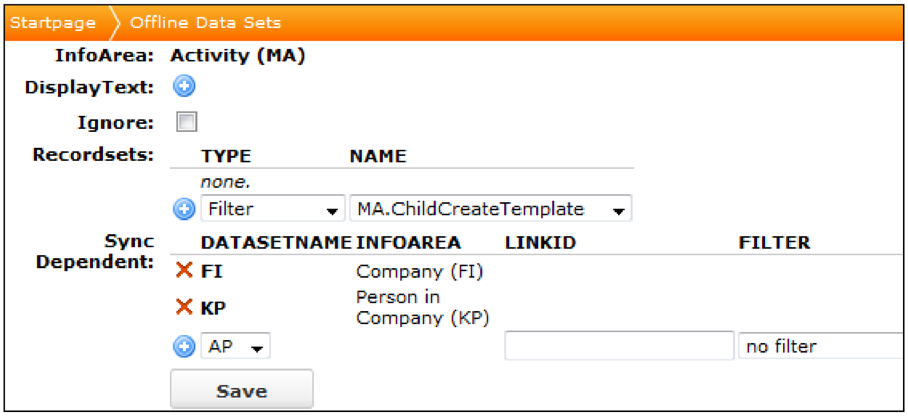


Startpage

NAME	INFOAREA	CONFIGURATION
AP	Offer Item (AP)	update.tablet
AR	Item Master (AR)	update.tablet
D1	Document (D1)	update.tablet
D3	Document link (D3)	update.tablet
F1	Questionnaire/Portfolio (F1)	update.tablet
F2	Question (F2)	update.tablet
F3	Answer (F3)	update.tablet
FI	Company (FI)	update.tablet
ID	Rep (ID)	update.tablet
KP	Person in Company (KP)	update.tablet
MA	Activity (MA)	update.tablet
MB	Activity Participant (MB)	update.tablet
PR	Offer (PR)	update.tablet
Y1	Opportunity (Y1)	update.tablet
	Language (00)	

2. For each offline data set, you can define:
   - The info area.
   - A list of filters. Only the records meeting the filter criteria are available offline.
   - A list of related datasets. Records included in this dataset can be linked offline.

3. When you add a dataset, you can select a filter.



4. You can disable datasets with the **Ignore** flag.

All defined fields and links that were defined in the **Infoareas** section are synchronized for each record.

A record is synchronized if it matches at least one of the filters in the datasets OR is linked to a record contained in at least one of the related datasets listed in the **Sync Dependent** area (i.e. only filters of the related datasets are applied, cascading links are not supported) and also matches the filter if defined.

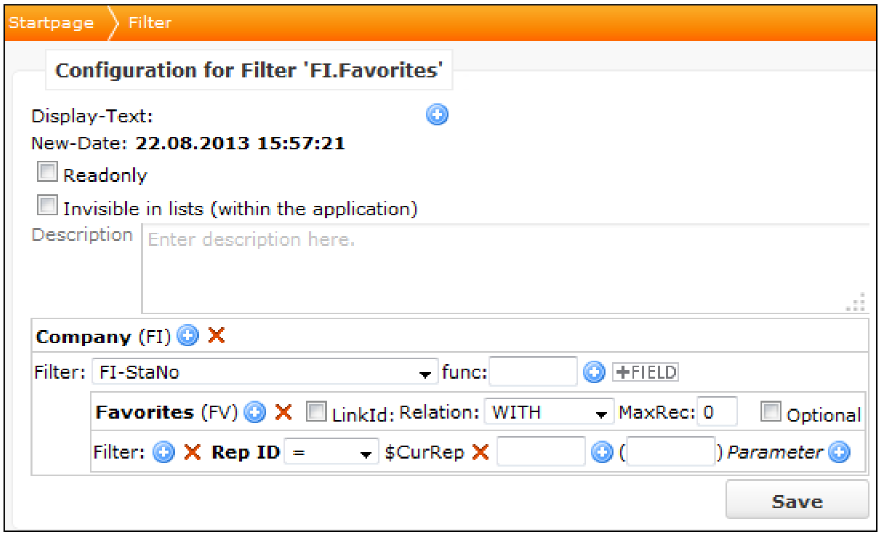
If a dataset contains `(Lookup)` in its name, these records are also synchronized, but they are stored for reference purposes only. The data is shown wherever it was added as linked fields, but is not displayed in Details view or in search results.

Offline Record Set / Offline Data Set - Example

You can define that only the companies and persons of a rep's Favorites are available on the device. In addition, you want persons (not defined as Favorites) of companies in the Favorites, and companies (not defined as Favorites) of persons in the Favorites to also be transferred.

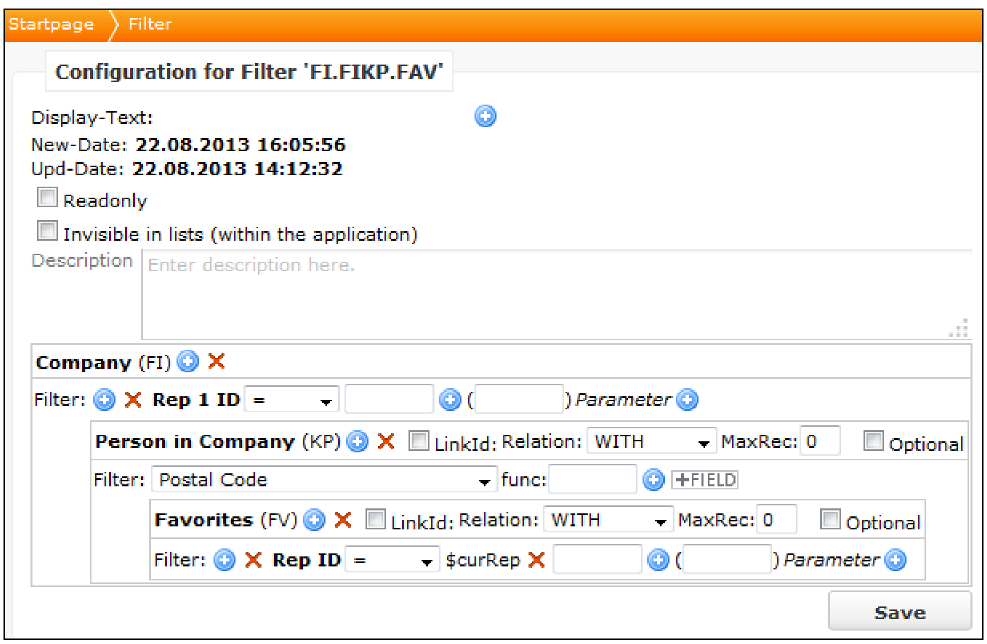
To configure this:

1. Define the following `FI.Favorites` filter:



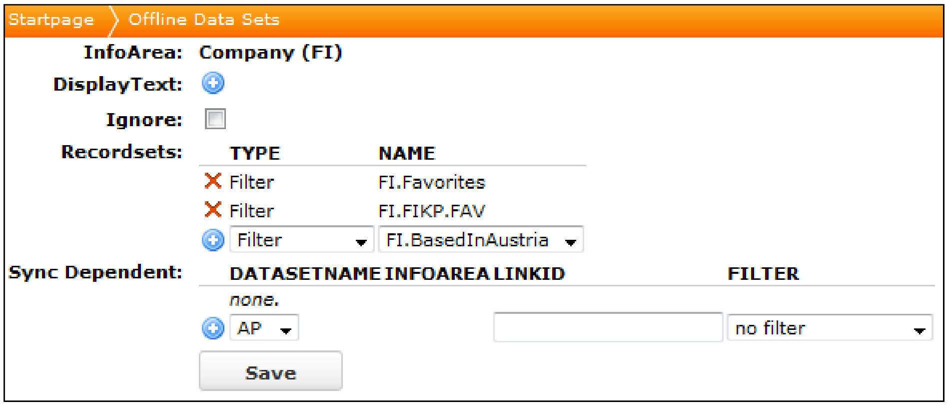
This filter identifies the company records that were added to the Favorites.

2. Define the following `FI.FIKP.FAV` filter:

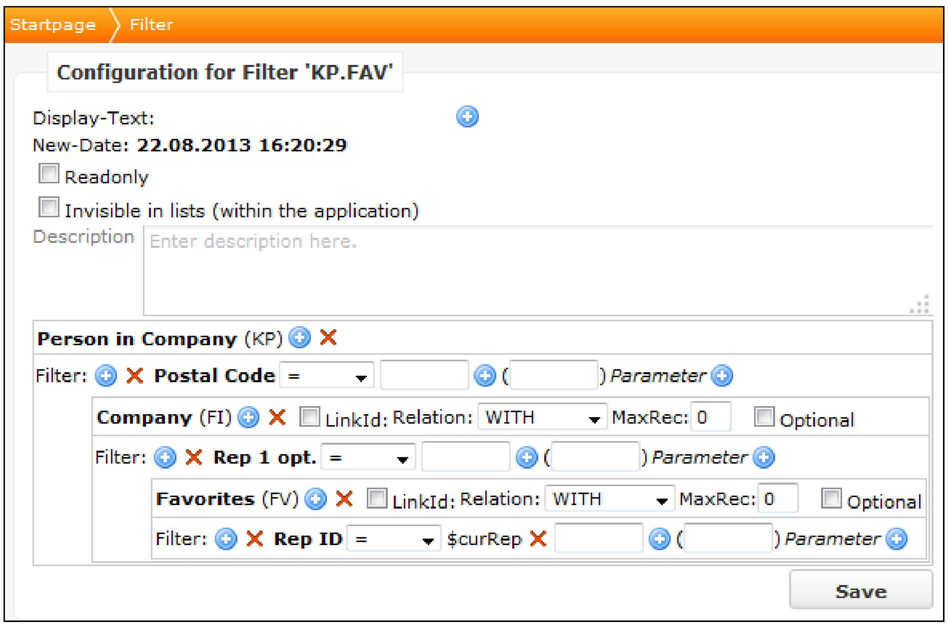


This filter identifies the companies (not defined as Favorites) that are linked to a person in the Favorites.

3. Define the following dataset for FI:

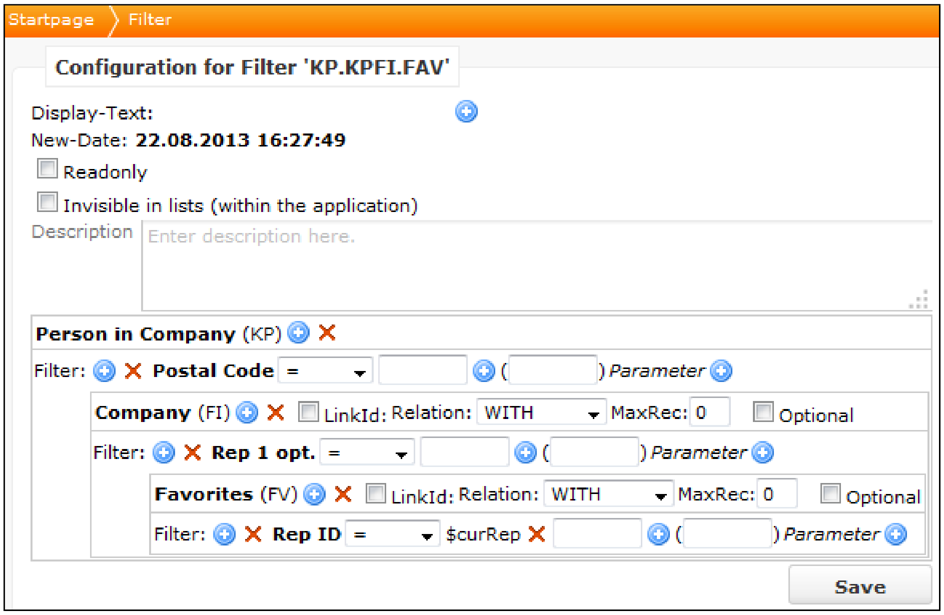


4. Define the following `KP.Fav` filter:



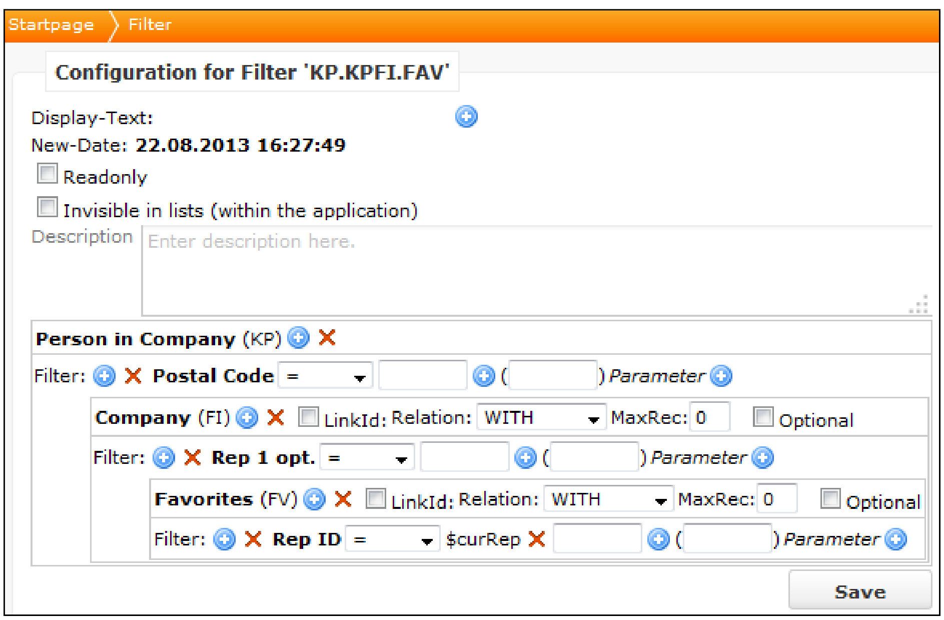
This filter identifies the company records that were added to the Favorites.

5. Define the following `KP.KPFI.FAV` filter:

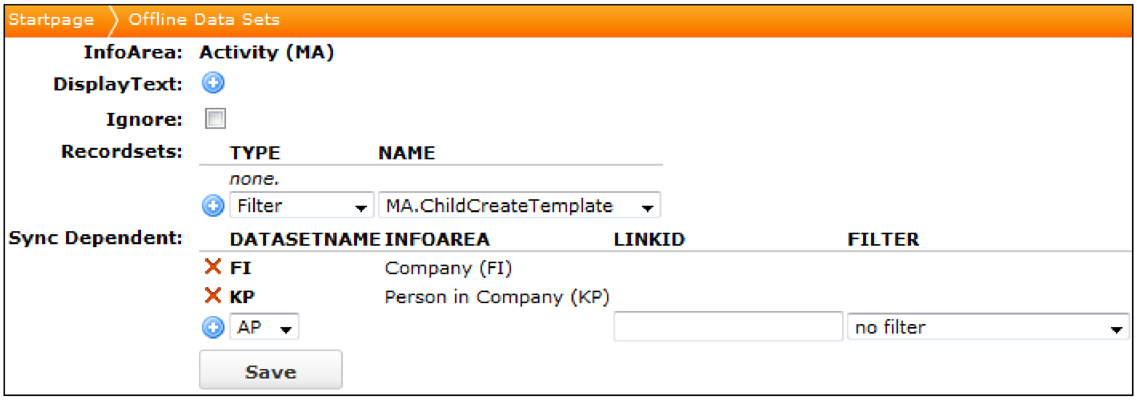


This filter identifies the persons (not defined as Favorites) that are linked to a company in the Favorites.

6. Define the following dataset for KP:



7. Furthermore, define the following Dataset for MA if you want to transfer the appointments for these companies or persons:



Using JSON

You can define parameters using JSON syntax.

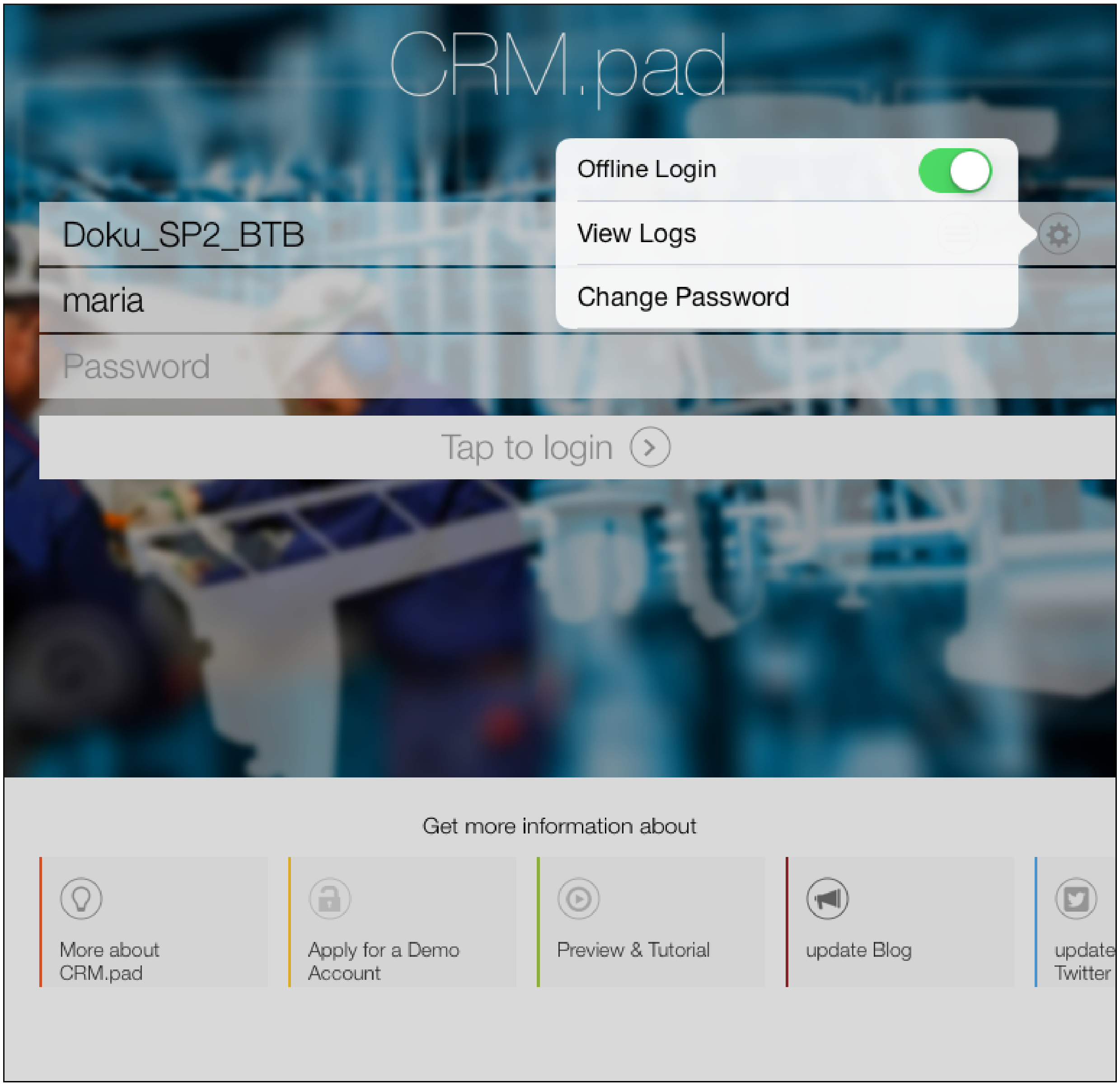
These JSON strings must be valid JSON as determined by the iPad's parser, e. g.

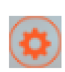
- Must start and end with double quotes
- Each element name must be placed in double quotes
- Double quotes in strings must not be escaped

Example for a valid JSON string: `{"MaxResults":"5"}`

Login Page

The login page allows the user to log on to the application.



The user can select the following options in the  **(Settings)**:

- Switch between online and offline Mode
- View the log file and send it to the administrator
- Change the password.

You cannot add or remove items from the **Settings**. However, you can:

- Add a background image and/or a logo, see Adding a Logo or Background Image to the Login Page.
- Add an Insight Board (included in the `update.tablet` configuration) that, e. g. contains information on the application, a demo or a tutorial, see Adding an Insight Board to the Login Page.

Adding a Logo or Background Image to the Login Page

You can add your company's logo or another background image to the login page.

As images are part of the configuration, you need to synchronize the configuration in order to display the image. Therefore, when the user logs in for the first time, CRM.pad default logo is displayed.

To add a logo or background picture to the login page:

1. Copy the picture to the installation and register the image file, see Images on page 220.
2. Define the Web Configuration parameters that define what images are displayed on the login page, see LoginLogo, Login.CustomBackgroundImage and Login.InsightBoard.
3. Define the Web Configuration parameters that determine how the background image is displayed, see Login.DisableBlurredBackground, Login.Fullscreen and Login.FullscreenAlpha.

   Note: If all three Web Configuration parameters are set to their default values, they are ignored.

4. Carry out a full synchronization.

Adding an Insight Board to the Login Page

You can define that an Insight Board is also displayed on the login page.

The items assigned to this Insight Board must have an alternative context menu assigned (`ContextMenu` argument in the `InsightBoardItem` action call, see InsightBoardItem on page 100).

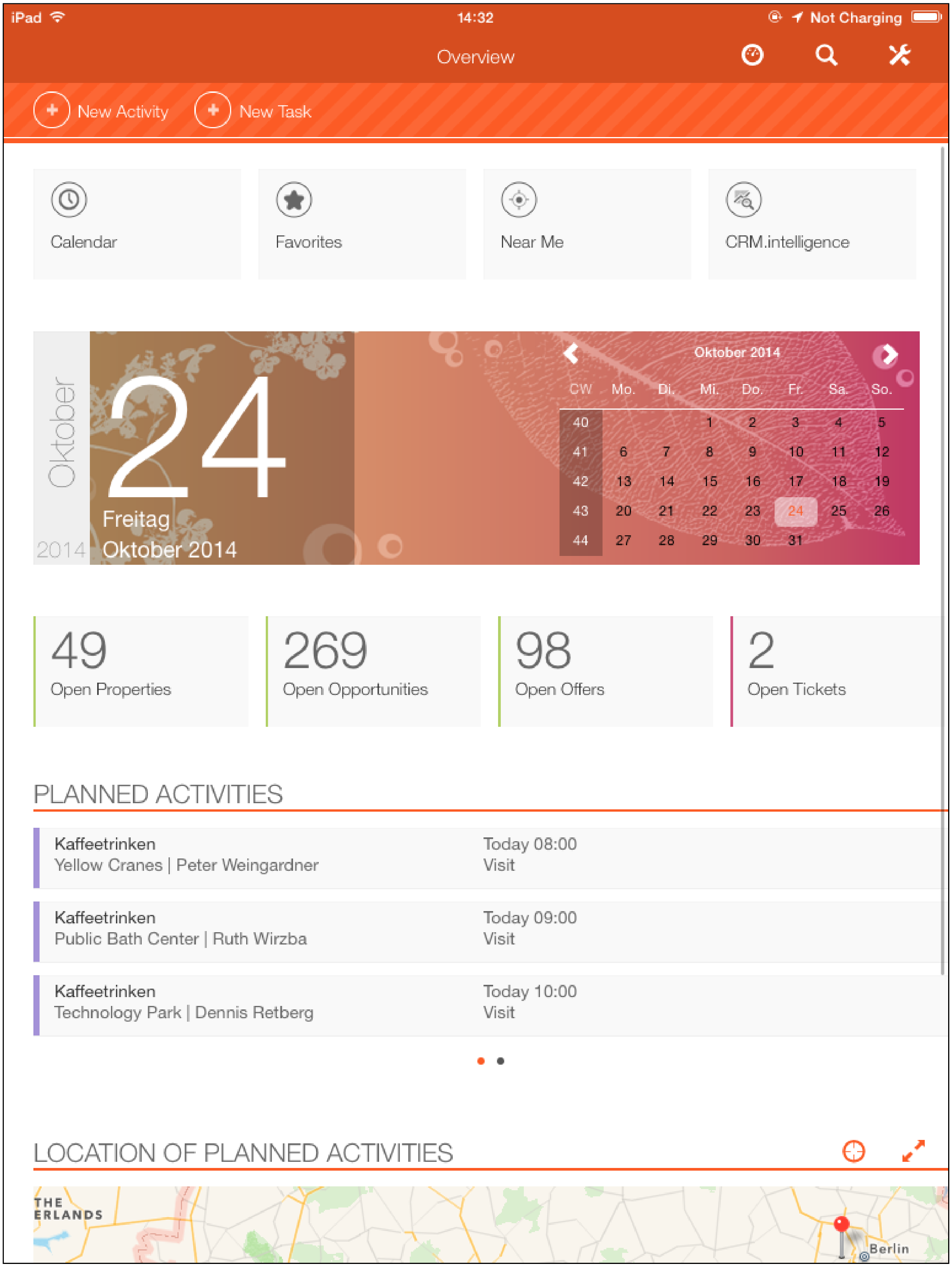
Note: The alternative context menu is restricted to items having the `WebView` action assigned with the `ReportType` argument set to "Url". A tutorial can be assigned by adding `%tutorial%` to the value of the `Url` argument.

Start Page

The start page is the page that is displayed after the user logs on to the CRM.pad app.

It is configured in the **Startpage** Legacy Form, see Legacy Forms on page 221.

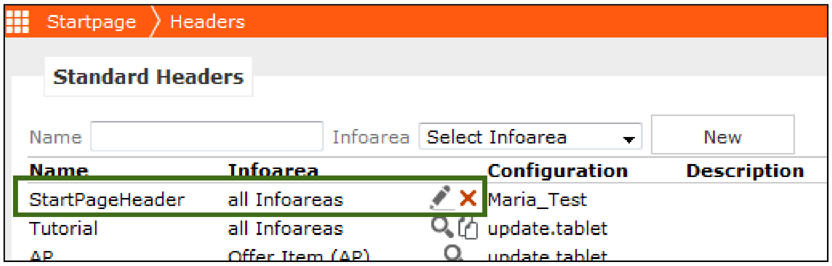
The following example shows the default start page included in the `update.tablet` configuration.


iPad
14:32
Not Charging
Overview
New Activity
New Task
Calendar
Favorites
Near Me
CRM.intelligence
Oktober
24
Freitag
2014
Oktober 2014
Oktober 2014
CW Mo. Di. Mi. Do. Fr. Sa. So.
40 1 2 3 4 5
41 6 7 8 9 10 11 12
42 13 14 15 16 17 18 19
43 20 21 22 23 24 25 26
44 27 28 29 30 31
49
Open Properties
269
Open Opportunities
98
Open Offers
2
Open Tickets
PLANNED ACTIVITIES
Kaffeetrinken
Yellow Cranes | Peter Weingardner
Today 08:00
Visit
Kaffeetrinken
Public Bath Center | Ruth Wirzba
Today 09:00
Visit
Kaffeetrinken
Technology Park | Dennis Retberg
Today 10:00
Visit
LOCATION OF PLANNED ACTIVITIES
THE
ERLANDS
Berlin

Adding Buttons to the Start Page

To add buttons to the start page:

1. Click **Headers** on the CRM.designer start page.
2. Add a header group that applies to all info areas, e.g. `StartPageHeaders`.



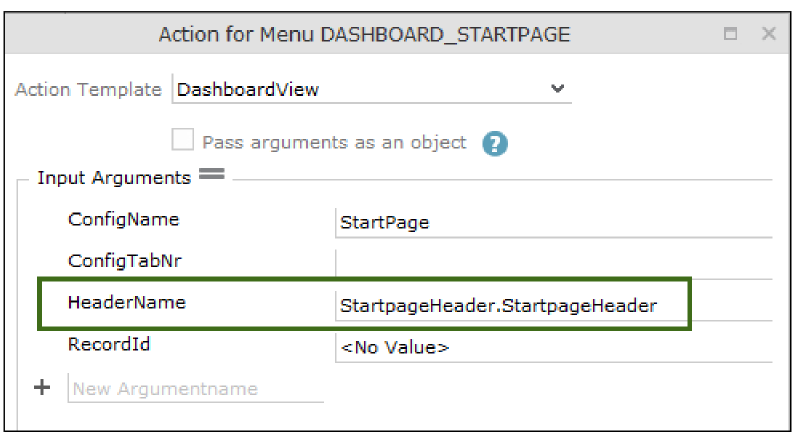
3. Switch to this header group and add a **Special Header**, e.g. `StartPageHeader`.
4. Add the desired buttons to this header. You can only add buttons without a record context, e.g. adding new records like tasks or activities.
5. Enter this special header's name in the `HeaderName` input argument of the `Dash-boardView` action call , DashboardView on page 92. This action is called in the default configuration by the `DASHBOARD_STARTPAGE` application menu action.

   Syntax:

   <Header Group Name>.<Special Header Name>

   Example:

   `StartPageHeader.StartPageHeader`:



Menus

Learn how to configure menus in CRM.pad.

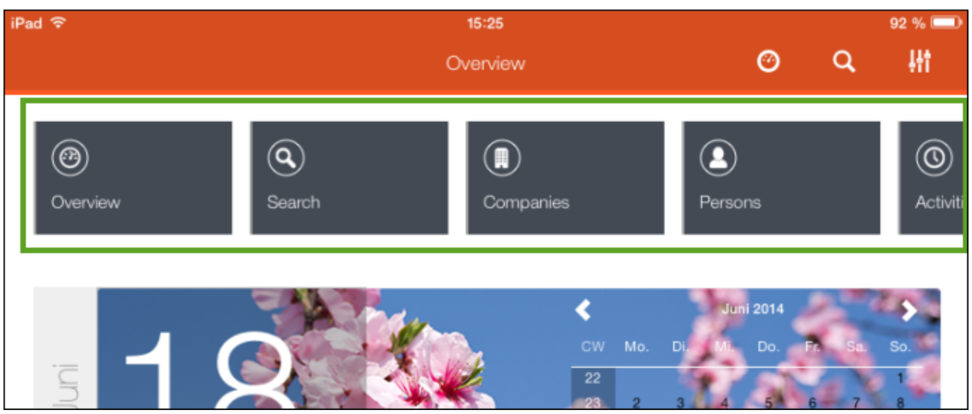
In the previous CRM.pad version 1.x, an application menu was available for navigation. This navigation menu is configured in the **Menus & Action** area with the SMARTBOOK application menu. This application menu is still included in the `update.tablet` configuration.

SMARTBOOK Menu from Version 1.x

In CRM.pad 1.x, the application menu was defined in the SMARTBOOK menu. This navigation menu was configured in the **Menus & Action** area and is still included in the `update.tablet` configuration.

If this menu still exists in your configuration, it is displayed in CRM.pad 2.x as follows:

- The contained menu actions that have the `RecordListView` action assigned are listed in the (**Search**) menu, see Search on page 237.
- The menu items are displayed as Insight Board on the start page, if the `StartPage.LegacyMenuTemplate` Legacy Form exists as defined in the `update.tablet` configuration, see StartPage.LegacyMenuTemplate on page 226.



Note: The SMARTBOOK menu is only available for compatibility reasons, as long as working with both versions, 1.x and 2.x is possible and is removed in the future. It is replaced by two menus `$AppSearchMenu` and `$GlobalActions`, see Global Actions and Configuring the Search Menu.

If you want to change the position where this Insight Board is displayed, set the `StartPage.LegacyMenuPosition` Web Configuration parameter, see StartPage.LegacyMenuPosition.

There are several ways to prevent this menu from being displayed:

- Set the `StartPage.LegacyMenuPosition` Web Configuration parameter accordingly (`-1`), see StartPage.LegacyMenuPosition.
- Remove the row that adds the Insight Board with the **SMARTBOOK** menu assigned from the **StartPage.LegacyMenuTemplate** Legacy Form.
- Empty your **SMARTBOOK** menu. However, do not empty it as long as users are also working with CRM.pad 1.x.

User-defined Menus

The user can navigate through CRM.pad 2.x using the (**Search**) menu. See Configuring the Search Menu on how to define the Search menu.

Buttons

You can define buttons to pages that allow the user to call the assigned page.

You can add the following buttons to the Details (or any other) view:

- `Edit`: Calls `OrganizerAction` action template, see OrganizerAction on page 110.
- `EditWithSerialEntryButton`: Call the quick add, see SerialEntryEdit and SerialEntryNew on page 136.
- `DownloadRecord`: Calls the `SyncRecord` action template, see SyncRecord on page 137.
- `Modify` (not included in the `update.tablet)`: You can add this button if you want to change records based on an existing Template filter by assigning the `ModifyRecord` action call, see Template Filter and ModifyRecord on page 102.

  To add a modify button, configure the `ModifyRecord` page (which actually is a function) to hand over the record and to apply a `TemplateFilter`.
- `Delete`: Calls the `Delete` organizer action to delete the current record, see OrganizerAction on page 110.

Adding a Button

Buttons are added to the header group of the desired info area for each view.

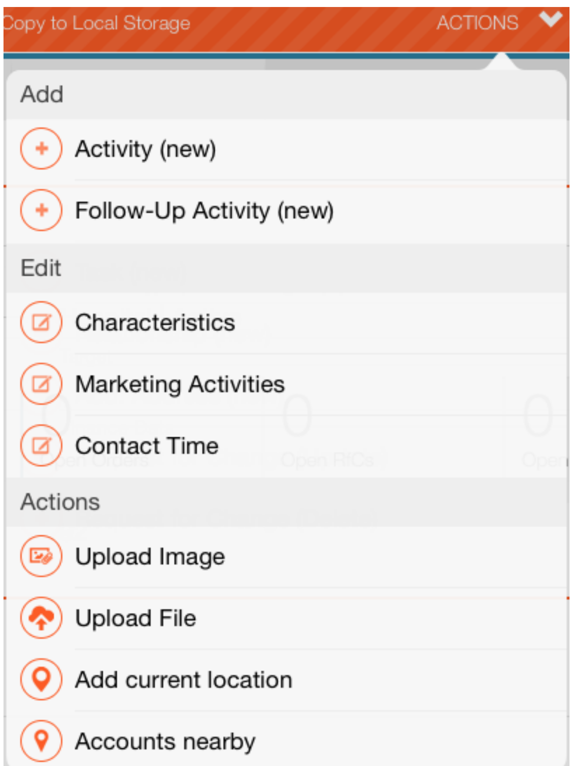
To add a button to the Details (or any other) view:

1. Switch to the header group (e. g. Expand) of the desired info area (e. g. Person).
2. Select a button in the drop-down list at the end of the existing buttons and click on (**Add**).
3. Click on the button's name to open the button's details.

4. Click on **Copy to this configuration** to switch to the Edit mode, then click on (**Edit**) besides **Action** to assign the action that is executed when the user clicks on this button.
5. Define the action's parameters. The mandatory parameters depend on the selected button.

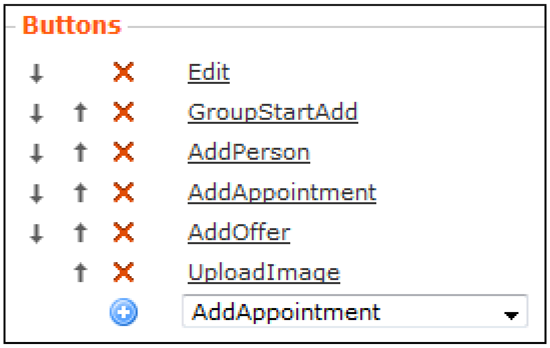
Adding a Button Group

You can also define that one or more buttons are grouped and that the options are displayed in a drop-down list when the user clicks the **Actions** menu in the record's header:



To add a button group:

1. Switch to the header group for the desired info area, e. g. Company.
2. Add the desired buttons to the **Buttons** area:



3. Add the `GroupStart` button (or any button with a name starting with `GroupStart`). The image defined for this button is used as icon for the button group. The GroupStartAdd button in the example above already has the  icon assigned.
4. Add the `GroupEnd` button to mark the end of a group. The `GroupEnd` button is only required if you want to add further buttons at the right side of a group.

Note: The grouping is ignored when no quick actions were defined.

Adding a Toggle Favorites Button

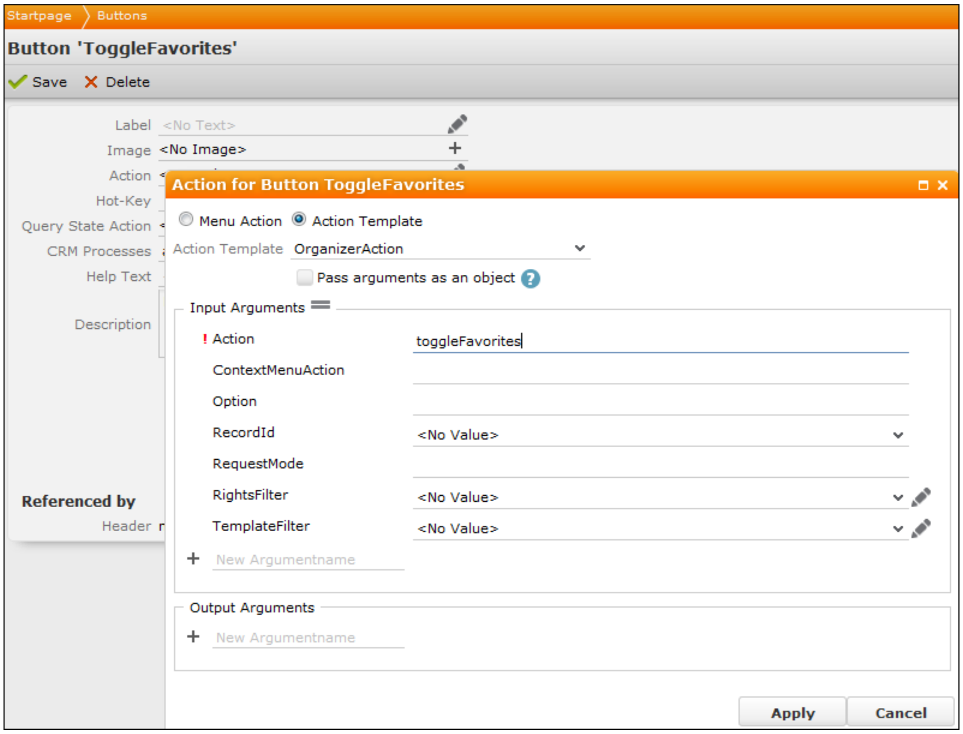
You can add a toggle button to companies or persons that toggles the Favorites state of a record.

To add the toggle button:

1. Add a button called `ToggleFavorites` to the **Buttons** area.

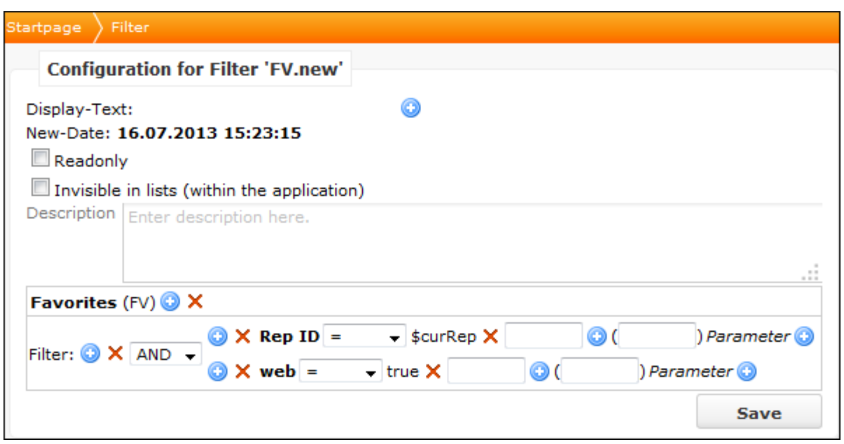
   Label and image are automatically assigned.

2. Assign `OrganizerAction` as **Action** and enter `toggleFavorite` in the `Action` input argument.



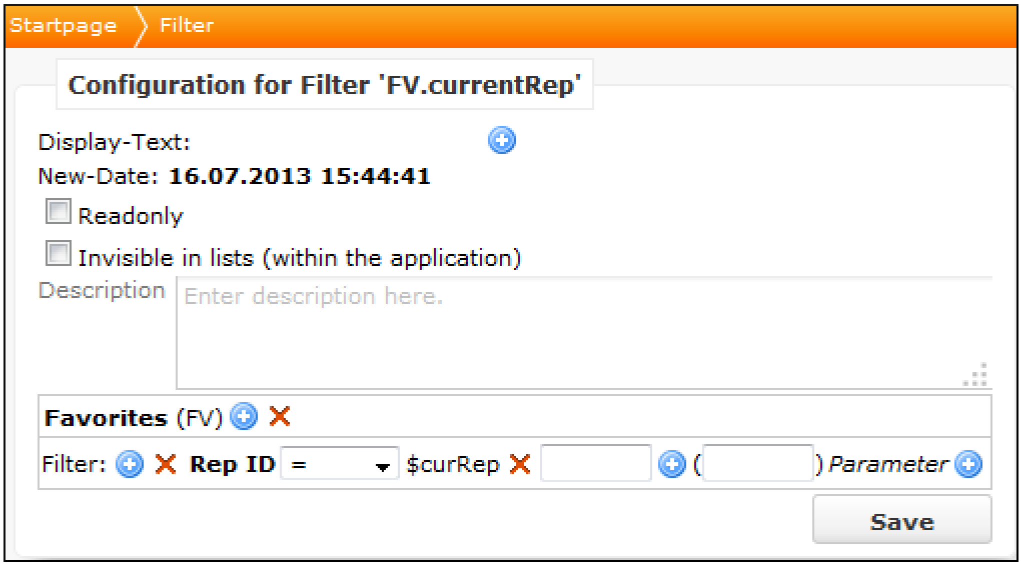
You can ignore the other parameters.

3. Add this button to the desired expand header.
4. Define the following two filters. Use the given names, other names are not recognized by the system.
   - `FV.new`: Defines how the FV record is created, when a user adds a record as a favorite, for example:



     This example defines that the field `Rep-ID` is set to the current rep and the favorite is marked as web favorite.

   - `FV.currentRep`: Defines which of the existing FV records are displayed.



Note: If a record is a favorite but is not marked as Aurea CRM web favorite, it is deleted even if it is marked as another applications favorite, e.g. connector, etc.

Actions

Actions define the happenings in CRM.pad.

Example - whether data is displayed in List or in Details view, whether they are in View or in Edit mode, and much more.

To assign actions, action templates are available. When assigning the actions to buttons or menu actions, you can specify input arguments that define which data is read, how it is read (from the server, locally), which filters are applied to limit the numbers of records, etc.

Global Actions

Global actions are actions that are available in the control center's **Global** area, see Control Center on page 210.

Global actions are defined in the `$GlobalAction` context menu. This menu is included in the `update.tablet` configuration.

If this menu does not exist, no global actions are available.

By default, the `$GlobalAction` context menu contains the following context menu actions:

- `INFO_SETTINGS`: Calls the `SettingsView` action, see SettingsEditView and SettingsView on page 136.

Quick Actions

Quick actions are actions that are available in a record's header (e. g. in a Details view).

Quick actions are not only used by CRM.pad, but also by CRM.mobile. However, CRM.pad only supports actions that:

- can be executed on the device (all other actions are ignored).
- do not contain an opening curly bracket (as they identify contents only supported by CRM.mobile)

To add quick actions, add a special header with the following syntax to the desired info area's header group:

- `<HeaderName>.QuickActions`

  Special headers using this syntax are available for CRM.pad and CRM.mobile. Example: `Expand.QuickActions` adds quick actions to the Expand header group.

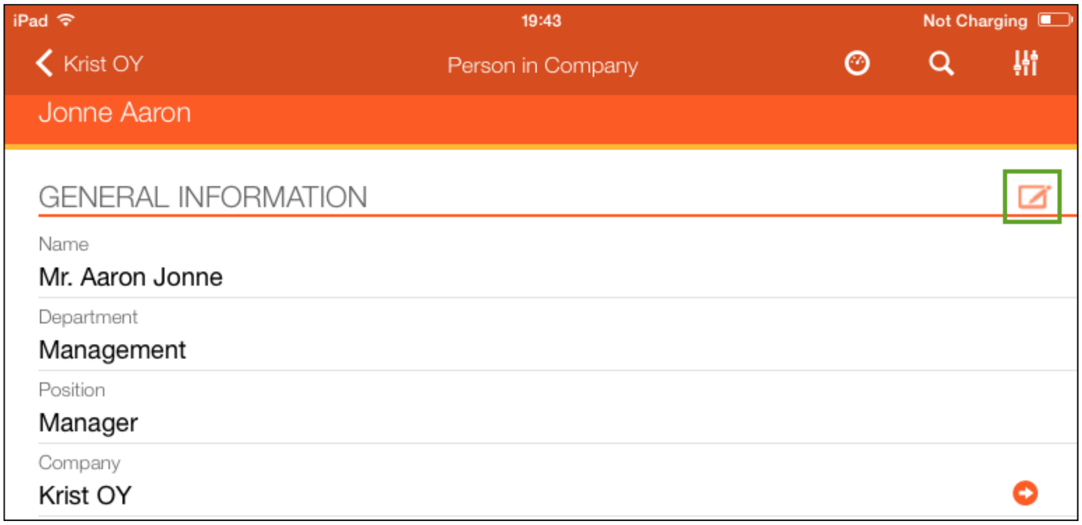
- `<HeaderName>.QuickActions[crmpad]`

  If you use this syntax, the quick actions are only available for CRM.pad.

Note: If no special header for quick actions exist, or the special header exists but the configured buttons cannot be displayed in CRM.pad, the first three buttons assigned to the respective header (e. g. the Expand header) are displayed as quick actions, the other buttons are listed in the **Actions** menu.

Group Actions

Group actions are actions that display a button on the desired panel in Details view allowing the user e. g. to change the content of the panel only.



You can assign group actions as control or panel attributes, see Edit Control Attributes.

Action Templates

Action templates define what is displayed in the application, what actions the user can carry out, etc.

You can assign action templates to e. g. buttons, context menu items etc. You can specify input arguments that define e. g. the info areas called, the filters that are applied and more.

This chapter only describes CRM.pad-specific action templates or CRM.pad-specific input arguments. For more information on action templates, seeAurea CRM web Administrator Guide.

FORM:<Name>

Some action templates exist twice, with and without the `FORM` prefix, e.g. `RecordListView` and `Form :RecordListView` .

Assign the prefixed action template when function parameters are handed over, for example, if you want to call an action template from a Legacy Form (e. g. start page) see Legacy Forms on page 221.

The parameters entered in the **Func** column are handed over in the given order:

Syntax: `( !Func0 ;!Func1;!Func2{…})`

Example:

If the function parameters `(MA; ;MA.Dashboard {…})` are given in `Form:RecordListView`, `MA` is handed over as `ConfigName` and `MA.Dashboard` as `FilterName`.

CONFIGURATTION:<Name>

Action templates with the `CONFIGURATION:` prefix contain pre-defined sub-configurations e. g. for Pricing or Questionnaires.

For example; `CONFIGURATION :PRICING` contains the basic pricing configuration that can be referenced from various action calls (and therefore needs not be defined in the action call separately).

TILE:<Name>

In CRM.pad only the following action templates are supported:

- `TILE:Record`, see Tile:Record on page 138.
- `TILE:Search`, see Tile:Search on page 139.

All other existing `TILE: <Name>` action templates are currently only used in CRM.mobile.

General Input Arguments

List of general input arguments for action templates.

The following input arguments apply to many action templates. The topics describing the action templates contain input arguments specific to them.

Input Arguments	Description
`AdditionalFilter`	Names of additional filters that are applied, separated by comma.
`ConfigName`	Name of the referenced Search&List configuration, fallback: field group of the same name. For client reports you can also define a specific link to be applied with the following syntax: `<ConfigName>#LinkId`
`EnabledFilter`	List of filters that are enabled by default, separated by comma.
`Filter1 - Filter<n>`	Defines the filters that are available to the user when he taps on the (**Filter**) button. The number of available filters depends on the action template.
`FilterName`	The name of an additional fixed filter that is applied.

Input Arguments	Description
`FullTextSearch`	If set to `true`, the full text search is applied. The asterisk (*) can also be used as a placeholder at the beginning of a search criterion.
`hideOnlineOfflineButton`	If set to `true`, the (online/offline) button allowing the user to search data online is not displayed. Only the offline search is available. Default value: `false`
`IgnoreSavedAction`	If set to `true`, the action defined by the `SavedAction` argument is ignored and after adding a new child record, its root record is displayed.
`InfoArea`	Defines the info area for which records are displayed.
`LinkId`	The identification number of the linked record, links the record to the given `LinkId`.
`LinkRecord`	The record linked to the current record.
`ParentLink`	If you do not want to use the default link to the parent record, you can define a specific link. With `NoLink` you can prevent a link being set.
`RecordId`	Defines the record that is the parent or root record, usually the current record (e.g. in edit mode, the record to be edited, and in New mode, the linked parent record, i.e. the record from which the New mode was started). If left empty, the parent record's `RecordId` is applied, e. g. for samples the activity's `RecordId`.
`RepFilter`	Records, e. g. calendar items, can be filtered by reps. Enter the filter name to be applied, e.g. `MA.ParamRep`. Some filters (as `MA.ParamRep`) require parameters.

<table>
<tr><th>Input Arguments</th><th>Description</th></tr>
<tr><td><code>RequestOption</code> or
<code>RequestMode</code></td><td>Defines how data is read. Depending on the action called, not all options are applicable. The default value also depends on the action.
The following options are available:
• <code>Fastest</code>: Data is read offline, if available. If not found offline, the data is searched online if a server connection is available.
• <code>Best</code>: Data is searched for online. If no network connection is available, data is read offline
• <code>Online</code>: Data is always read online (data cannot be displayed if no network connection was established)
• <code>Offline</code>: Data is always read offline only.</td></tr>
<tr><td><code>RightsFilterName</code></td><td>Name of a filter that specifies the conditions the current record has to meet so that an operation is carried out. For example, if a report (i.e. an order) was signed by a customer, it cannot be edited anymore.</td></tr>
<tr><td><code>SearchOptions</code></td><td>Available option:
<code>NoEmptySearch</code>: If the search field is empty, the search is not started.</td></tr>
<tr><td><code>SendByEmail</code></td><td>If set to <code>true</code>, a button to send an e-mail is displayed after the action was executed, e.g. after a client report was generated.
Default value: <code>false</code></td></tr>
<tr><td><code>SendbyEmailAction</code></td><td>Name of a context menu action that generates the e-mail's body based on the current record's data. Transfers data like the e-mail address and the subjects to the e-mail client.
If more than one action (separated by comma) are given, a dialog box is displayed and the user can select the desired action.</td></tr>
<tr><td><code>SendByEmailAttachReport</code></td><td>If set to <code>true</code>, a PDF report based on the record's and its children's data (e. g. an offer and its offer items) is generated and attached to the e-mail.
Default value: <code>false</code></td></tr>
</table>

Input Arguments	Description
`SortSequence`	If set to `ASC` (default), data is sorted in ascending order. `DESC` sorts the data in descending order.
`SwipeDetailRecords`	If set to `true`, the user can swipe to the next or previous record. If left empty, the `View.RecordSwipeEnabledDefault` Web Configuration parameter is applied, see **View.RecordSwipeEnabledDefault** in the *Aurea CRM web Administration guide*.
`TemplateFilter`	Name of a template filter that sets, for example, initial values when a child record is created.

Analysis

Action templateList for analysis.

Displays analysis (that was defined in Aurea CRM web).

Input Argument	Description
Analysis	Name of the executed analysis.
CopySourceFieldGroupName	Name of the field group where the fields you want to copy have been marked accordingly. These fields can be used as filter criteria.
CopySourceRecordId	RecordId where from data is used as filter critera.

Input Argument	Description
AdditionalParameters	Additional parameters that are applied to the query conditions in JSON syntax.
Options	Additional options that can be given in JSON syntax. Example: `{"ShowEmpty" :true }`: By default, empty values (i. e. categories that have no values) are ignored. Set this option to include them in the analysis. `{"defaultChartType": "Pie"}` : Set this value to any of the supported chart types in Aurea CRM.Pad to get desired result. **Note:** The chart type select in `defaultChartType` is selected by default when you open Analysis. Below is the list of supported chart types: • "Bar" • "Horizontal Bar" • "Stacked" • "Pie" • "Lines" • "Stacked Area" • "Lines Zoom" • "Funnel" • "Radar" • "Bar" • "Lines"

CalendarView

This action template displays activities in a calendar (by default).

You can also define that additional company- and person-related data such as absences are displayed in the calendar.

Input Argument	Description
AdditionalCalendar-Configs	A comma-separated list of context menu actions to display additional info areas in the calendar, e. g. absences. These context menu actions must have the `CalendarView` action assigned. You can use this argument to display any company- or person-related info area in the calendar.
Default View Type	Defines the default view when the user opens the calendar. Possible values are: `DAY`, `WEEK`, `MONTH` (default), `LIST`. **Note:** Setting the web configuration parameter `CalendarView.RememberViewType` to true opens the Calender in the view type last used by the user. For more information, see CalendarView.RememberViewType.
CalendarPopOverConfig	The field group where the Details control was defined that is used to display the calendar's pop-up.
IncludeSystemCalendar	Enables the function that displays iPad calendar items in CRM.pad.
NewAppointmentAction	The name of a button. With a long tap on the week or month view, the action assigned to this button is executed (e. g. a new appointment is created).
ShowSystemCalendarFilter	If set to `true`, the global settings are overridden (defined in the Web Configuration parameter, see CalendarIncludeSystemCalendar).
RepFilter	Calendar items can be filtered by reps. In the default configuration, the `MA.ParamRep` filter is applied. If data from additional info areas such as absences are also displayed, all records are displayed by default, see Displaying Absences or other Info Areas in the Calendar on page 262.
RepFilterCurrentRepActive:	If set to `true`, the records are filtered by the current rep.

Characteristics

This action template displays characteristics/interests.

Below is a list of characteristics/interests:

Input Arguments	Description
RecordId	The ID of the root record.
DestinationFieldGroup	The name of the destination record's field group (i.e. the field group used to store data).
SourceFieldControl	Name of the source record's field group.
GroupSearchAndList	Name of the Search&List configuration defined for the item groups.
ItemSearchAndList	Name of the Search&List configuration defined for the items.

CharacteristicsEdit

This action template allows the user to edit characteristics/interests.

Below is a list of input arguments.

Input Argument	Description
DestinationReadFilter	The filter that is used to read the existing destination record's data, e.g. to read only company-related records.
DestinationRequestOption	Defines how the destination data is requested, see `RequestOption` argument in General Input Arguments on page 70.
SourceRequestOption	Defines how the destination data is requested. Possible values: `Offline`, `Online`, `Best`, see `RequestOption` argument in General Input Arguments on page 70.
DestinationFieldGroup	The name of the destination record's field group (i.e. the field group used to store data containing the Group and Item fields for the required catalog values).

Input Argument	Description
SourceFieldControl	Name of the field group where the fields that can be used as parameters in filters (prefixed with `$par`) have been marked accordingly.
GroupSearchAndList	Name of the Search&List configuration referencing the field group (and filter) defining the items to be displayed.
ItemSearchAndList	Name of the Search&List configuration referencing the field group (and filter) defining the records displayed when the user edits the data.
SyncParentInfoAreaID	Comma-separated list of info areas, to include the root record for synchronization, e. g. KP,FI also synchronizes all company and person in company records linked to characteristics.

CircleOfInfluenceTreeView

This action template displays related data in a Circle of Influence.

Below is a list of input arguments:

Input Argument	Description
ConfigName	The name of the referenced Tree view.
SwitchToCoi	If set to `true`, the Circle of Influence is displayed if a user taps on a node or on an item in the relationships list.

CircleOfInfluenceView

This action template displays data from the Relationship info area in a Circle of influence.

Below is a list of input arguments:

Input Argument	Description
AdditionalFIFilter	Name of the filter that loads the child records from the Company (FI) info area.
AdditionalKPFilter	Name of the filter that loads the child records from the Person (KP) info area.

Input Argument	Description
ConfigName	Name of the Search&List configuration that defines the data read as nodes and edge information.
RootNodeConfig-Name	The name of the Search&List configuration for the CoI's root node in the FI and KP info areas. If given, the value in `RootNodeFieldGroup` is overridden.
RootNodeField-Group	The name of the field group defined for the CoI's root node in the FI and KP info areas. As the root node does not yet exist in the PB info area when no connection exists yet, this field group from either the Company (FI) or Person (KP) info area must be given. **Note:** You can only define either the `RootNodeFieldGroup` or the `RootNodeConfigName` parameter, but not both.
SwitchToCoi	If set to `true`, the CoI page is displayed when calling a child node (otherwise, the overview is displayed).
TreeMaxDepth	Defines the maximum number of levels displayed when the Circle of Influence is loaded. Default value: `1`
nodesViewConfigu-ration	Defines styles that differ from the default node and edge styles (JSON syntax). Default value: empty **Note:** Do not define this parameter without consulting with your Aurea representative.
recordId	Defines the root node's record ID, usually the current record.

ClientEmail

This action template allows the user to send an e-mail based on data stored in the database.

Below is a list of input arguments:

InputArgument	Description
EmailField-Group	Name of the field group that contains the e-mail's field definition. For information on the referenced field group, see Defining E-Mail Fields on page 208.

ClientReport

This action template allows the user to create a report for data consisting of a main record (e.g. order) and one or several child records (e.g. order items).

Below is a list of input arguments:

Input Argument	Description
AdditionalConfigPar-entLinks	The link of additional parent links, if needed. Possible values for each parent link: • `<InfoArea>`, e.g. `FI` (uses the default link) • `<InfoArea>:<LINKID>`, e.g. `FI:175` • `nolink`
AppearanceJScript	JavaScript action that is executed when the report is created.
ConfigParentLink	The link of the first parent link.
DissappearanceJScript	JavaScript action that is executed when the report is closed.
HeaderName	Name of the header that is applied for the label and additional pages, if the report is displayed in a new page.
Parameters	The report's parameters, e. g. entered in JSON syntax.

Input Argument	Description
XmlRootElementName	Name of the XML root element. Data is queried in all branches/knots of this root element.
Xsl	The (optional) XSL name to transform the XML to HTML. Xsl items are stored in the **Images** section of CRM.designer.
RootName	The name of the query's XML root element (in most cases this is the info area name).
AdditionalConfigName	A semicolon-separated list of additional Search&List configurations to be searched. You can also define a specific link to be applied with the following syntax: `<ConfigName>#LinkId`
AdditionalRootNames	By default, an order summary is available for the quick add. In the `update.tablet` configuration, this order summary is based on a report. The XML file for this report must be transferred to the device. To include the XML file to the data transfer, it is handled as an image, as no other resources can be transferred. Therefore, an item (for the order summary: `Xsl :OrderSummary`) must exist in the **Images** area of CRM.designer for the XML file.
ReportType	`ClientReport` (fixed value)
RequestOptionDependentOnRoot	Defines that the `RequestOption` for the root record is switched to online, if this record is only available online.
Record	The link record used in the data search for the report.
ButtonPrint	If set to `true`, a print button is displayed.

ClientReportWithAction

This action template allows the user to generate the report and also have the customer sign it (e.g. if the report is an order).

In addition to the input arguments available for the **ClientReport** action template, the following arguments can be specified for `ClientReportWithAction`:

Input Argument	Description
CopySourceFieldGroup	The field group where the fields you want to copy have been marked accordingly.
DocumentUploadConfiguration	Name of a context menu action that calls the `DocumentUpload` action, see DocumentUpload on page 92. In this `DocumentUpload` call, the `DocumentFieldFieldGroup` must be given specifying the field in which the generated document (PDF file) is stored.
SendByEmailAttachementReport	If set to `true`, the report is send as attachment.
SigningConfiguration	See SigningConfig Input Argument.
XmlRootElementName	Name of the XML root element. Data is queried in all branches/knots of this root element.
AdditionalConfigNames	A semicolon-separated list of additional Search&List configurations to be applied.
AdditionalRootNames	By default, an order summary is available for the quick add. In the `update.tablet` configuration, this order summary is based on a report. The XML file for this report must be transferred to the device. To include the XML file to the data transfer, it is handled as an image, as no other resources can be transferred. Therefore, an item (for the order summary: `Xsl :OrderSummary`) must exist in the **Images** area of CRM.designer for the XML file.
ButtonName	Name of the button allowing the user to sign the document.
ButtonShowFilter	Name of the filter defining whether the button given under `ButtonName` is displayed.

Input Argument	Description
ButtonPrint	If set to `true`, a print button is displayed.
BlockActionsUntilComplete	If set to `true`, it disables user interaction while embedding signature in report.

SigningConfig Input Argument

The `SigningConfig` input argument defines that an e-mail can only be sent after the report was signed by the customer, e.g. an order.

Default: `{}`

The following options can be entered in JSON syntax:

- `" sign ":false:` Defines that no signature is required.

  Default: `true`

- `" upload ":false:` Defines that the client report is not uploaded to the database. The report is then only attached to the e-mail.

  Default: `true`

- `emptySignatureName :` A comma-separated list of empty signature images. An empty signature image must be given for each required signature.

  Default: `Button :PleaseSign .`

- `signatureImageTagName :` A comma-separated list of XML tags that references the signature images. A tag must be given for each required signature.

  Default: `SignatureImage.`

- `signatureImageId`: Name of the `img` tag's `id` attribute in the XSL file.

  Default: `img-signature`

- `lineWidth`: Font width

  Default: `3`

- `strokeColor:` Font color

  Default: `0;0;1;1` (= blue)

- `imageWidth`: Width of the image file containing the signature

  Default: `300` (px)

- `signedReportFileName`: Name of the generated PDF file

  Default: `SampleReport.pdf`

- `signedReportFileNameDateFormat:` Date format in case a date parameter is appended to the file name

Default: `yy-mm-dd`

- `signatureTitle` : Name of a field group. This field group's List control is applied and the field containing the `SIGNATURETITLES` function name is searched. This function may contain `{$var}` placeholders that are replaced according to the given `CopySourceFieldGroupName` argument in the `ClientReport` action.

  You can also define that more than one signature must be given. The following example requires two signatures:

  ```
  {"emptySignatureName":"Button:PleaseSign;Button:PleaseSign2","signatureIm-
  ageTagName":"SignatureImage;SignatureImage2","sign": true, "upload":
  true}
  ```

Configuration:Listing

This action template configures the quick add's listing.

Below is a list of input arguments:

Input Argument	Description
ListingControlName	Name of a Search&List configuration where the referenced field group contains a List control that defines which Listing records are linked to a FI record.
ListingOwnerFieldGroup-Name	Name of a Search&List configuration where the referenced field group contains a List control that defines the fields allowing to filter the LS records. The fields are identified by the **Functions** assigned.
RelatedListingOwnersConfig-Name	Name of a Search&List configuration where the referenced field group contains a List control that defines the fields that identify the related FI records to filter the LS records.
DistinctListingFunction-Names	List of function names (separated by comma) that define whether an item is included in a listing.
ApplyHierarchy	Defines which combinations of the given `DistinctListingFunctionNames` must be valid fields so that a listing record is applied to an item.

Input Argument	Description
DestinationFieldFunction-Names	Defines which fields must be the same as in the destination record, so that a destination record is identified as a valid destination record.
ListingRequestOption	Defines how data is requested, see `RequestOption` argument in General Input Arguments on page 70. **Default:** `Fastest`

Configuration:OfflineDocumentCreation

This action template defines how documents are created in offline mode.

See also Offline Upload.

Below is a list of input arguments:

Input Argument	Description
DocumentLinkInfoAreaId	Defines the info area in which the linked documents are stored.
DocumentLinkId	The ID of the linked document.
DocumentLinkTemplate-Filter	Defines the template filter to find the linked document.
DocumentTemplateFilter	Defines the filter for the document template.
RecordLinkId	Defines the link ID of the record containing the docu-ment.

Configuration:Participants

This action template configures the participation options.

Below is a list of input arguments:

Input Argument	Description
LinkParticipantsLinkId	Defines the info area in which the linked participants are stored. Default value: `MB`.
LinkParticipantsRequestOption	Defines how data is requested, see `RequestOption` argument in General Input Arguments on page 70.
LinkParticipantsSearch&ListName	Name of the Search&List configuration referencing the field group involved in the link participants.
RepAcceptanceRequestOption	Defines how data is requested, see `RequestOption` argument in General Input Arguments on page 70.
LinkParticipantsInfoAreaId	The Link ID linking the source info area and the info area given in `LinkParticipantsInfoAreaId`.
RepAcceptanceConfigName	Name of the Search&List configuration defining how the `RepAcceptance` is determined. The List or Edit control of the referenced field group is applied (using the filter specified in the Search&List). The referenced control determines the Rep column and the column containing the `Acceptance` value. See also Participants on page 258. **Note:** If this configuration does not exist, or if no `RepAcceptanceConfigName` is given, the acceptance state for the rep cannot be displayed and, therefore, it cannot be changed.
RepAcceptanceLinkId	Defines the `RepAcceptanceConfigName`. Records that are linked with the source record by this Link ID are read.

Configuration:Pricing

This action template configures the quick add's pricing.

Below is a list of input arguments:

Input Argument	Description
BulkVolumeConfigName	Name of the Search&List configuration defined for bulk volumes.
Options	Currently not implemeted.
PriceListConfigName	Name of the Search&List configuration that references the field group that defines the pricing information in its List control, e.g. `PLPricing` in the `update.tablet` configuration.
ConditionConfigName	Name of a Search&List configuration referencing a filter that defines which records (meeting the filter criteria) are loaded.
ConditionScaleConfigName	Name of a Search&List configuration referencing a field group's List control that defines the scales displayed. Specifying a filter for this Search&List configuration is in most cases not required, as conditions are automatically filtered.
FunctionNameApplyOrder	Defines the order of the price lists and is specified by the assigned functions in the List control of the referenced field group.
CompanyConfigName	Name of the Search&List configuration referencing • the field group where the List control has the required functions assigned (defining the fields that are used as filter criteria) and • the filter that only displays condition records for the current company.

Input Argument	Description
CompanyScaleConfigName	Name of a Search&List configuration referencing a field group's List control that defines the company condition (KD) scales displayed. Specifying a filter for this Search&List configuration is in most cases not required, as conditions are automatically filtered.
ActionConfigName	Name of a Search&List configuration containing a filter that defines which special offer item (AA) records (meeting the filter criteria) are loaded.
ActionScaleConfigName	Name of a Search&List configuration referencing a field group's List control that defines the special offer item (AA) scales displayed. Specifying a filter for this Search&List configuration is in most cases not required, as conditions are automatically filtered.
BundleConfigName	Name of a Search&List configuration referencing a filter that defines which bundle discounts (Item Group (FG) and Volume (FW) info areas) records (meeting the filter criteria) are loaded.
BundleScaleConfigName	Name of a Search&List configuration referencing a field group's List control that defines the bundle discounts (Item Group (FG) and Volume (FW) info areas) scales displayed. Specifying a filter for this Search&List configuration is in most cases not required, as conditions are automatically filtered.
BulkVolumeMatchFunctionNames	List separated by semicolon containing comma-separated function names

Input Argument	Description
BulkVolumeItemNumber	Function name of the source item number
PriceListPriorityColumns	List of **Function** names (separated by comma) that define the order in case that multiple PL records exist for one AR record.

Configuration:Questionnaire

This action template configures the questionnaire.

Below is a list of input arguments:

Input Argument	Description
QuestionnaireSL	Name of the questionnaire's Search&List configuration.
QuestionsSL	Name of the questions' Search&List configuration.
QuestionAnswersSL	Name of the answers' Search&List configuration.

Configuration:Quota

This action template configures the quick add's quota.

Below is a list of input arguments:

Input Argument	Description
DateParameterName	Name of the **Function** defining the sample issue date given in the field group specifiedin`SourceCopyFieldGroup` parameter in the `SerialEntry` action call. This field group must contain a field in the List control where this **Function** was assigned. If the `DateParameterName` is not given, the current date is applied.
NewQuotaTemplateFilter	Name of the template filter used to create new quota (MU) records.

Input Argument	Description
QuotaFieldFunctionNames	**Function** names (separated by semi-colon) defined in the quick add's Edit Control (default: `ItemsSent` and `ItemsIssues`).
QuotaLinkId	You can define that a new MU record is linked with its parent record by a specific link ID. By default, they are lined with Link ID `-1`.
ArticleConfigName	Name of the Search&List configuration that defines which article information is read. The referenced List control must contain fields that have the `ItemNumber` and `Quota` **Functions** assigned.
QuotaConfigName	Name of the Search&List configuration that defines which quota records are read.
MaxSamplesPerPeriod	Number of samples that the user can hand out per time period (year), applicable when no max. quota was defined in the MK records.
MaxPeriods	Max. number of time spans that a user is allowed to hand out the sample.
ParentLink	The quota is calculated for each root (usually a company or person) record. However, if the root record e. g. is an MA record, the quota cannot be calculated as the quota is linked to the person (KP) record.
RequestOption	Defines how data defined under `ArticleConfigName` and `QuotaConfigName` is requested, see `RequestOption` argument in General Input Arguments on page 70. **Default:** `Fastest`

Input Argument	Description
ItemNumberFunctionName	Name of the function given for the Item Number field in the Edit control of the field group referenced by `ArticleConfigName` and `QuotaConfigName` **Default:** `ItemNumber` (= XML name)).
RowItemNumberFunctionName	Name of the function given for the Item Number field in the Edit control of the field group referenced by `ArticleConfigName` and `QuotaConfigName`.
Options	Additional options can be given in JSON syntax: • `{"allowQuotaExceed" :true }` defines whether the quota can be overridden. **Default:** `false` • `{"autocorrectQuota" :true }` defines that if a rep tries to hand out more samples than defined in his quota, the value is automatically set to the (remaining) quota. • `{"showZeroQuota" :true }` defines whether items with no quota left are not shown. **Default:** `false` • `{"hideZeroQuota" :false }` defines Whether items with no quota left are not shown. **Default:** `true`

Input Argument	Description
CalendarYearPeriods	If set to `true`, issue periods are defined according to calendar years. **Note:** This input arguments is required to create the MJ records in offline mode. These records are overwritten by the records created by the server's business logic.
MaxSamplesPerPeriodNoConfiguration	If no MK record exists for an Item Master record, samples cannot be issued. If you enter a value in this input argument, issues can also be issued for these records.

ConflictsView

This action template can be added to the System area to display all uncommitted requests.

The user can either export the requests or delete individual ones (e. g. if they cause the system to crash when switching from offline to online mode).

Below is a list of input arguments:

Input Argument	Description
AllowDeleting	If set to `true`, the user can delete records that caused a conflict (i.e. the **Delete** button is displayed).
ShowAllOfflineRequest	If set to `true`, not only conflicts but also all offline records are shown, i.e. records that have been added in offline mode and have not yet been synchronized.

ContactTimesEditView

This action template allows the user to edit contact times.

Below is a list of input arguments:

Input Argument	Description
SearchList	Name of the Search&List configuration that defines the records to be edited.

DashboardView

This action template displays the dashboard.

Below is a list of input arguments:

Input Argument	Description
ConfigName	Name of the generic form that is displayed.
ConfigTabNr	
HeaderName	Defines the header applied. Syntax: `<headergroupname>.<headername>`

DataSync

This action template allows to call the Synchronization page from any other page in the application.

No input arguments are required.

DebugView

This action template displays the Debug page.

No input arguments are required.

DocumentUpload

This action template allows the user to upload documents to the database.

Below is a list of input arguments:

Input Argument	Description
DocumentFieldField-GroupName	The name of the field group containing a field in its Edit control where the document is stored.
DocumentFieldId	The field's ID where the document is stored (can be given instead of `DocumentFieldFieldGroupName`).
ParentLink	By default, the document is linked to the document's root record. You can specify another parent.
FilterName	Filter that defines whether a document is valid, i.e. the report is displayed. For example, if a report was already signed, but the order is re-opened, the PDF file is still displayed (instead of the report). The given filter determines which documents are displayed as report or as PDF files.

DocumentView

This action template displays the available documents.

Below is a list of input arguments:

Input Argument	Description
FilterName	The name of the default filter to be applied.
HeaderName	Name of the header applied. Can be given for D1 records, must be given for all other document records.

EditView

This action template displays the Edit view allowing the user to edit existing data.

Below is a list of input arguments:

Input Argument	Description
CopySourceFieldGroup-Name	Name of the field group where the fields you want to copy have been marked accordingly.
SavedAction	Defines the action that is executed when the record is saved. Possible values: • `"Button:<buttonName>"` • `"Menu:<contextMenuName>"` • `"Return"` returns to the page that was displayed before the Edit/New view was called.
SyncParentInfoAreaId	Comma-separated list of parent info areas to be synchro-nized. For example, if changes in parent records are caused by a trigger, these changes are only synchronized when the info areas are given in this argument.
ExpandName	Name of the Expand configuration to be applied.
CopyRecordId	Record ID to determine the copy fields defined in the field group given in `CopySourceFieldGroup`. This value can differ from the value given under `LinkRecord` when a new record is created.
RefreshFilterTokens	If set to `true`, Filter Tokens that were defined with the `RepCopySearchAndList` Web Configuration parameter are refreshed, see RepCopySearchAndLists.
Options	Additional options can be given in JSON syntax. Example: Link fields can be set locally, e. g. if linked records are added in offline mode, see System.OfflineStationNumber. If you specify `{"ComputeLinks" :true }` in argument, link record IDs are generated for the server records.

Input Argument	Description
ErrorTranslation	List of field groups separated by semicolon containing alternative error messages, see Defining Error Messages on page 375.
ClientRightsFilterName	Name of the filter applied when editing a record, e.g. you can define that MA records cannot be created in the past. If not given, `ClientFilterForUpdate` is applied instead (and if this is not given, `ClientFilter` is applied as fall-back).

EditViewFromSpecificRecord

You can apply this action template in case you want to provide the `RecordId` as value (instead of `Record` or `LinkedRecord` as in the `EditView` action template).

Below is a list of input arguments:

Input Argument	Description
ExpandName	Name of the Expand configuration to be applied.

ExecuteTrigger

Use this action template to execute a trigger.

Below is a list of input arguments:

Input Argument	Description
TriggerName	Name of the trigger as defined in Aurea CRM win.

ExecuteWorkflow

Use this action template to execute a workflow.

Below is a list of input arguments:

Input Argument	Description
ExecutionFlags	Defines the execution flags. You can combine the following values separated by |: • `Transaction`: Executes inside a transaction. • `Rollback`: Rollback, if aborted / executed with error execution. • `ReturnRecords`: Returns the changed records. • `ReturnParameters`: Returns the (workflow data) parameters after execution. Example: `ReturnRecords | ReturnParameters`
Parameters	An array of parameters given in JSON syntax: Example: `{"Id":0,"Name":"Param1","Values":[]},{"Id":1,"Name":"Param2", "Values":["bla25"]},{"Id":2,"Name":"Param3","Values":["bla25","bla25"]}]`
WorkflowName	Name of the workflow as defined in Aurea CRM win.

FileUploadAction

Use this action template to link a document from the Document Inbox to a record.

Below is a list of input arguments:

Input Argument	Description
Action	Fixed value: `uploadFile`
UploadFields	Name of an Edit control in the D3 info area. If given, the user can edit the fields defined in this Edit control before uploading a document. If a field was assigned the `Filename` function, the file's name is automatically entered in this field.

Input Argument	Description
RemoveUploadedFile	If set to `true`, the uploaded file is removed from the Document Inbox.
SkipUploadPageIfPossi-ble	If set to `true`, the second upload page of the upload pro-cess is not displayed if no Edit control was given under `UploadFields`.

FormControl

This action template defines a `FormControl` (like Legacy Forms).

For more information see Legacy Forms on page 221.

Below is a list of input arguments:

Input Argument	Description
ControlType	Defines the type applied to a Legacy Form row.
Func	Functions (separated by semicolon) that can be given in the **Func** field of a Legacy Form row.
ItemAtrributes	Item attributes that can be given in the `Item-Attribute` field of a Legacy Form row.
Options	Additional options can be given in JSON syntax.
Func0 - Func4	Name of a function to be applied. You **cannot** specify both the `Func` and the `Func0` - `Func4` input arguments.

FormView

This action template displays a generic form.

Below is a list of input arguments:

Input Argument	Description
configName	Name of the generic form applied.
Options	Additional options can be given in JSON syntax.
Record	See `RecordId` argument in see General Input Arguments on page 70. **Note:** This arguments is called `Record` (instead of `RecordId`) for compatibility purposes.

GeoSearch

You can use this action template to define the vicinity search.

Below is a list of input arguments:

Input Argument	Description

AdditionalParameters	Additional parameters for GPS coordinates given in JSON syntax.
	
	Examples:
	
	`{"GPSX":"$GPSX$","GPSY":"$GPSY$"}`
	
	If you want to include address data as fallback for your current location, in case no GPS data is available:
	
	`{"GPSX":"$GPSX$","GPSY":"$GPSY$","GpsCity":"$GpsCity$", "GpsStreet":"$GpsStreet$","GpsCountry":"$GpsCountry$"}`
	
	The `CopyFieldRecord` and `CopyFieldGroup` input parameter override the parameters defined in this input argument.

Config1Filter-Config3Filter	Name of a Location filter, see Location Filters. Must be specified if the `ConfigName` is given.
Config1Name - Config3Name	Name of a Search&List configuration that defines the vicinity search, see Vicinity Search.
Modus	`GeoSearch` (fixed Value).

Input Argument	Description
CopyFieldRecord	The record for which the initial location is calculated using the field group given in `CopyFieldGroup`.
CopyFieldGroup	Name of the field group where the fields you want to copy were marked accordingly by defining their **Function** names. The given field group's List control is used to determine the initial location. **Note:** The defined **Function** names in this field group override (if given) the defined `AdditionalParameters` (`GpsX`, `GpsY`, `GpsStreet`, `GpsCountry`, `GpsCity`).
CopyFieldRecord	The record for which the initial location is calculated using the field group given in `CopyFieldGroup`.
DefaultRadius	Defines the default radius in meters for the vicinity search.

GlobalSearch

This action template displays the global search defined by the Quick Search.

For further details, see Searching Several Info Areas (Global Search).

Below is a list of input arguments:

Input Argument	Description
ConfigName	Name of the applied QuickSearch configurations, **currently not supported in CRM.pad**.
Modus	`GlobalSearch` (fixed value)
MinSearch-TextLength	If defined, the given criterion is only searched if the user enters at least the specified number of characters.

HistoryListView

You can use this action template to display the history (i.e. all the changes) of a record.

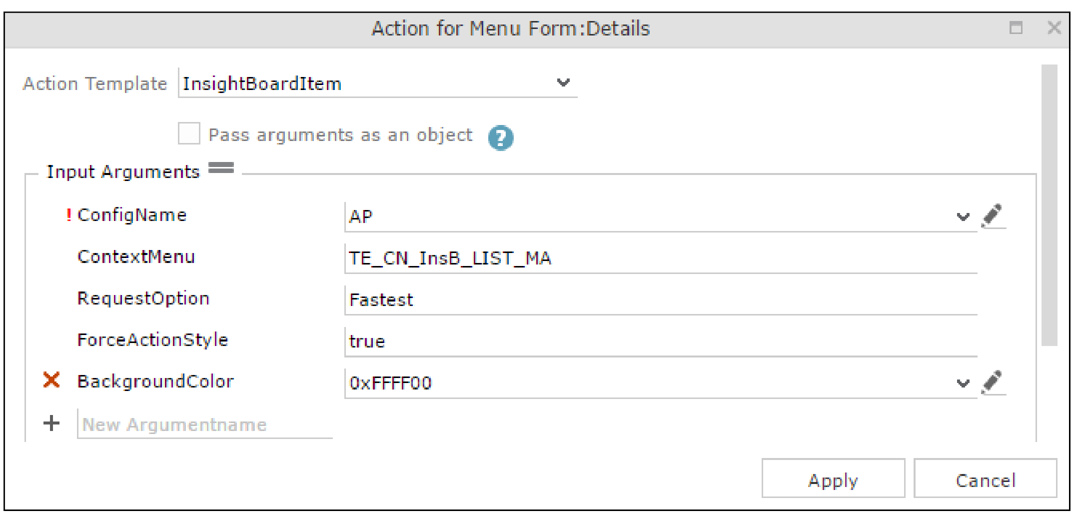
Below is a list of input arguments:

Input Argument	Description
HeaderName	The name of the special header to be applied. Syntax: `<HeaderName>.<SpecialHeaderName>` Example: `History.History`
InfoAreaFilter	Filter that is applied in JSON syntax, e.g. `{"filter":["FI","KP"]}` If you do not specify a filter, the history is displayed for **all** info areas.
MaxNumber-HistroyRecords	Defines the max. number of history records displayed.

InsightBoardItem

This action template displays the Insight Board on Details pages.

Below is a list of input arguments:



Input Argument	Description
ConfigName	The name of the default tiles' Search&List configuration. Filters limiting the number of hits can be referenced in the Search&List configuration.
ContextMenu	Name of an alternative context menu action. If given, the default configuration is overridden. Must be given for Insight Boards that are displayed on the login page, see Insight Board on page 263.
RequestOption	Defines how data is requested, see `RequestOption` argument in General Input Arguments on page 70. Default value: `Offline`
ForceActionStyle	Defines that instead of a number an icon is displayed on the tile. Possible values: `true` | `false` (default)
BackgroundColor	Allows you to provide a background color to the Insight Board item. For example, the value can be set to `0xFFFF00`.

ImageView

This action template displays an image on a separate page.

Below is a list of input arguments:

Input Argument	Description
ImageName	The name the image was registered with in the **Image** area of CRM.designer.

LogfileView

This action template allows to call the Log page from any other page in the application.

It does not require arguments.

MessageBox

Use this action template to display a message box for specific occurrences.

Below is a list of input arguments:

Input Argument	Description
TextGroupName	The name of the text group containing the message text. **Note:** Text groups you add to the `update.tablet` configuration must start with the `smartbook:` prefix. However, when you reference a text group in this input argument, omit the `smartbook:` prefix.
TextNr	The message text's number within the text group.
DefaultText	Optional language-independent default text that is displayed if no specific message text was defined.
TitleTextGroupName	The name of the text group containing the title's text.
TitleTextNr	The title text's number within the text group.
TitleDefaultText	Optional language-independent default text that is displayed if no specific title text was defined.

ModifyRecord

This action template is assigned to buttons or to apply pre-defined changes.

Below is a list of input arguments:

Input Argument	Description
RightsFilter	See the `RightsFilterName` input argument, see General Input Arguments on page 70.
TemplateFilter	Name of the Template filter that defines the applied changes.
LinkRecordId	See >> the `LinkRecord` input argument in General Input Arguments on page 70.
SavedAction	Defines the action that is executed when the record is saved. Possible values: • `"Button:<buttonName>"` • `"Menu:<contextMenuName>"`
RecordIdForSavedAction	An alternative record that is handed over to the defined `SavedAction`.
CopySourceFieldGroup-Name	The field group where the fields you want to copy have been marked accordingly. **Only available in CRM.pad 2.x**
Options	Additional options can be given in JSON syntax. Example: Link fields can be set locally, e. g. if linked records are added in offline mode, see System.OfflineStationNumber. If you specify `{"ComputeLinks" :true }` in the `Options` argument, link record IDs are generated for the server records.

Note: If you use the `ModifyRecord` action, the `Copy` functions fields can also be applied:For example, if assigned in the Person info area, the `CopyCountry` function copies the company name to the FreeC2 field. However, you must adjust the field length. By default, only the first 20 characters of the company name are copied.

MultiSearch

This action template combines several search results in a single record list, e. g. Favorites (i. e. the FI and KP records defined as Favorites are displayed in the same list).

Below is a list of input arguments:

Input Argument	Description
Config1Name - Config3Name	Name of the referenced Search&List configurations, fallback: field group of the same name.
Config1Filter-Config3Filter	The names of additional fixed filters applied to limit the number of hits.
Modus	`MultiSearch` (fixed value).

NewInBackground

This action template creates a new record in the background without user interaction, e.g. you could define that when an order is created, a delivery date record is automatically created with "current day + 2 days", if no specific delivery date was given.

Below is a list of input arguments:

Input Argument	Description
InfoAreaId	Name of the new record's info area.
LinkRecordId	Optional link record of the new record.
TemplateFilter	Filter that defines the values that are automatically applied upon creation.
CheckExisting	If set to `true`, the new record is only created if no record was yet created containing the `TemplateFilter's` values. Default: `false`
CreatedAction	Name of a context menu action that is executed after a record was created.

Input Argument	Description
SyncParentInfoArea	Comma-separated list of info areas that are linked to the new record and that are synchronized as well, e.g. entering MB synchronizes all linked MB records, if an MA record was created.
CreateFailureAction	If the record could not be created, this action is executed.
ExistsAction	Name of a context menu action that is executed if a record containing values based on the `TemplateFilter` already exists. If given, `CheckExisting` needs not to be set.
Options	Additional options can be given in JSON syntax. Example: Link fields can be set locally e. g. if linked records are added in offline mode, see System.OfflineStationNumber. If you specify `{"ComputeLinks" :true }` in the `Options` argument, link record IDs are generated for the server records.
CopySourceFieldGroup-Name	The field group where the fields you want to copy have been marked accordingly.

NewOrEditView

This action template can be applied in specific cases. If a child records exists, the Edit mode is applied, if no record exists, the New mode is applied.

Below is a list of input arguments:

Input Argument	Description
Mode	`EditOrNew` (fixed value)
LinkRecordId	ID of the linked record.
LinkRecord	The record linked to the current record.
NewOrEditViewQuery	Name of the query to be executed.

NewView

This action template displays the New view allowing the user to enter data for a new record.

Below is a list of input arguments:

Input Argument	Description
`AdditionalParameters`	Example: You can define a JSON that adds a value to the copy field parameters (i.e. the fields must have the corresponding functions assigned in the field groups). Example: `{"Date":"$Date$", "Time":"$Time$"}` When the user wants to add a new appointment in the calendar by tapping on the desired date and time and, the variables in the JSON are replaced.
`LinkId2`	The LinkId to the second linked record, e.g. an MA record that is linked to a FI and AU record.
`LinkRecordId2`	The LinkRecordId to the second linked record.
`Mode`	`New` (fixed value).
`SyncParentInfoArea`	Comma-separated list of parent info areas that are also synchronized. For example, if changes in parent records are caused by a trigger, these changes are only synchronized when the info areas are given in this argument.
`LinkRecordId`	ID of the linked record.
`TemplateFilterName`	Filter that defines the initial or fixed values.
`CopySourceFieldGroupName`	The field group where the fields you want to copy have been marked accordingly.
`CopySourceFieldGroupFromOnline`	When you use the `CopySouceFieldGroupName` parameter (described above), the field values are copied from the offline records (by default). To ensure that the field values are copied from the online record, set this parameter to `true`.
`CopyRecordId`	Defines the record for `CopySourceFieldGroupName` (if `LinkRecordId` was left empty or another than the given record should be taken into account).

Input Argument	Description
ErrorTranslation	List of field groups (separated by semicolon) containing alternative error messages, >> Defining Error Messages on page 375.
ClientRightsFilterName	Name of the filter applied when editing a record, e.g. you can define that MA records cannot be created in the past. If not given, ClientFilterForNew is applied instead (and if this is not given, ClientFilter is applied as fallback).
Options	Additional options can be given in JSON syntax. Example: Link fields can be set locally, e. g. if linked records are added in offline mode, >> System.OfflineStationNumber. If you specify {"ComputeLinks":true} in the Options argument, link record IDs are generated for the server records.
keyboardWithScannedSuggestions	Set this option to CONTROLS to use the QR-Code scanner.
SubRecordTemplateFilterName	Template Filter name that defines whether a new parent record and it's child records are created automatically based on a filter's criteria. **Note:** This input argument is not yet included in the update_tablet configuration, therefore, copy the action template to your configuration, add the input argument manually and assign the Filter - optional type.

ObjectivesEditView

This action templates allows the user to edit objectives (AF).

By default, two pre-defined groups (Individual and Sales) are included in the update.tablet configuration.

Below is a list of input arguments:

Input Argument	Description
AdditionalDestinationField-Groups	You can add groups (in addition to the two pre-defined ones) by specifying them in this input argument (separated by comma).
AdditionalDestinationFilter	Name of an additional filter that limits the number of the existing destination records displayed.
AdditionalSourceFieldCon-trols	Names of the field groups (comma-separated) containing the List controls for the groups specified in the `AdditionalDestinationFieldGroups` input argument.
DestinationFieldGroup-NameSales	Name of the field group where the records for the Sales group read from.
DestinationFilterIndividual	Name of a filter that limits the number of the existing Individual group's records displayed.
DestinationFilterSales	Name of a filter that limits the number of the existing Sales group's records displayed.
SourceFieldControlIndividu-al	Field group name where the List control is applied to display the Individual group's data.
SourceFieldControlSales	Field group name where the List control is applied to display the Sales group's data.

OpenMap

This action template opens a map and shows the position of the transferred geo coordinates.

Below is a list of input arguments:

Input Argument	Description
geo	If specified, the values are used as geo coordinates and are passed to the external maps application.
address	The specified address is used to query the external maps application if no geo coordinates were given.
displayName	Defines the name that is displayed on the map indicating the locations found (if the geo coordinates were given).

OpenUrl

Use this action template if you want to define that URLs are handed-over including the current record as a parameter.

Below is a list of input arguments:

InputArgument	Description
FieldGroup	Name of the field group were the required functions were assigned in the List control, or the Search & List configuration that references this field group.
RecordId	The current record (`Record`).

InputArgument	Description
encoding	The encoding for the selected parameters. Possible values: `utf8 (default), ascii, nextstep, japanese, isolation1, symbol, nonlossyasci, shiftjis, isolatin2, unicode, windowscp1250, windowscp1251, windowscp1252, windowscp1253, windowscp1254, iso2022jps, macosroman, utf16`

url	The URL to be called. You can add the current rep' name to an URL, e.g.
	`http://www.google.at/$curRep;`
	The URL can also contain optional parameters that are mapped to a field in the field group given under `FieldGroup`.
	Syntax: `{$<NAME>}`
	Examples:
	• `{$town}` is replaced with the value of the field that has the `town` function assigned in the List control, e. g.:
	`yelp://search?location={$town}`
	• `{recid0}` is replaced with the root record's ID of the referenced field group. If more than one parent root record exist in a row, you can specify the desired record by specifying the sequential number, e. g. `{recid3}`.Passing parameters are supported.

Note: `OpenUrl` is supported in `RecordListView` under list selection action.

OrganizerAction

In some cases, you can assign the `OrganizerAction` action template and define the `Action` input argument to determine which functions are carried out.

Below is a list of input arguments:

Input Argument	Description
Action	Defines which functions are executed. The following actions are available: • `Edit`: Enables the edit mode. • `Delete`: To delete existing records. • `switchToEditSerialEntry`: Enables the quick add's Edit mode. • `ModifyRecord`: To apply pre-defined changes to a record. • `RecordSwitch`: Defines the follow-up action to be executed. Example: A report was signed and cannot be changed anymore. If you send this report by e-mail, the report is generated anew (possibly containing different data then the signed one). You can prevent this behavior using the `RecordSwitch` action. • `toggleFavorite`: Adds/removes a record to the Favorite list.
ContextMenu-Action	Name of a button that has the desired context menu action as-signed.
TemplateFilter	Name of the filter that defines pre-defined values that are applied.

ParentRecordView

This action template displays a record's parent when assigned to a context menu action.

Below is a list of input arguments:

Input Argument	Description
ConfigName	Name of the Expand configuration applied.
InfoAreaId	The info area's ID.
LinkRecordId	The ID of the linked record.

PhotoUploadAction

You can define that users can upload photos (e.g. taken with the device) by assigning this action template.

Below is a list of input arguments:

InputArgument	Description
Config	See Config Input Argument on page 112
Action	`uploadPhoto` (fixed value)
saveToCamer-aRoll	If set to `false`, photos taken with the CRM.pad photo upload are not automatically stored in the iPad camera roll. This input argument is not included in the default configuration. If needed, add it to your configuration.

Config Input Argument

Define the options for the photo upload in JSON syntax:

- `compression`: The JPEG compression rate. A float value between 0 (bad quality) and 1 (best quality).

  Default: `0.8`
- `fileNamePattern`: File name, use the `%1` placeholder for the `RecordId`.

  Default: `photo.jpg`.
- `textPattern`: The options for the selected size. You can use the following placeholders:
  - `%1`: Contains the file name
  - `%2`: Width in pixel
  - `%3`: Height in pixel
  - `%4`: File size

  Default: `%1 (%2 * %3, %4)`
- `sizes` (int array): Determines the image side that is larger. This value is used to scale the image. The user can select the desired size when uploading the picture.

  Default: `[320 ,640,1280,0 ]`: Max. four values possible:
  - `320`: the longer side is scaled to 320 pixel if larger
  - `640`: the longer side is scaled to 640 pixel if larger
  - `1280`: the longer side is scaled to 1280 pixel if larger
  - `0`: the current size is applied.
- `skipOne true|false`: Defines whether the selected size can be exceeded, if only one option is available.

  Default: `false`
- `imageNameEditable true | false`: Defines whether the user can enter in the image's name.

Default: `false`

Example:

```
{"compression": 0.8, "sizes": [320 ,640,1280,0 ], "fileNamePattern": "pho-
to.jpg", "skipOne": false, "textPattern": "%1 (%2 * %3, %4)", "imageNameEd-
itable": false}
```

If you do not enter a JSON script, default values are applied.

QlikView

This action template calls CRM.intelligence (powered by QlikView), an external application to analyze client-related data.

Below is a list of input arguments:

Input Argument	Description
DocumentName	The name of the QlikView document on the QlikView server.
HeaderName	Name of the special header that is displayed in this page's header.
ReportType	`QlikView` (fixed value)
ServerName	URL to the QlikView server
UserLogin	QlikView prefix that is added to each user name, e.g. `Custom\<user name>`. **Note:** You need a license for each user, otherwise users are not able to login automatically.
ConnectionType	Defines how the redirect URL is called. Possible Values: `http` (default) and `https`

Query

This action template displays query that was defined in Aurea CRM web.

Below is a list of input arguments:

Note: This action template is available in CRM.pad 2.1.1. It is not yet included in Version 2.1.

Input Argument	Description
Query	Name of the executed query.
CopySourceFieldGroup-Name	Name of the field group where the fields you want to copy have been marked accordingly. These fields can be used as filter criteria.
CopySourceRecordId	RecordId of the record from which data is applied as filter criteria.
AdditionalParameters	Additional parameters that are applied to the query conditions in JSON syntax.
Options	Additional options that can be given in JSON syntax.

Output Argument	Description
Options	Additional options that can be given in JSON syntax.

QuestionnaireEditView

This action template allows the user to edit a questionnaire (i.e. to fill in the answers).

Below is a list of input arguments:

Input Argument	Description
DestinationRequestOption	Defines how data is requested, see `RequestOption` argument in General Input Arguments on page 70.
SourceRequestOption	Defines how data is requested, see `RequestOption` argument in General Input Arguments on page 70.
QuestionnaireLinkName	Not applicable in Aurea CRM, for compatibility purposes only.
SourceCopyFieldGroupName	Name of the field group from where the source fields are read (= where the functions have been assigned), e.g. `QuestionnaireId` and `SurveyDate` from the MA record.

Input Argument	Description
SurveySearchAndListName	Name of the Search&List configuration referencing the field group containing the `Questionnaire` and `Date` fields in the U1 info area.
SurveyTemplateFilter	Name of the filter defining the key field for the U1 record. If no questionnaire record can be found locally (e.g. if an MA record was created locally), it is created on the device. However, the local U1 record must not be synchronized, the record created by the server must be used. By defining the key field, the server does not create a new questionnaire if the key field and the `LinkRecords` (FI and FI/KP) match.
SurveyAnswerSearchAndListName	Name of the Search&List configuration referencing the field group containing question (U2) fields.
SurveyAnswerTemplateFilter	Template filter defining initial values for the survey's answers.
ConfirmButtonName	Defines the button displayed in the application. No button is displayed if a non-existing button is given. In the `update.tablet` configuration, this is the `ConfirmSurvey` button that has the `SaveAndConfirm` action assigned.
ConfirmedFilterName	Defines the records (e. g. surveys, reports) that are confirmed/finalized. The user can only edit data that does **not** fulfill these conditions (i.e. the Confirm/Finalize button is still displayed). Records meeting the conditions are displayed read-only.

Input Argument	Description
ReadOnly	If set to `true`, the questionnaire is always displayed in read-only mode, the user cannot edit it. Default value: `false`
Portfolio	If set to `true`, a portfolio is displayed instead of a questionnaire (if a portfolio was defined in Aurea CRM web). Default value: `false` If set to `true`, a portfolio is displayed in the record's Details page.

QuestionnaireView

This action template displays a questionnaire (if it was assigned to a MA record).

Below is a list of input arguments:

Input Argument	Description
DestinationRequestOption	Defines how data is requested, see `RequestOption` argument in General Input Arguments on page 70.
QuestionnaireLinkName	Not applicable anymore in Aurea CRM.
SourceCopyFieldGroupName	Name of the field group from which the source fields are read, i.e. where the functions have been assigned, e.g. `QuestionnaireId` and `SurveyDate` from the MA record.
SourceRequestOption	Defines how data is requested, see `RequestOption` argument in General Input Arguments on page 70.
SurveyAnswerSearchAndList-Name	Name of the Search&List configuration referencing the field group containing question (U2) fields.
SurveyAnswerTemplateFilter	Template filter defining initial values for the survey's answers.
SurveySearchAndListName	Name of the Search&List configuration referencing the field group applied to the survey.

Input Argument	Description
SurveyTemplateFilter	Name of the filter defining the key field for the U1 record. If no questionnaire record was found locally (e.g. if a MA record was created locally), it is created on the device. However, the local U1 record must not be synchronized, the record created by the server must be used. By defining the key field, the server does not create a new questionnaire if the key field and the `LinkRecords` (FI and FI/KP) match.
QuestionnaireReadOnly	If set to `true`, the questionnaire cannot be edited anymore.

RecordListView

This action template displays records in List view.

Below is a list of input arguments:

Input Argument	Description
Enabled Filter	Comma-separated list of filters that are enabled by default. **Note:** Do not enter filters that contain parameters.
ScanMode	If set to `true`, the user can scan items with a barcode scanner. Default value: `false`.

Input Argument	Description
SearchPageMode	Defines the mode that is applied if an alternate Expand configuration (or a configuration that is different from the info area name) exists where the colors for the bar left of a search result item is defined. Possible values: • `&1` (default): For example, it is not checked, whether the search result contains items of a virtual info area. • `&2`: Displays the colored bar for the default Expand configuration as well. • `&4`: Displays the colored bar for virtual info areas as well, if the Expand configuration's name is the same as the virtual info area's name. • `&8` (ignore colors): No colored bars are displayed. To combine values, add the values (e. g. `&3` for `&1` plus `&2`).
SwipeDetailRecords	If set to `true`, the user can swipe from one record to the next when this action is executed. **Note:** You can also define that the swipe function is globallyenabledbysettingthe`View.RecordSwipeEnabled-Default` Web configuration parameter, see View.RecordSwipeEnabledDefault.
Sections	If this argument is empty or set to `false`, items in (company and person) lists are not grouped. If set to any other value, list items are grouped based on the list item's first letter. An index is displayed on the left side of the list. If the user taps on the desired letter, records starting with that letter are displayed.
CalendarView	Obsolete, for compatibility purposes only. Use CalendarView on page 74 instead.
CalendarPopOverConfiguration	Obsolete, for compatibility purposes only. Use CalendarView on page 74 instead.
MaxResults	Max. number of records displayed in list.

Input Argument	Description
DetailsActionSwitchFilter	Name of the filter that defines the quick action applied.
MapView	If set to `true`, geo coordinates of address data is displayed in the map (if the geo coordinate fields were added to the referenced List control).
ListStyle	Defines the style how a list is displayed, see ListStyle Input Argument.

ListStyle Input Argument

Note: The `ListStyle` Input Argument only applies to CRM.pad version 2.x.

You can define with the `ListStyle` argument, how a list is displayed. The following options are available:

Input Argument	Description
`row`	Each record is displayed in a row.
`rowOnly`	Each record is displayed in a row. Cannot be changed.
`card`	Two records are displayed in portrait mode or three records as cards in landscape mode in a row.
`card23`	Two records are displayed in portrait mode or three records as cards in landscape mode in a row. Cannot be changed.
`card2Only`	Two records are shown as cards in a row. Cannot be changed.
`classic`	Former CRM.pad 1.x list style. Cannot be changed.

The following only applies if you have the `row` and `card` value is assigned:

- The user can switch from row to card style with a pan gesture.
- In landscape mode only: If no value is specified, `row` is default behavior. The default behaviour can be overwritten by the `List.ClassicLayout` Web Configuration parameter, see List.ClassicLayout.

Showing All Annotations in a Map

By default, CRM.Pad does not load all annotations in a Map view. It only displays those items that are viewed in the result list view pane. For example, if a search result list view for a Company infoarea search has 100 Companies in the search result list and the user scrolled and viewed only 40 companies in the search result list, then only those 40 company locations are annotated and displayed in the Map view.

To annotate and display all the items in the result list view, set the option `ShowAllAnnotationInMap` to true in the **RecordListView** action template. See the screenshot below:

Aurea recommends that you do not display all annotations in a map, if you do not have a set up to provide the latitude and longitude geographic coordinates for the searched items. See, Adding a Map on page 171 for more details.

Without such a setup, CRM.Pad uses the reverse geocoder to deduce the latitude and longitude based on the address. The geocoder API has a limitation of 50 requests per minute. This is usually not sufficient to load all annotations for the search result items, in a map.

RecordSelect

This action template allows you to select data from another info area and "copy" them to the current info area.

You can use the `RecordSelect` action template for record selectors of type `record`.

Below is a list of input arguments:

Note: The `RecordSelect` action replaces the JSON script formerly used. Do not use the JSON script anymore.

Input Argument	Description
LinkRecord	The info area from which field values are copied.
TargetLinkInfoAreaId	Info area of the record that is linked with the record selector. You can also specify `NoLink`, e. g. if you use a record selector in several info areas.
TargetLinkId	The `LinkId` set by the record selector.
ClearValues	JSON array with function names to define the fields whose values are cleared before a new link is assigned.
FixedValues	JSON dictionary containing fixed values to be applied when the record selector is applied. Syntax: `{"FunctionName":"Value"}`
TargetPrefix	Prefix for the function names defined in the target configuration (`ConfigName`). This prefix is combined with the identifiers (**Function** field) specified for the source copy fields in the target field group.
LinkId	The given `LinkId` is used for the record selector's Search&List configuration.

Input Argument	Description
FormattedParameters	You can define certain format parameters (using JSON syntax). Examples: • If the source List control of the record selector contains fields with the `FirstName` and `LastName` **Functions** assigned, you can enter `{"FullName":"%FirstName %LastName"}`. This allows you to assign a function called `FullName` in the desired target Edit control resulting in first and last name being copied to the target record. • If you enter `{"FullName":"my :2 "}`, the text no. 2 from the `smarbook:my` textgroup is applied (i.e. a language-dependent string is displayed).
HideStandardFilter	Defines whether default filters (e. g. recently used) are displayed. Default value: `false`
DisableLinkOption	If set to `true`, a filter button allowing the user to show all records is displayed.
IgnoreFieldInfo	If set to `true`, the read-only flag is ignored.

RecordSwitch

If a report was generated and added to the database, and the user taps the button that creates the report again, you can define with this action that the report stored in the database is displayed instead of the report being created anew.

Below is a list of input arguments:

Input Argument	Description
existsAction	Name of the action executed if the record exists.
Link	LinkId for the `existsAction` and `NotExistsAction` input arguments, either `InfoareaID` (e.g. FI) or `Infoarea#LinkId` (e.g. FI#4).

Input Argument	Description
NotExistsAction	Name of the action executed if the record does not exist.
DecisionFilterName	Name of the Decision Filter based on which different actions calls are applied. Conditions are searched in the following order: `Action`, `ButtonAction`, `DefaultAction`, `DefaultButtonAction`

RecordView

This action template displays a record (Details view).

Below is a list of input arguments:

Input Argument	Description
SyncParentInfoAreaId	Comma-separated list of parent info areas to be included in the synchronization process. For example, if changes in parent records are caused by a trigger, these changes are only synchronized when the info areas are given in this argument.
OrganizerRefresh	If set to `true`, a Details organizer (that is open in the background) is refreshed when it is accessed anew. **Note:** If you set this option, ensure that the `DefaultAction` is set to `ShowRecord`.

RecordViewSpecific

This action template displays a record that has been specified by the `RecordId`.

Below is a list of input arguments:

Input Argument	Description
ConfigName	The name of the configuration applied.
RecordId	Id of the record.

Report

Users can generate a report with this action template, e.g. for an order containing the ordered items.

Below is a list of input arguments:

Input Argument	Description
AppearanceJScript	JavaScript action that is executed when the report is created.
Disappearance-JScript	JavaScript action that is executed when the report is closed.
Headername	Name of the header applied.
LoadManually	If set to `true`, the HTTP request handling is applied manually. **Note:** Aurea software recommends not to change the default setting of this parameter.
ReportType	`CoreReport` (fixed value).
Report	The report's name.
Parameters	An array of report parameters given in JSON syntax.

ReportWithAction

This action template generates a report allowing the user to carry out an action.

Below is a list of input arguments:

Input Argument	Description
signingConfig	SigningConfig Input Argument.
Parameters	An array of report parameters given in JSON syntax.
ButtonName	Name of the button that has the desired action assigned.

Input Argument	Description
ButtonShowFilter	Name of the filter defining whether the button given under `ButtonName` is displayed.
ReportType	`CoreReport` (fixed value)

RequestForChange

Learn about the use of RequestForchange action template.

The default LSI/CG industry solution does not allow users to add or edit company or person records. They must request the back office to do so. This action template allows requesting these changes.

Below is a list of input arguments:

Input Argument	Description
CopySourceFieldGroup-Name	The field group where the fields you want to copy have been marked accordingly.
ExpandName	Name of the Expand configuration for the Request For Change info area.
Mode	`RequestForChange` (fixed value), must not be modified.
R4CInfoAreaId	Name of the Request for Change info area.
R4CRootInfoAreaId	Name of the Request for Change root info area.
RightsFilterName	Name of the Rights Filter for the Edit control.
SourceCopyFieldGroup	Field group for the source info area.
TemplateFilterName	Template filter name for the Request for Change root info area.
TableCaption	Name of the table caption to be displayed.

SaveAndConfirm

This action is usually assigned to a button in reports, questionnaires and the quick add. If the user taps on this button, the displayed data is saved and cannot be edited anymore afterward.

Below is a list of input arguments:

Input Argument	Description
ConfirmFilter	Name of the Template filter used for changing the (root) record. Defines the actions that are carried out when the user taps on the button, e. g. in the U1 (Survey) info area. Default value: `FINALIZED`.
BaseRecordConfirm-Filter	Applicable to questionnaires. Defines that data in the assigned source record can be changed based on this filter, e.g. the questionnaire's confirmation can be saved in the MA record.

Selector

You can use the `Selector` action template for record selectors of type `address`.

A record selector allows you to select data from another info area and "copy" them to the current info area.

Below is a list of input arguments:

Input Argument	Description
Type	Defines the used record selector type. Currently only the `Address` type is supported.
SearchAndListName	Name of the Search&List configuration used for this record selector, e.g. ZA to add additional addresses.
ListConfigs	List of additional Search&List configuration, separated by comma.

Input Argument	Description
LinkInfoArea	The info area linked to the current info area, e.g. if you enter FI, values from the company record linked to the current record are used.
TemplateFilterName	Defines a Template filter to set initial values.

SendAdminInfo

This action template enables the user to send the Admin Info (e.g. synchronization information) to the desired e-mail address.

Below is a list of input arguments:

Input Argument	Description
eMailAddress	The recipient's e-mail address.
syncStats	Currently not implemented.

SerialEntry

This action template calls the quick add.

Below is a list of input arguments:

Input Argument	Description
ImageActiveShoppingCard	For compatibility purposes only (not applicable in V 2.x).
ImageInactiveShoppingCard	For compatibility purposes only (not applicable in V 2.x).
PositionAdditionalFilter	List of additional search filters applied on items, separated by semicolon. These filters can contain parameters.
PositionFilter1 - PositionFilter4	Name of the search filters applied on items, separated by semicolon. These filters can contain parameters.

Input Argument	Description
SavedAction	Name of a context menu action that defines the action that is executed when the quick add is closed by a tap on the **Close** button.
ScanMode	If set to `true`, an additional button ( ) is displayed next to the quick add's search field. The user can enable the scan mode with this button.
EditType	Must be given, the following options are available: • `Order`: The order entry is applied with its predefined item list format. The fields having the `UnitPrice` and `Quantity` **Functions** assigned are displayed on the right in List view (can be overridden using the `RowDisplayConfiguration` input argument. • `POS`: Currently no specific formats implemented.
SourceConfigName	Name of the field group that defines which data (source info area) is read, e. g. the products (item master) that are displayed in the quick add.
DestinationConfigName	Name of the field group (destination info area) whose Edit control defines in which fields the data of the (entered) items (e. g. an order item) is stored. If a List control containing sort fields also exists in this field group, its sort fields are used to sort existing records (e.g. the records in the shopping cart).
SourceChildConfigName	Name of the Search&List configuration (or field group) that references the root record's source child info area where the Edit control defines the content of an item (= row) that is displayed in the quick add, e. g. delivery dates (LD) for orders.
DestinationChildConfigName	Name of the field group where the Edit control is used to fill in data of a row item's child record, e. g. the delivery dates for order items.

Input Argument	Description
SourceCopyFieldGroup	Name of the field group from where data is copied from the parent to the child record (e. g. the currency from an order to all its items).
DestinationRootConfig	Name of the Search&List configuration referencing the parent info area specified in the `DestinationConfigName` input argument.
Filter1 - Filter6	Defines the filters displayed in the quick add. Filters specified in these input arguments must not contain parameters.
InfoPanelDefintion	Defines the Info Panel (i.e. the information available for each item) in JSON syntax, see InfoPanelDefinition Input Argument.
PricingConfiguration	Name of the context menu action that has the `Configuration :Pricing` action assigned.
ListingConfiguration	Name of the context menu action that has the `Configuration :Listing` action assigned.
QuotaConfiguration	Name of a context menu action (not defined in the default configuration) that has the `Configuration :Quota` action call assigned. The `Configuration :Quota` action template defines that the quota configuration.
SourceRequestOption	Defines how data is requested from the source info area, see `RequestOption` argument in General Input Arguments on page 70.
DestinationRequestOption	Defines how data is requested from the destination info area, see `RequestOption` argument in General Input Arguments on page 70.
Sections	If set to `true`, sections are displayed in the quick add.
SourceParentInfoAreaId	The info area of the (linked) source's parent record (that is linked with the source record).

Input Argument	Description
DestinationParentConfigName	Name of the field group in the destination's parent record.
DestinationFilter	Name of the filter applied to destination records.
Options	Additional options in JSON syntax, see Examples for Additional Options of the SerialEntry Action Template.
OrganizerContextValues	Comma separated list of items that map fields with assigned **Functions** to JavaScript variables. These values can then be used for JavaScript operations. Syntax: `<JSVariableName>:<designerFunctionname>`
RootFilterTemplateForStartSerialEntry	Name of the filter that is applied on the root record when the quick add is started (i.e. when the first item is stored).
RootFilterTemplateForEndSerialEntry	Name of the filter that is applied on the root record when the quick add ends (i.e. the quick add is closed, the user switches to another tab, etc.).
RowDisplayConfiguration	Name of a Search&List configuration that defines which fields (max. 4) are displayed on the right side of a row in the quick add list. (The fields displayed on the left are defined in the `SourceConfigName` input argument.) If given, the `EditType` "Order" is overridden.
SearchFilter1 - SearchFilter2	Names of additional filters that are applied to the quick add's text search.
SearchAdditionalFilter	List of additional filters applied to the quick add's text search, separated by semicolon.
DestinationTemplateFilter	Name of the Template filter of the row item's root record and is applied if a new row record is created.

Input Argument	Description
DestinationChildTemplateFilter	Name of the Template filter that is applied if a new child record is created for a row item.
DestinationParentTemplate-Filter	Name of the Template filter that is applied if a new destination parent record is created.
HierarchicalPositionFilter	If set to `true`, items are filtered hierarchically from left to right.
DocumentsDefinition	Defines the name of the document list and the filter applied for selecting records in JSON syntax, e. g.: `[{"name":"D1","filter":"D1.Brand"}]`
FixedCopyFieldValues	List of fixed copy fields (in JSON syntax) defining additional fixed parameters applied in the `Sorce-CopyFieldValues` input argument, e.g.: `{„Parameter1":23,"Parameter2":"Text"}` The defined valued can be applied by specifying `$parParameter1` bzw. `$parParameter2` in the `SourceCopyFieldValues` input argument.
FieldGroupDecider	Name of a Decision filter for optional filter masks that depend on the source record, see Decision Filters on page 203.
ErrorTranslation	List of field groups separated by semicolon containing alternative error messages, see Defining Error Messages on page 375.
FinishAction	An optional second closing action for the quick add displayed in the lower right corner.
QuantityChangeInList	Adds the +/- buttons directly in the quick add's items list allowing the user to change the quantity of an item. Step size is applied if configured.
ProductCalalogSource	Name of the Search&List configuration defining the Documents that are loaded as `ProductCatalog-Source`.

DocumentsDefinition Input Argument (OTC/LSI/CG only)

The `DocumentsDefinition` input arguments is only available in the OTC vertical:

Define this input argument in JSON syntax:

- `name` (mandatory): The content of the `Name` field in the Document record.
- `filter` (mandatory): Filter name that limits the documents to the ones linked to the quick add record.
- `style` (optional): The document type.
  - `IMG`: Pictures only
  - `NOIMG`: All documents except pictures
  - `DEFAULT`: All documents (default value)
- `addPhotoDirectButtonName` (optional): The name of the **Button** that has the `PhotoUpload` action assigned. This parameter displays a button with which the user can upload photos.

  If you do not set this parameter, existing documents are displayed (no upload possible) according to the value given in `style`.
- `hasDocumentsColumnFunctionName` (optional): Allows to upload pictures for empty item records.

  Default: [empty]
- `hasDocumentsColumnValue` (optional): Used, when pictures are uploaded for empty records.

  Default: [empty]

  Example:

  ```
  {"name":"D1","filter":"D1.U022","style":"IMG", "addPhotoDirectButtonName":"AddPhoto_SerialEntry"}
  ```

InfoPanelDefinition Input Argument

Defines the Info Panel (i.e. the information available for each item) in JSON syntax, e.g.

```
[{"name":"name1","maxResults":"2"} ,{ "name":"name2"}]
```

Available attributes:

- `name`: Either the name of a Search&List configuration or "Pricing" to apply pricing details such as discounts or bulk prices. This attributes must be given.
- `verticalRows` (`true` | `false`): Defines whether records are displayed horizontally or vertically.

  Default: `false` for Search&Lists, `true` for "Pricing".

- `maxResults` (numeric): Max. number of records to be displayed. Only applicable for if a Search&List configuration was given as `name`.
- `NoLink` (`true` | `false`): If set to `true`, links to AR are not set. Applicable when FIGr und FINo were defined as copy fields and are handed over to queries as parameters.

Examples for Additional Options of the SerialEntry Action Template

The following examples can be used as additional options for the quick add's SerialEntry action template in the **Options** argument field:

- Min. length of a search string if no filter is applied:

  `{"minSearchTextLength":"4"}`

- By default, the root record of an item is also synchronized. You can define, that records linked to the root record are also synchronized:

  `{"syncStrategy":"UP"}`

  You can define that all order items of an order are synchronized once more when the quick add is closed.

- By default, each row is transferred to the server after the user exits a row.

  `{"disableRowUpdate" :true }` defines that all the rows are transferred when the user exits the quick add.

- If the step size for packaging units were defined, quantities entered erroneously are automatically corrected to the package size. You can disable this function:

  `{"disableAutoCorrectPackageSize":true}`

- If the minimum/maximum quantities were defined, lower/higher quantities entered are automatically corrected to the defined limits. You can disable this function:

  `{"disableAutoCorrectMinMax":true}`

- `{"addSourceConfigs" :true }` defines that additional filters that include info areas linked to the source info area (e.g. AT) are also displayed. However, if one record is linked to many records (1:n link) in the source info area, this record is only displayed once (and not multiple times for each linked record).

  Note: Some linked records may not be synchronized if there are custom filters defined on the record. To override the filters and ensure that the linked records are synchronized, set the Sync.IncludeLinkedData parameter to true. For more information, see Filters on page 193.

- By default, values are calculated and saved after the user exits a row.

  `{"computeRowOnEveryColumn" :true }` defines that each field is calculated anew and saved when the field is exited.

- `{"autoCreatePositions" :true }` defines that POS items are added (as defined in the listings).

By default, before the items are added you must confirm their creation. The texts that are displayed as message and in the message's header can be defined as follows:

- `{"autoCreateMessage":"mytexts:0"}`: The message text. The given text should contain the %d parameter (for the number of records).

  The syntax for specifying the texts is `smartbook: <clientTextgroup>:<no>`.

- `{"autoCreateMessageHeadline":"mytexts:1"}` -> The message box's title

You can also define that they are added automatically without confirmation:

`{"autoCreateDoNotAsk":true}`

- `{"disableRowDelete" :true }` defines that a row is not deleted when the value is set to 0.
- `{"DontShowRebates" :true }` defines that only quantity multiplied by unit price is displayed. Discounts or the net price are not taken into account.
- You can define that an alternative label is displayed in the shopping cart:

  `{"SummaryTitle":"<TextGroup>:<TextNr>"}`

- You can define an alternative text to be displayed for the **All Items** button:

  `{"AllItemButtonTitle":"<TextGroup>:<TextNo>"}`

- You can define an alternative title for the **Summary** button:

  `{"SummaryButtonTitle"}:{"<TextGroup>:<TextNr>"}`

- You can define that the third row is **not** displayed in the quick add (depending on the number of fields in the configuration)

  `{"DisableThirdListRow":1}`

- `{"hideZeroStock" :true }` or `{"showZeroStock":false}` define that items are not displayed if no quota was defined for them or where all the samples have already been issued.

  Note: Do not use both parameters in combination but only one or the other.

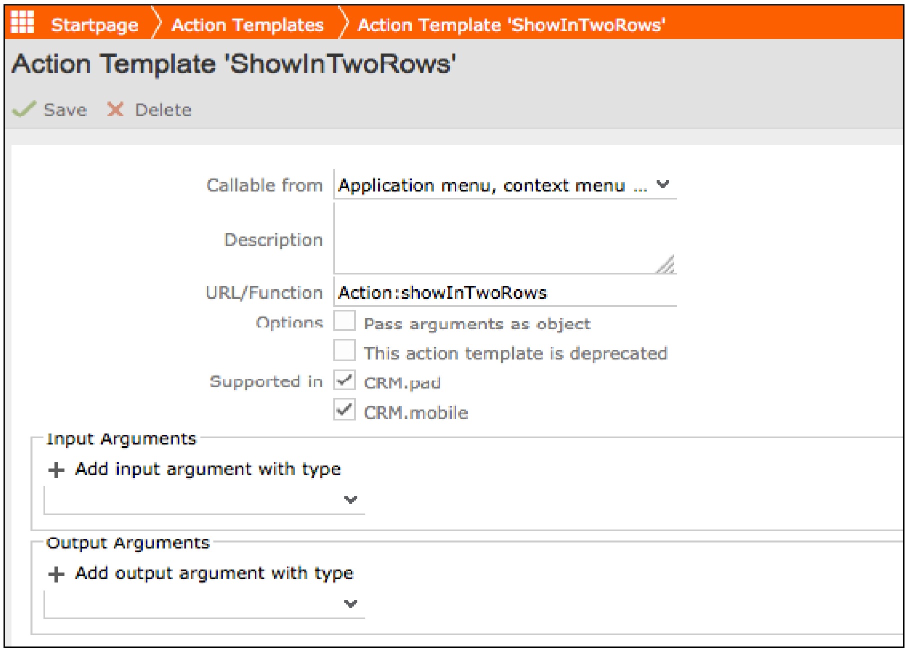
  Default: `{"showZeroStock":false`

- `{"DisableDuplicateRow" :true }` defines that the user is not allowed to duplicate rows.
- `{"pdfHIColor":"#00FF0045"}` defines the color that marks barcodes with available PDF annotation. The color must be given in hex format.

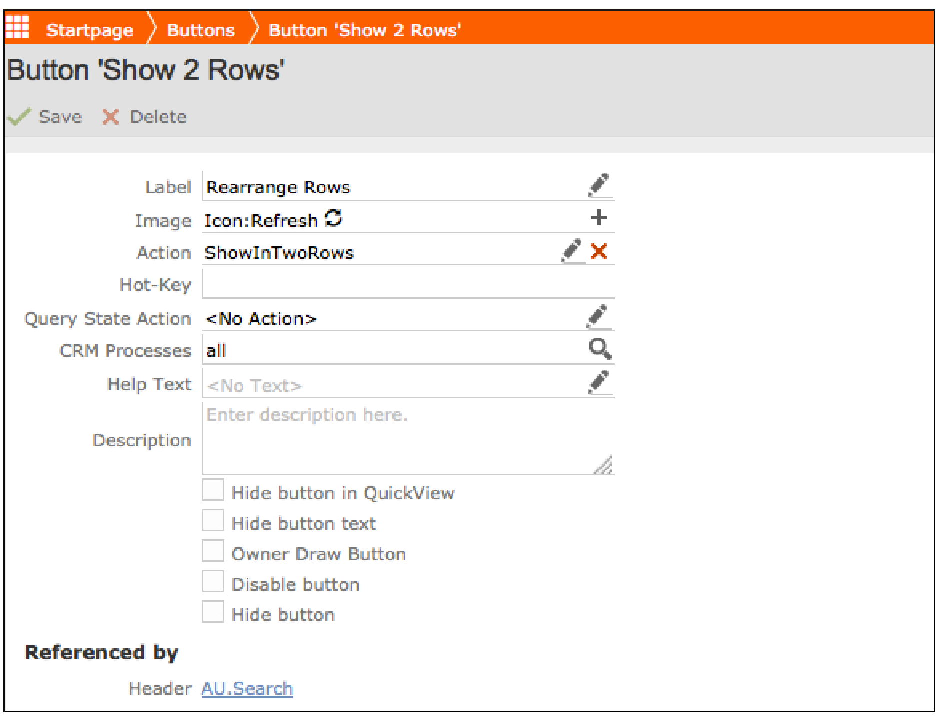
Listing serial entries in a two column layout

To list serial entries in a two column layout, perform the following:

1. Create an action template with the URL or Function parameter set to Action: showInTwoRows.

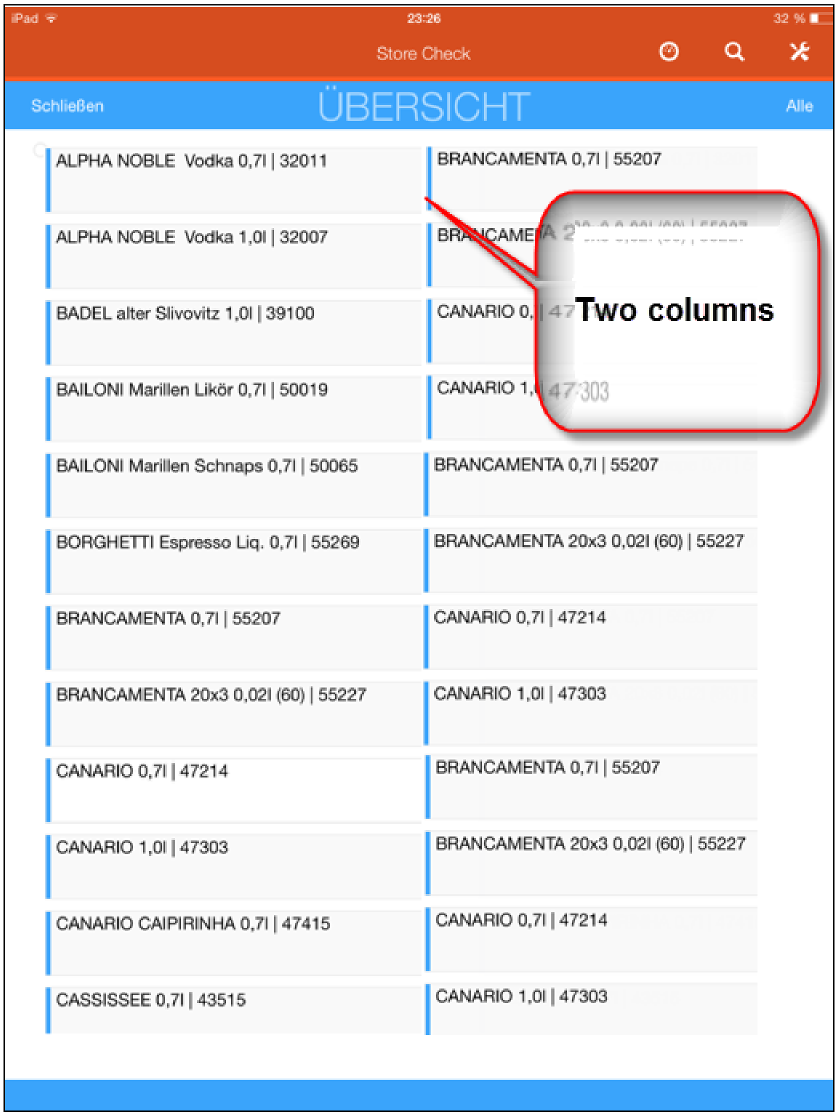


2. Create a Button and then attach the action template to it.



3. Add the button to the action bar of the desired serial entry.

The following screenshot shows a two column display of a serial entry listing in the CRM.pad application:



SerialEntryEdit and SerialEntryNew

These action templates are included in the default configuration for compatibility purposes only.

No input arguments

SettingsEditView and SettingsView

This action template allows the user to view (SettingsView) or edit (SettingsEditView) the settings in the Control Center.

Below is a list of input arguments:

Input Argument	Description
LayoutName	Name of the Web Configuration Layout applied.
HeaderName	Name of the header applied.
OrganizerColor	Not applicable anymore.!

SwitchTabIndexAction

This action template opens a separate tab (sub-list). You can assign it to buttons.

Example: The **Process** panel implemented in the LSI template uses this page/action, see Switch Tab Index.

Below is a list of input arguments:

Input Argument	Description
TabIndex	Index number of the tab (starting with 1), value between 1 and TabCount.

SyncRecord

Using this action template a record is synchronized instantly.

It also synchronizes existing child records (e.g. company, contact person, contacts). The `SyncRecord` action template can e. g. be assigned to a button (that is added to the Edit page).

Example: You add an appointment to a company and to one of your contact persons in that company. If you tap the button for synchronizing in the company, the appointment for both records, the company (parent record) and the contact person (child record) are synchronized.

The record and all child records (if defined in the offline dataset, all depending record sets and the dataset with the same name as the `infoareaid`) are downloaded.

To include parent info area in the synchronization process, specify the `SyncParentInfoAreaId` input argument in `NewView`, `RecordView` and `EditView` action calls, see NewView on page 106, EditView on page 93 and RecordView on page 123.

Below is a list of input arguments:

Input Argument	Description
RecordId	The current record's ID.

SystemInfoPage

This action template displays the system information in the control center.

Below is a list of input arguments:

Input Argument	Description
fieldgroup	Defines the field group that is used to display the contents of the system information. The `SYSTEMINFO` field group in the Company (FI) info area exists in the `update.tablet` configuration. However, you can assign a field group of any info area.

Tile:Record

Learn about the use of `Tile:Record` action template.

If you assign the `Tile:Record` action, you can display a tile in the Details view header showing the content a field of the record, e. g. for a company you could display the Status. However, no action can be executed in CRM.pad when the user taps on the tile.

Below is a list of input arguments:

Input Argument	Description
uid	The record's UID.
tableCaption	Name of the table caption that defines which fields are displayed on the tile.
imageMapFilter	Name of the filter that defines the mapping between field values and images.

Input Argument	Description
style	The search tile's style. You can use a CSS class or a CSS color (e.g. `"#ABC"`, `"#ABCDEF"` or `"rgba(255,234,210,0.7)"`)
size	Defines the width and height of the tile in the following syntax: `<width>x<height>`. `<width>`supports the following values: • `f`: full screen • `h`: half of the screen • `t`: third of the screen • `q`: quarter of the screen `< height >` supports numbers between 1 and 3, where 1 is one line high and 3 is three line high. By default the tile is full width and full height.

Tile:Search

Learn about the use of `Tile:Search` action template.

If you assign the Tile:Search action, you can display a tile in the header of a record displaying the number of records found. However, you cannot assign an action to the tile in CRM.pad.

Below is a list of input arguments:

Input Argument	Description
name	Name of the Search&List configuration defining the output field and the filter applied.
filter	Name of a filter name that is additionally applied. If no filter was given in the Search&List, this filter is applied instead.
link	Name of a link record.
linkId	If a link record was given, you can specify the `linkId` to be applied.
infoAreaId	You can specify an info area ID that is applied if `name` and `filter` were not given.

Input Argument	Description
style	The search tile's style. You can use a CSS class or a CSS color (e.g. `"#ABC"`, `"#ABCDEF"` or `"rgba(255,234,210,0.7)"`)
size	Defines the width and height of the tile in the following syntax: `<width>x<height>`. `<width>`supports the following values: • `f`: full screen • `h`: half of the screen • `t`: third of the screen • `q`: quarter of the screen `< height >` supports numbers between 1 and 3 where 1 is one line and 3 is three. By default the tile is full width and full height.

TimelineView

This action template displays data on a timeline.

Below is a list of input arguments:

Input Argument	Description
SortSequence	The list is sorted by date. You can define, whether it is sorted ascending (`ASC`) or descending (`DESC`).
TimeLineConfig-Name	Name of the **Timeline** configuration that was defined in CRM.designer's **Timeline** area.

WebView

This action template is used to display web pages in CRM.pad.

Below is a list of input arguments:

Input Argument	Description
HeaderName	Name of the special header that is applied if the web page is displayed in an own page.
Height	The height of the panel if the web page is displayed in the Details view.

Input Argument	Description
ReportType	`Url` (fixed value).
Url	The web pages URL. Use `index` if you want to display an internal tutorial.

Info Areas

Learn how to customize display on Aurea CRM.pad using info areas.

Field Groups

Learn about the field group info area.

For general information on fields and field groups, see Field Groups in the *Aurea CRM web Administrator Guide*.

Field Controls

Learn about the field controls in CRM.pad.

The following field controls are used by CRM.pad.

Details Control

Learn how to define the contents of Details view.

You can define the contents of the Details view in the Details control. In addition, it is used as a fallback if no Edit control was defined.

As in Aurea CRM web fields are grouped in panels, see Panels on page 164.

If no child records are found on the device (i.e. the database on the device does not contain any records for an info area), CRM.pad automatically searches for data online.

Adding Panels to the Details Control

To add panels, e. g. to display a map click on the **Add new Tab/Group** button.

Adding an Image

You can add a field to display an image to the Details and Edit control.

To add an image:

1. Add a field to store the image's document key and assign the **Image** field attribute.
2. If you want to limit the display size of the image, you can define a maximum size with the following JSON syntax:

   ```
   {"previewWidth":"90","previewHeight":"120"
   ```

Defining Field Contents to be shown in the Details Header

You can define in the Details control that fields from the Details control are displayed in the header below the table caption, see Table Captions.

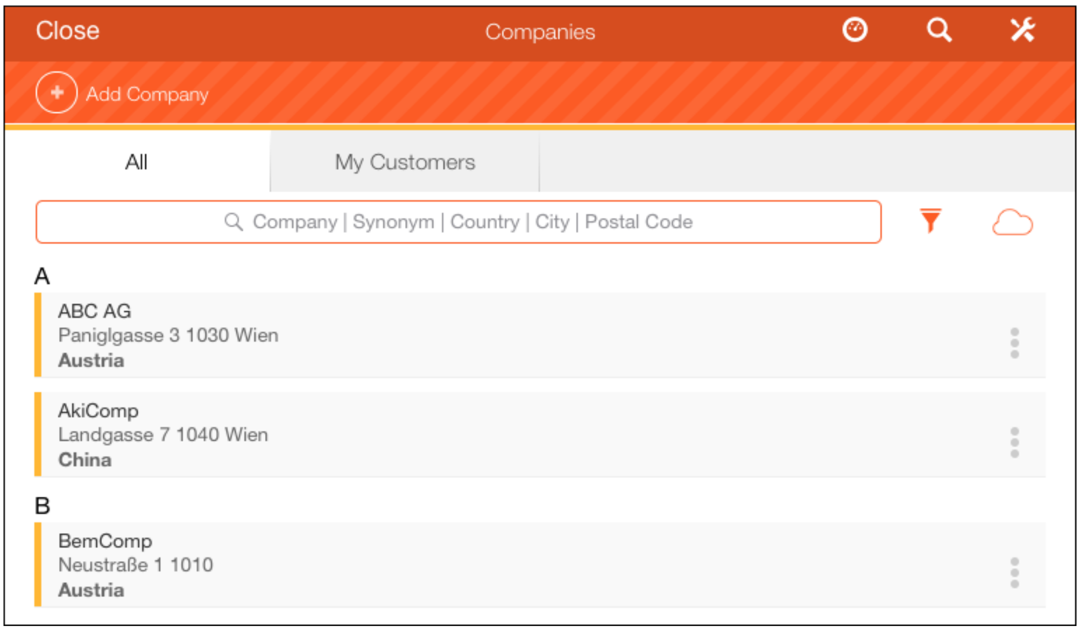
Assign `OrganizerHeaderSubLabel` as **Function** to the field(s) you want to display it in the header. You can also assign the `Colspan` field attribute, see Field Attributes on page 158.

List Control

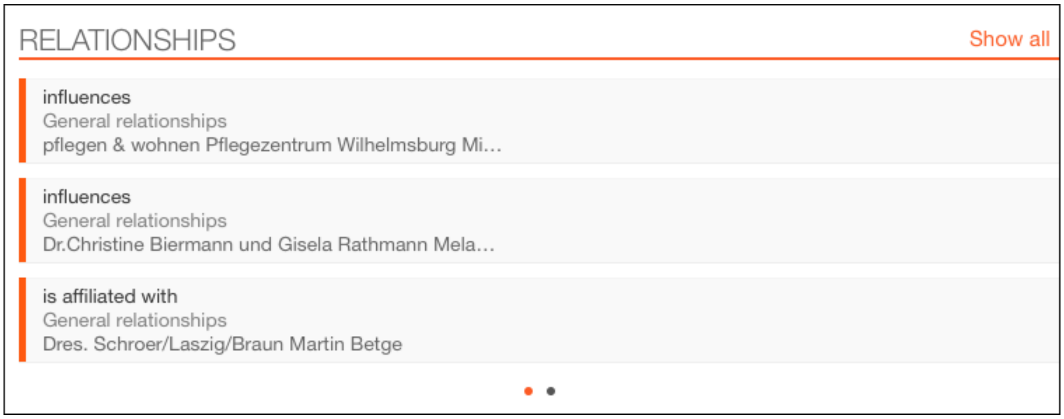
You can define which fields of a record are displayed in lists and the sort order of the records displayed.

Lists are displayed in CRM.pad as follows:

- As search result list:



- As a list containing child info area that are displayed in the parent info area's Details page: (e. g. all relationships of a company or person).



Both lists use the (same) List control.

The fields you selected are displayed in the following order by default, but you can change this order, see Defining the Fields' Order. The first selected field is formatted in bold. The order how they are displayed is shown by the direction of the arrows in the following example:

- You can also define that an image is displayed on the left in the corresponding Expand configuration.
- You can combine the content of several fields with the `ColSpan` option, see Combining Field Values on page 163. You can also define whether a field is defined in bold or in a specific **Field Color**, see Field Attributes on page 158.
- You can change the displayed text (YES/NO) for Boolean fields, see Boolean Fields in the List and MiniDetails Control.

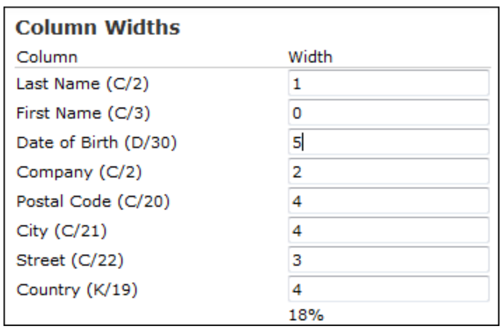
Defining the Fields' Order

In CRM.pad, the fields displayed in lists are not displayed in columns. Therefore, the **Column Width** attributes are used to define how data is displayed in lists.

By default, the fields are displayed in the order they were added to the **Fields** panel in two columns within a row:

In this example, the person's first and last name were combined with the `Colspan` attribute, see Colspan.

However, you can also use the **Column Width** attributes to define how the fields are displayed. You can enter order numbers, for example:



Column Widths

Column	Width
Last Name (C/2)	1
First Name (C/3)	0
Date of Birth (D/30)	5
Company (C/2)	2
Postal Code (C/20)	4
City (C/21)	4
Street (C/22)	3
Country (K/19)	4
	18%

This results in the data being displayed as follows:

- Data is sorted according to the sort numbers.
- Fields with the same sort number are combined (in the order they were added to the panel), separated by a space.

Note: If you use the V 1.x layout, you can only enter the values 1 to 6 into this column, however, this restriction does not apply in CRM.pad 2.x. The width settings are otherwise not used in CRM.pad.Some functions may initially have a negative value entered in this column. You can replace these values with your desired order.

Grouping Lists by a Specific Field

You can define that a list is grouped by a specific field, e. g. the appointments list can be grouped by date.

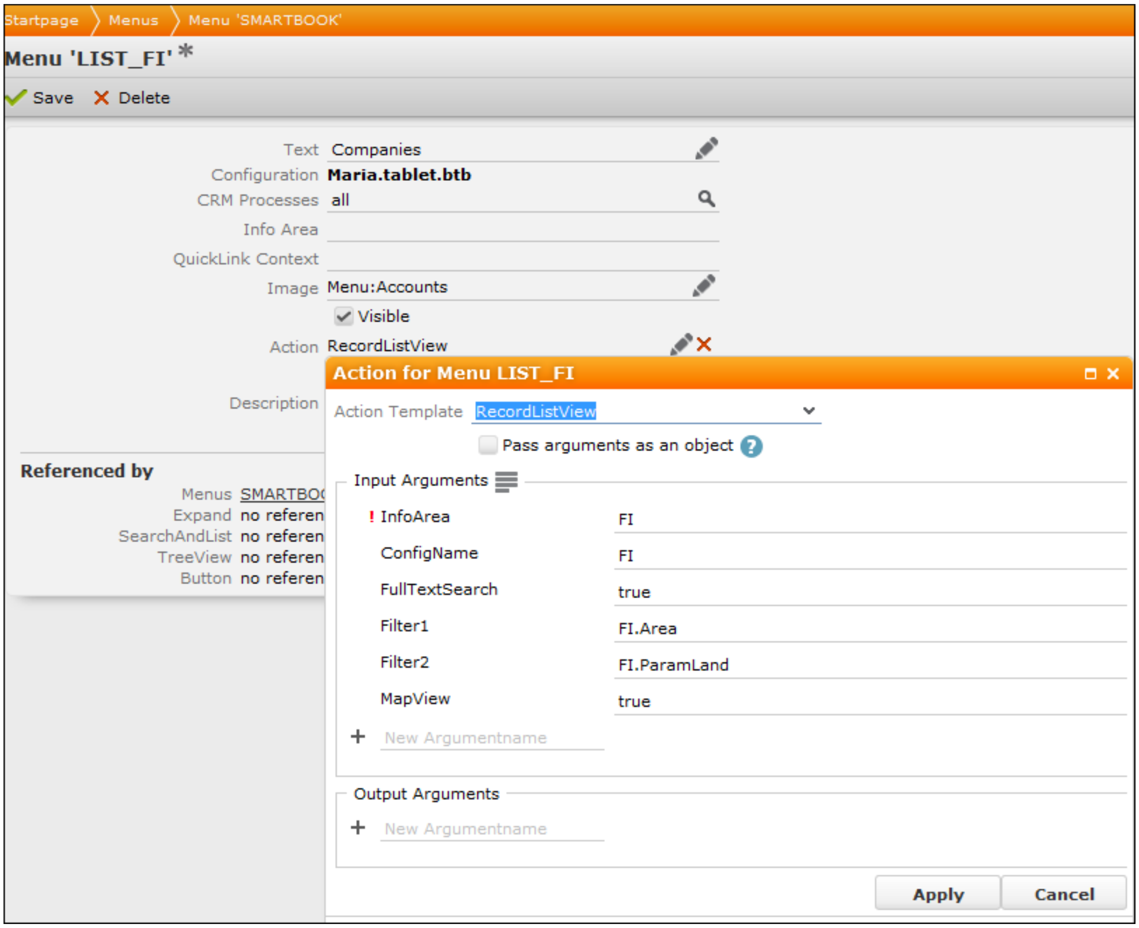
Select the desired field in the List control, enable the **Extended Options** check box and enter `{"SectionField":"true"}`

Displaying a Map in a List containing Search Results

You can define that a map showing the search results is displayed in the result list, e.g. a rep searches for the appointments of the day, and the location of the companies to be visited are marked in the map.

To display a search result in the map:

1. Add the GPS fields (Longitude and Latitude) to the desired List control (e.g., FI field group's List control).
2. If you do not want them to be displayed in the list, set the **Hide Field** attribute for these fields.
3. Assign the `RecordListView` action to the desired item (e. g. to the `LIST_FI` application menu action of the `SMARTBOOK` application menu) and set the `MapView` input argument to `true`:



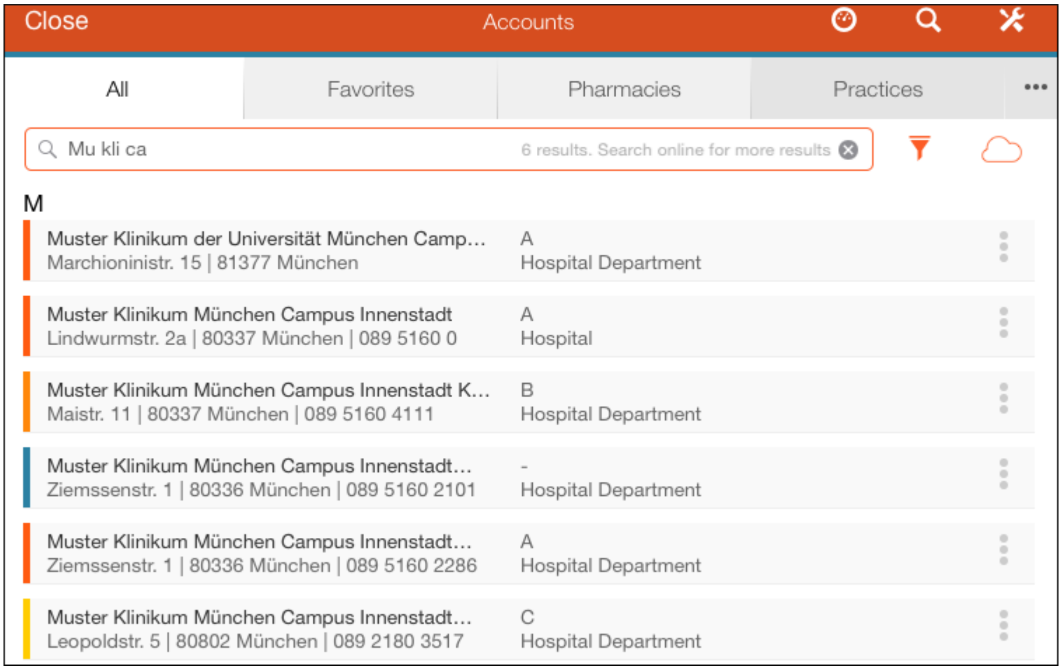
4. You can limit the displayed items with the (numeric) `MaxCountMapView` Web-Configuration Parameter, see MaxCountMapView.

Search Control

You can define the search fields available in the info area search and for record selectors in the Search control.

Only a text-based search field is available in CRM.pad.

The search results are listed below the search field:



You can add all field types, but the following restrictions for adding fields apply:

- Currently fixed catalog data types are not supported. You can add them as filters.
- You can include computed fields ("Z fields") and catalog values in offline mode only, as Aurea CRM does not support wildcard text search in catalog values.
- If you only add max. three fields of the current info area and the user enters more than one search criterion separated by a blank, then all the search criteria are searched in all search fields.

Edit Control

The Edit control defines the contents of the Edit view when creating new records (Edit control in New mode) and when updating existing records (Edit control in Update mode).

If no Edit control is defined for an info area, the Details control is used as a fallback.

You can define the following:

- The header text "Edit Details" uses text no. 1 from text group `Details_core`.
- You can group the fields on the control by defining group panels (), see Panels on page 164.
- The panel's **Label** is displayed as sub-heading for the group, e.g. "General Information", "Company Details", etc.
- Add fields and set their attributes as in Aurea CRM web, see Customizing a Control in the *Aurea CRM web Administrator Guide.*

- When using hierarchical catalogs, ensure that both the parent and child catalog fields to the control were added. Users need to save the record before being able to select a value from a child catalog.
- For detailed information on all available input controls used for the different field types (Boolean, catalog, date, etc.). For more information on input types, see the article “How To configure expand edit mode” at https://support.aurea.com.
- Combined fields are displayed separately in Edit mode.
- Z-fields without a Select-Function are automatically hidden in Edit mode.

The Edit control is used when you assign the following actions:

- EditView on page 93 (e. g. for new records)
- NewView on page 106.
- NewOrEditView on page 105.

If no Edit control was defined and the Details control is used as fallback, the following restrictions apply:

- `PARENT` & `MAP` panel types are not available.
- `CHILDREN` panel type is not available. If the Details control is used, these child panels are removed.
- Images cannot be displayed for Document fields.
- Fields that were set to `Readonly` in the Details control cannot be edited. Link fields are automatically read-only.
- If the `ColSpan` field argument was defined: If the field can be edited, at present only the first field is displayed. Only if a date field is followed by a time field, these two fields can be combined and only one edit field is displayed.

  `ColSpan` can also be used for dependent catalogs where the parent and the child value can be selected in one control.

  If fields are set to `ReadOnly`, `ColSpan` behaves exactly the same way as in the Details control, see Combining Field Values on page 163.

Configuring the Keyboard

You can define that the keyboard is shown according to the field type, e.g. that for an e-mail address field the @ character is available.

To set the keyboard's layout:

1. Switch to the desired Edit control.
2. The options **Phone**, **Hyperlink** and **Mail** display customized keyboards.

Note: At present, in addition to the Edit control these options are also available for Interests (Characteristics) and Objectives.

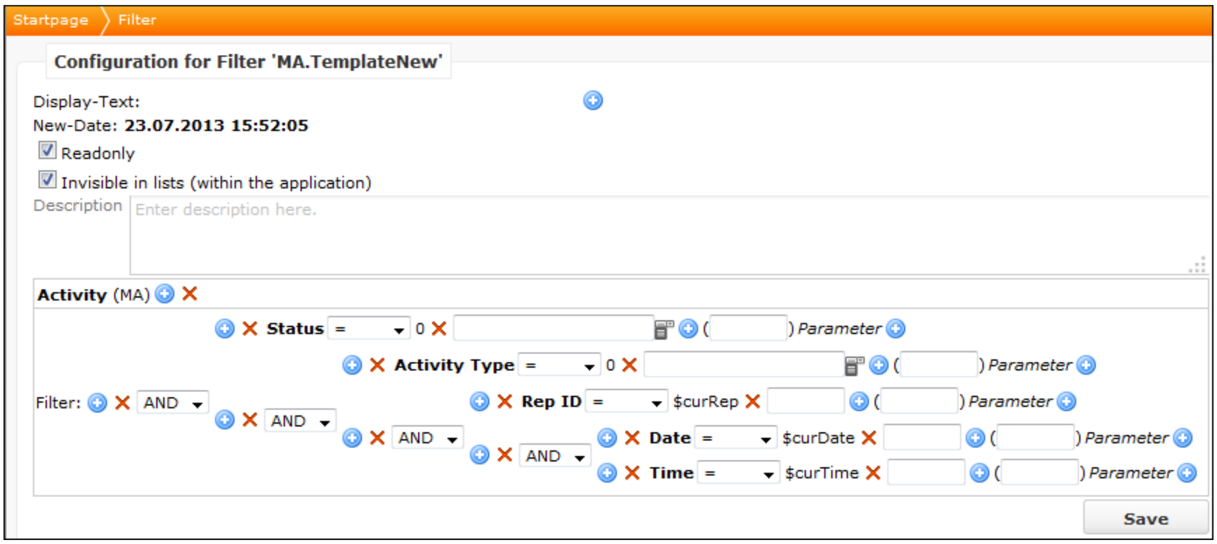
Predefined Values for New Records

You can define values that are automatically entered when a new record is created. This requires a Template filter that you specify when you assign the `NewView` action call.

To specify a Template filter in the `NewView` action:

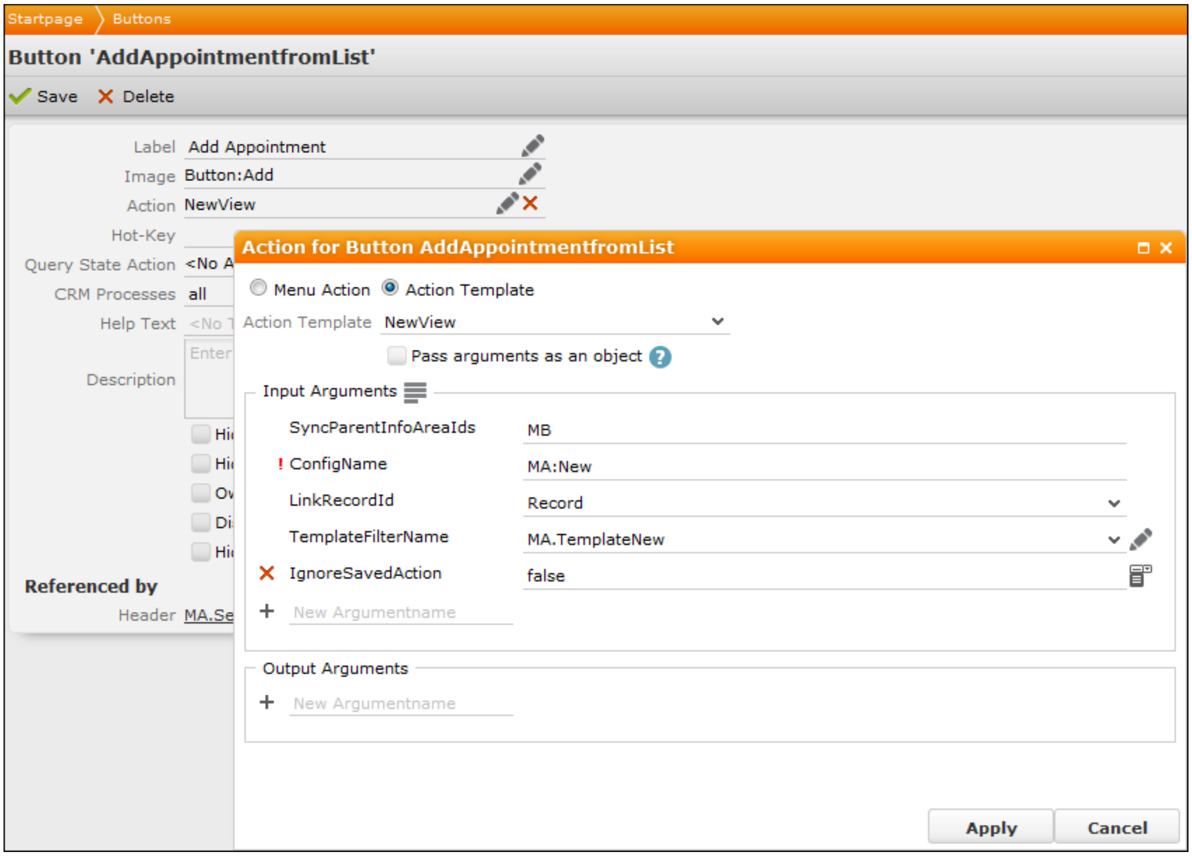
1. Define a filter with the values you want to enter automatically, see the CRM.designer Administrator Guide on how to define filters.

   For example, the following filter (`MA.TemplateNew`) sets the activity type, the status, the current date and time, and the current rep name.

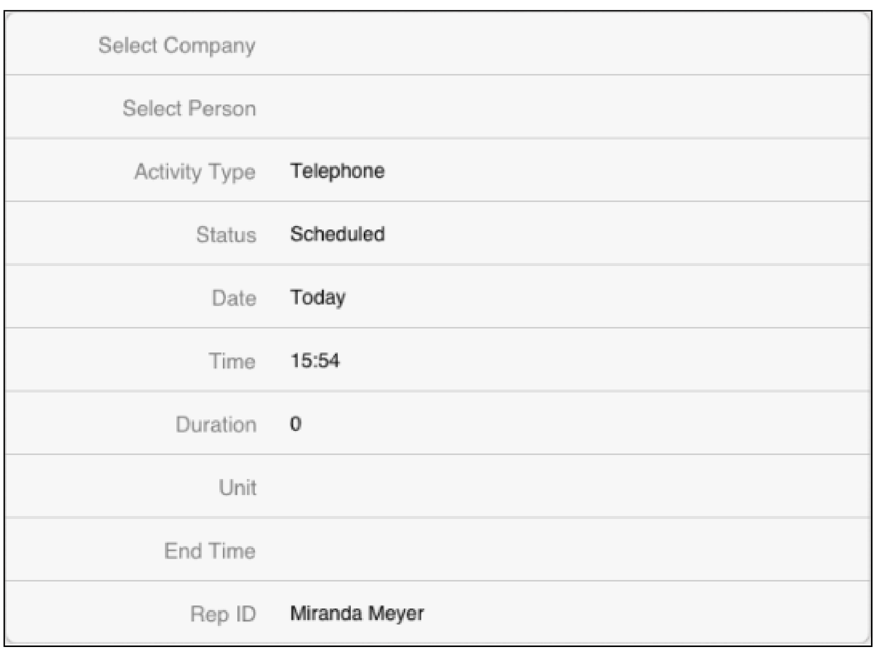


2. Assign this filter to a button, e. g. the `AddAppointment` button, see Buttons on page 63.

3. Click on (**Edit**) besides **Action** and enter your filter's name in the `Template-FilterName` field.



In this example, when the user clicks on the (**Add Appointment**) button, the filter is applied and the following data is automatically entered in the new record.



Field values are either used as predefined field values (the field must be included in the record's Edit control or as fixed field values (the fields are configured on the Edit control and, therefore, cannot be changed by the user).

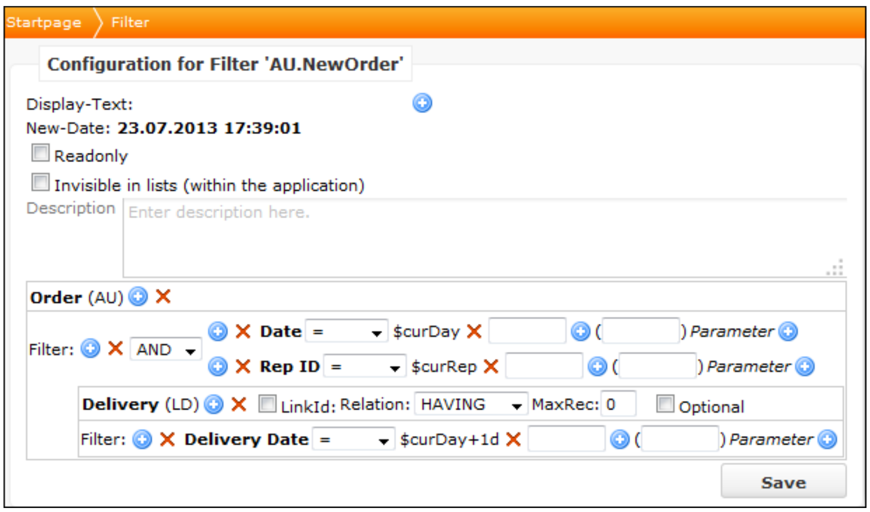
You can also add template filter values to tabs of type `CHILDREN` by adding the name of the filter as third argument, e. g. `CHILDREN_MA_MA.MyTemplateFilter`.

Creating Child Records for a New Record Automatically

You can also define in Template filters that child records are automatically created for new records.

Example:

The following filter enters the current date and the current rep in a new order (AU) record and also creates a delivery date record with the following day of the current date (`$curDay+1d`).



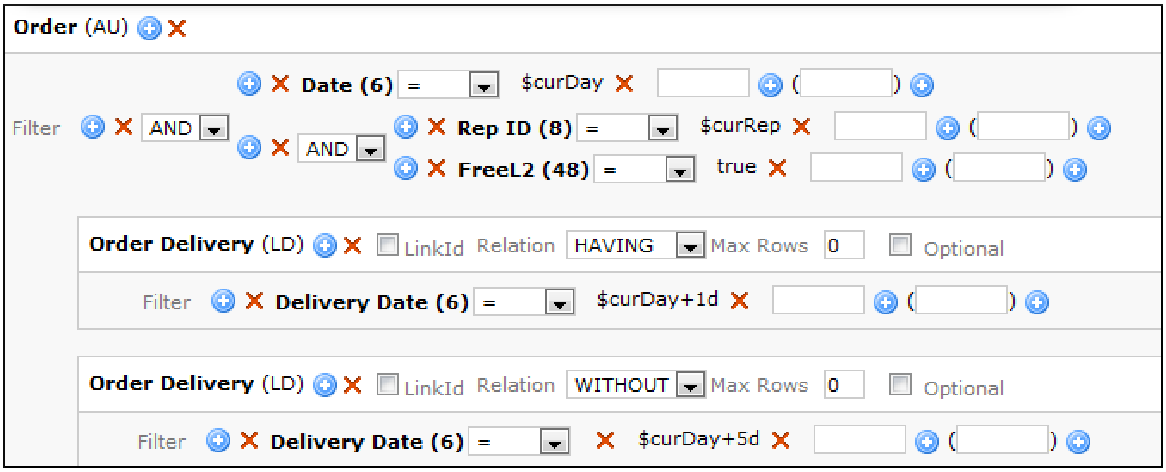
If you want to create more than one child record, add the same info area again.

Predefined Values for Additional Child Records

You can also set predefined values for further child records (if more than one child record are created) that are entered automatically when a user creates a child record by adding a sub-info area to the Template filter using the `WITHOUT` relation.

Example:

The following filter, in addition to the example from, see Creating Child Records for a New Record Automatically also creates a delivery date record where the delivery date is entered in the `WITHOUT` relation (`$curDay+4`).



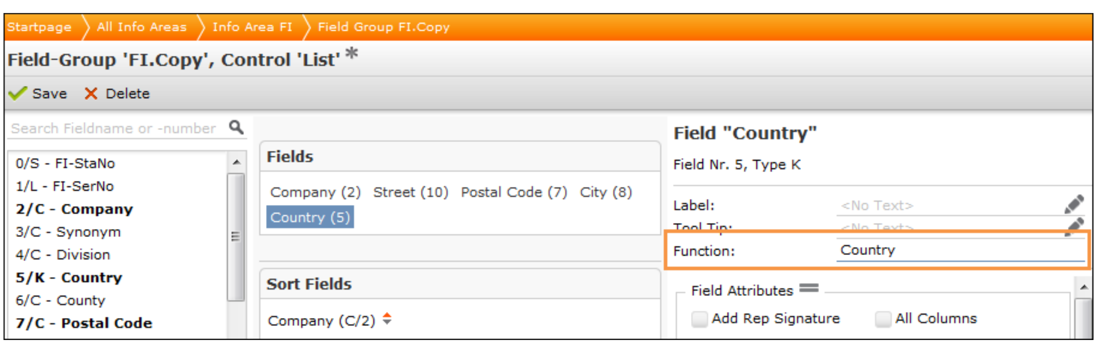
The first record is displayed with date = current date + 1 day, all subsequent child records are created by default with current date + 5 days:

Field Values from a Parent Record

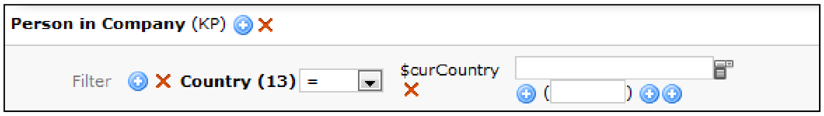
You can copy field values from the parent record to a new child record.

Example: When the user creates a new person for a company, the content of the company's Country field is copied to the person record.

1. Define a List control in the field group containing the fields you want to copy.
2. Select the desired field in the **Fields** area and enter the names for the fields you want to copy in the **Function** field.



3. Enter this field group's name in the `CopySourceFieldGroupName` parameter of the assigned action, e.g. the `NewView` action for the `AddPerson` button.
4. You can enter the fields with the `$par` prefix combined with their function names in the Template filter to access the values from the parent record.



Adding Icons

You can define an area containing icons that execute a function when a user taps on them. For example, if your reps have to cover predefined steps while visiting a client, you can add them to a page.

To configure such icons:

1. Define a context menu action.
2. Define and/or assign the desired actions for the icons to this context menu. The icons assigned to each menu action are displayed in the application.
3. Switch to the desired field group and open the control, where you want to add the icon area.
4. Add a tab and enter the context menu action's name into the **Type** field.

Editing Geographic Coordinates

You can add the geo-coordinate fields (longitude (x) and latitude (y)) to an Edit control and allow the user to clear the stored values and/or set the coordinates to his current location. To do so, perform the following steps:

1. Switch to the desired Edit control. For example, Edit control for the FI Field group.
2. If necessary, add the Longitude (x) and Latitude (y) fields (the Longitude field must be added first) to the desired Edit view panel.
3. Define the following **Field Attributes**:

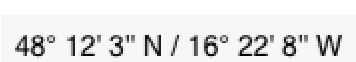
   For the Longitude (x) field:

   - Enable **No Label**
   - Enable **Extended Options** and enter `{"GPS":"X"}`

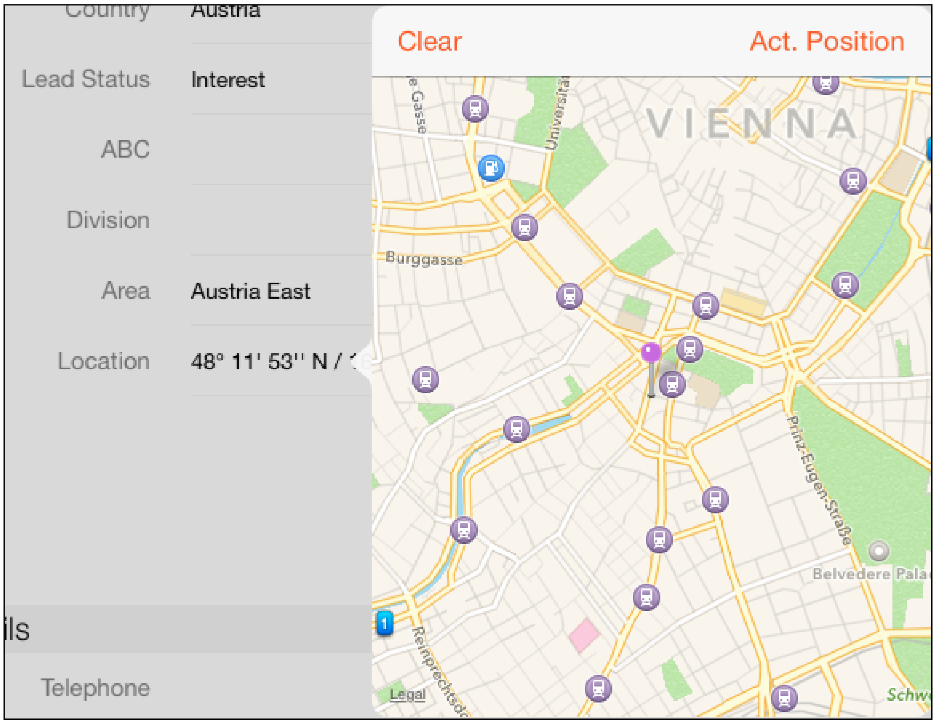
   For the Latitude (y) field:

   - Enable **Colspan** and enter `2`
   - Enable **Extended Options** and enter `{"GPS":"Y"}`

The two coordinates display as shown below, in the Edit view:



When the user taps on the coordinates, the MiniMap displays the location:



Users can change the geographic coordinates to their current position, delete them (with a long tap on the map), or enter new coordinates manually.

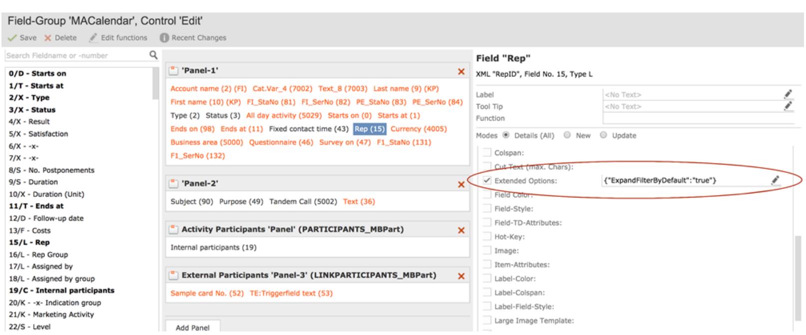
Rep Selection Filter

In some infoareas like Activities you can add Reps as internal participants. The Reps can be presented in groups or as a plain list of all the reps (default view).

To display the Reps in groups by default, add the following JSON configuration to the extended options:

```
{"ExpandFilterByDefault":"true"}
```

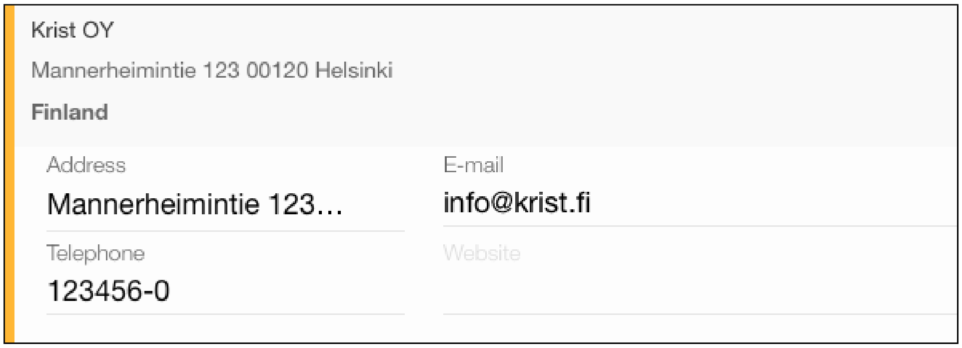
The following screenshot shows the settings for Rep fields in the Edit control mask of the MACalendar field group:



MiniDetails Control

Learn how to define the contents and layouts of minidetails control.

You can define the contents and layout of the MiniDetails (a pop-up that is displayed after the user clicks on (on the right of a list item). If you do not define a MiniDetails control, the first tab of the Details control is applied as fallback.



Adding Buttons to the MiniDetails

You can add buttons to the MiniDetails, e.g. **Add Activity**. This allows the user to add records from the list without having to switch to the Details view of a record.

To add buttons to the MiniDetails:

1. Switch to the **ExpandOptions** header of the selected field group's (e. g. FI) **Header Group**.
2. Add the desired button.

The following restrictions apply to MiniDetails:

- If you add a button that has the `OrganizerAction` assigned, you can only specify `toggleFavorite` as the template's `Action` argument. All other options are **not** supported.
- If you add the MiniDetails to a list that is displayed in a Legacy Form, `toggleFavorite` is also ignored.

Boolean Fields in the List and MiniDetails Control

By default, for Boolean ("logical") fields the text YES/NO (in English) or I/O (in German) is displayed. You can define that a different text is displayed.

There are two ways to display different text :

- Switch to the desired field group control. Select the desired field and enter three strings separated by semicolon in the **Label** field.

  The first string is the field name, the second string is the value to be displayed for TRUE (YES) and the third string is the value for FALSE (NO).
- If you set the `Format.ShowFieldNameForTrueValue` Web Configuration parameter, the field label is displayed (and no `FALSE` value is available), see Format.ShowFieldNameForTrueValue.

Matchup Control and MatchupList Control

These controls are not used in CRM.pad.

Control Attributes

You can define control attributes that apply to a control.

Note: Control attributes are sometimes also referred to as mask attributes.

To define a control attribute:

1. Open the desired control. The following **Attributes** pane is displayed.

   If the **Attributes** pane contains more options, you have clicked an item in the control. Click on the empty space to display the control attributes.

2. Define the desired attribute.

Edit Control Attributes

The following control attributes are available for the Edit control:

- **AddRecords**

  You can define the maximum number of child records that can be added to a parent record with the following syntax:

  ```
  AddRecords_<Number>
  ```

  Example: You can define the max. number of delivery dates for an order.

- **LabelWidthPercent**

  You can define a fixed label width for all fields in a specific panel in the Edit control with the following syntax:

  ```
  LabelWidthPercent_<PanelNo>
  ```

  Possible values: 1 - 100.

- **ProtectedParent**

When editing a MA record, the user can remove all external participants. You can define with the `ProtectedParent` parameter that specific linked info areas cannot be removed. Syntax:

`ProtectedParent_<PanelNo>`

Default value: `CP ;PE ;FI`

List Control Attributes

The following control attributes are available for the List control:

MaxResults

You can limit the number of results displayed in a list with the following syntax:

`MaxResults=5`

Can also be applied to a client report and sublists in a Details control.

Details Control Attributes

- **action**

  You can define that a button is displayed in a panel of the Details view e.g. an edit button allowing the user to change only the panel's content.

  Syntax: `action<PanelNumber> = <ButtonName>`

  Example: `action3 = Edit`

  You can also assign the action in the Panel Editor's **Attribute** field. In this case, you need not specify the `<panelNumber>`.

  You can enter more than one button separated by semicolon.

- **ParentInfoImage**

  If the `Image` field argument was defined for a field, the image in the field is displayed in the details view. You can override this definition by setting the `ParentInfoImage` control attribute.

  Syntax: `ParentInfoImage<PanelNumber>=<ImageName>`

  You can also assign this attribute in the Panel Editor's **Attribute** field. In this case, you need not specify the `<panelNumber>`.

- **disablePaging**

  If you have enabled paging in lists, you can define that paging is disabled for specific panels in the Details view.

  Syntax: `disablePaging<panelNumber>`

  Example: `disablePaging3` disables the paging for the third panel in the Details view.

  You can also assign the option in the Panel Editor's **Attribute** field. In this case, you need not specify the `<panelNumber>`.

Assign Actions to Tabs using Control Attributes

You can define that specific actions are available for the user.

The actions can be assigned by entering the desired button name in one of the following ways:

- As control attribute
- As panel attribute for a specific panel

Syntax:

`action( <n>)`

`<n>` is the panel number, if the action is assigned as control attribute and needs not be given if assigned to the specific panel.

Possible values: name of the desired button(s), separated by semicolon (;)

Field Attributes

Learn about the field attributes for CRM.pad.

The following field attributes are available:

ColSpan

Combines field values in view mode, see Combining Field Values on page 163.

Extended Options

You can use this argument to define specific options:

- `{"newLine" :true }` inserts a line break after the field.
- `{"supportsDecimals":"false"}` defines that the user cannot enter decimals in this field
- `{"catalogStyle":"popOver"}` defines that catalog values are displayed in a pop-over.
- `{"ShowFieldNameForTrueValue" :false }`: By default, Yes/No is displayed for boolean fields. You can define, that the field name is displayed instead, see Format.ShowFieldNameForTrueValue. If you specify `{"ShowFieldNameForTrueValue" :false }` the `Format.ShowFieldNameForTrueValue` and the `Format.ShowFieldNameForTrueValue` Web Configuration parameters are ignored.
- `{"phone":"tel"}` defines that you can call the telephone number in the field (if exists) directly from your iPad.

  Note: This option is only available with hardware that supports Apple Continuity. For more information see https://www.apple.com/ios/whats-new/continuity/.

Field Color

Defines the color of the field value. You can also enter a hex value, e. g. `#FF0000FF`, or in the format "`<red>;<green>;<blue>;<alpha>`", each entry to be given as a value between 0,0 and 1,0.

Hyperlink (Details view only)

The field is displayed as a link, i.e. fields from the current info area are handled as a hyperlink. For linked fields the default action defined for the linked info area is executed (i.e. in most cases the target record is displayed).

Image (for Document fields' details only)

If the document field contains the key of a document that is available offline, the image is displayed if this check box is checked. The image is shown to the right of the tab group. Empty fields are not displayed.

Label Color (Details and MiniDetails only)

Defines the color of the field label. See Field Color for more information.

No Label

Hides the field label.

Note: The field label is hidden by default in lists. Enabling this option results in the field label being displayed (which can be useful, for example, if you only display and sort numbers).

No Multi Line

All line breaks and multiple spaces are removed.

Note: If you copy the text to the clipboard, the actual text (incl. line breaks and spaces) is copied.

Render Hooks

You can influence how a field's content is displayed with the `Render Hook` field attribute. For example:

- `{"FieldType":" <type >"}`: Defines the number type. You only need to define this if the desired field type differs from the current field type.

  Possible values: `F` (float) and `L` (long integer)

- `{"PercentField" :true }`: Formats the field as percentage field (i.e. the % character is appended) and also calculates its content as percentage value.

  Example: 0,10 is displayed as 10 %

- `{"DecimalDigits": <number >}`: Defines how many decimal digits are displayed.

Example: `{"DecimalDigits":0}` does not display decimal digits

Note: You can also combine these formatting options, e.g. to display a percentage field with one decimal point: `{"PercentField" :true ,"DecimalDigits":1}`. These formatting options do not change the actual value in the database, but only influence how the value is displayed in CRM.pad.

- `{"GroupingSeparator":"false"}`: Defines that the digit grouping symbol is not displayed.

  Possible values: `true` (default) / `false`.

  Example: `{"GroupingSeparator":"false"}`

Field Functions

You can specify functions for each field in the **Function** field (that is displayed on the right side after selecting a specific field).

Most function names are the XML name of the field. However, in some cases you need to use specific function name, e. g. Copy, Copy1 etc. if you want to copy field values to another info area.

If specific functions are required for a feature, they are indicated in the specific chapter.

Linked Fields

Learn about the use of linked fields in CRM.pad.

In many info areas, so-called "Z" fields are available. These are write-protected/decoded fields whose values are entered automatically by the business logic, e.g. system and link fields. These fields can cause problems in CRM.pad, therefore, always add these fields as linked fields and set a specific link, e.g.:

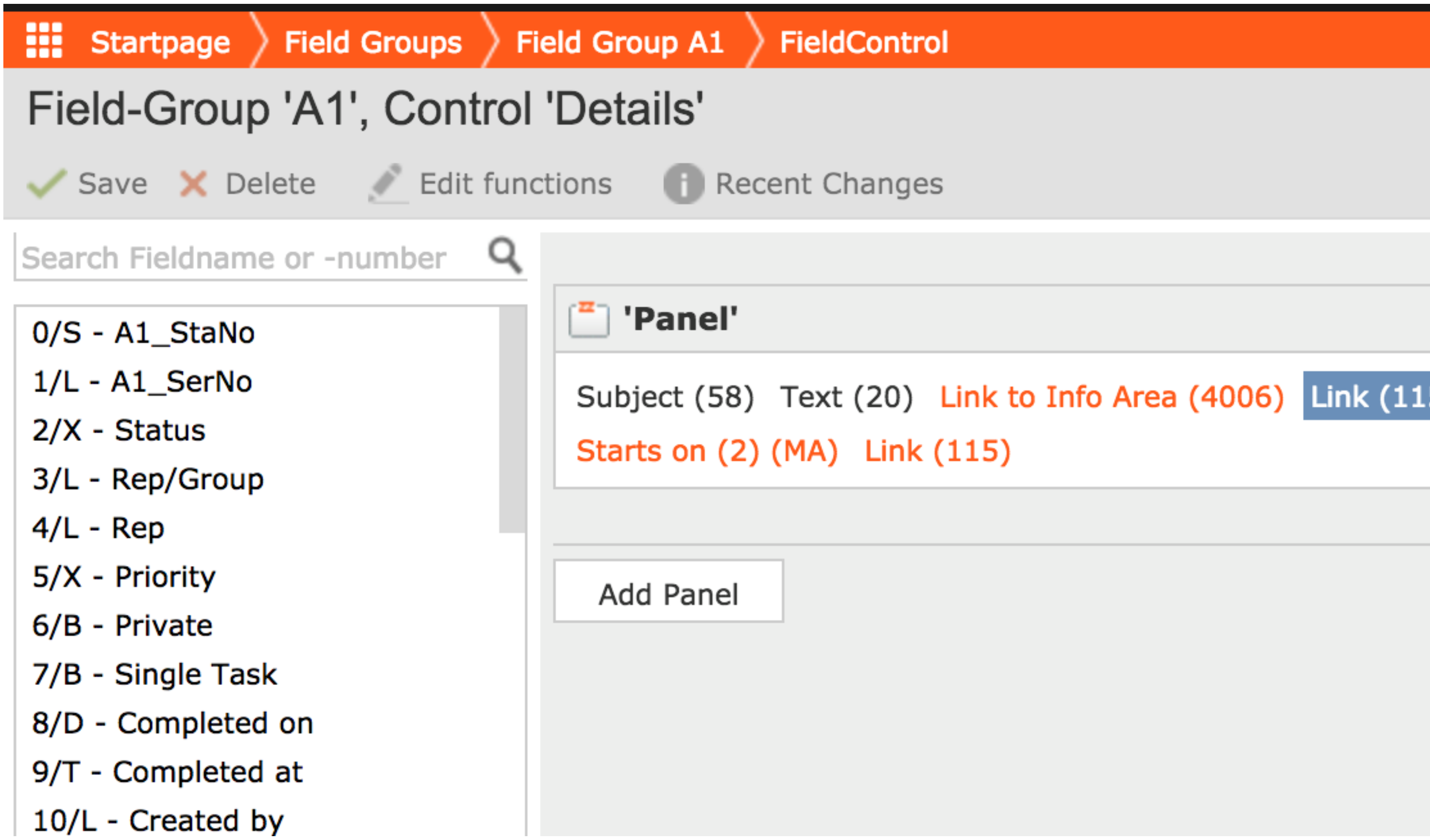
Linked fields are displayed read-only when in Edit mode.

For the `NewView` action call, linked fields are only supported if a record selector is defined. Example: The **Select Company** and **Select Person** fields in the `MA : New` Edit control.

implement the following changes in designer To ensure that Generic Links can work properly:

Hyperlink

Provide a function name as `GenericLink` for a field, which you want to define as a hyperlink to navigate to generic link, as shown in the screenshot.

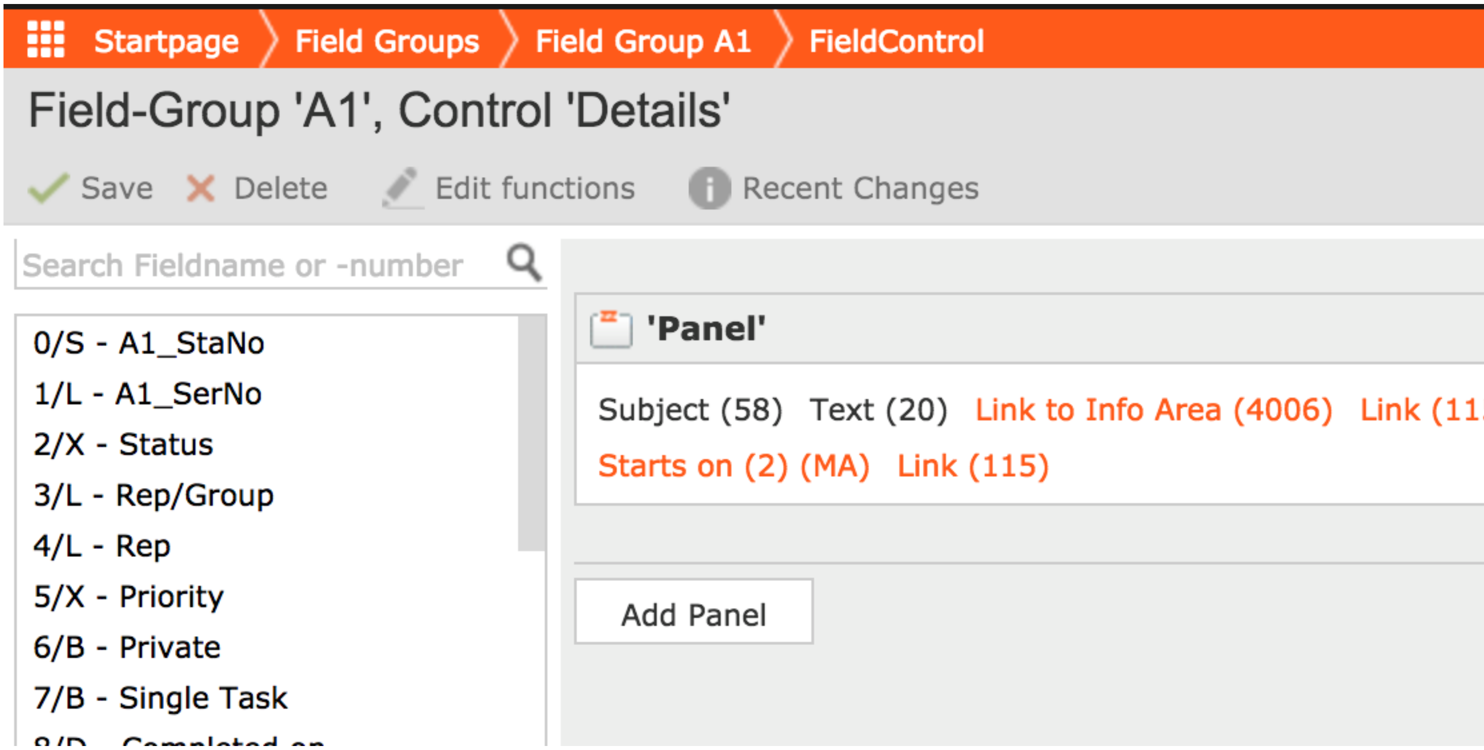


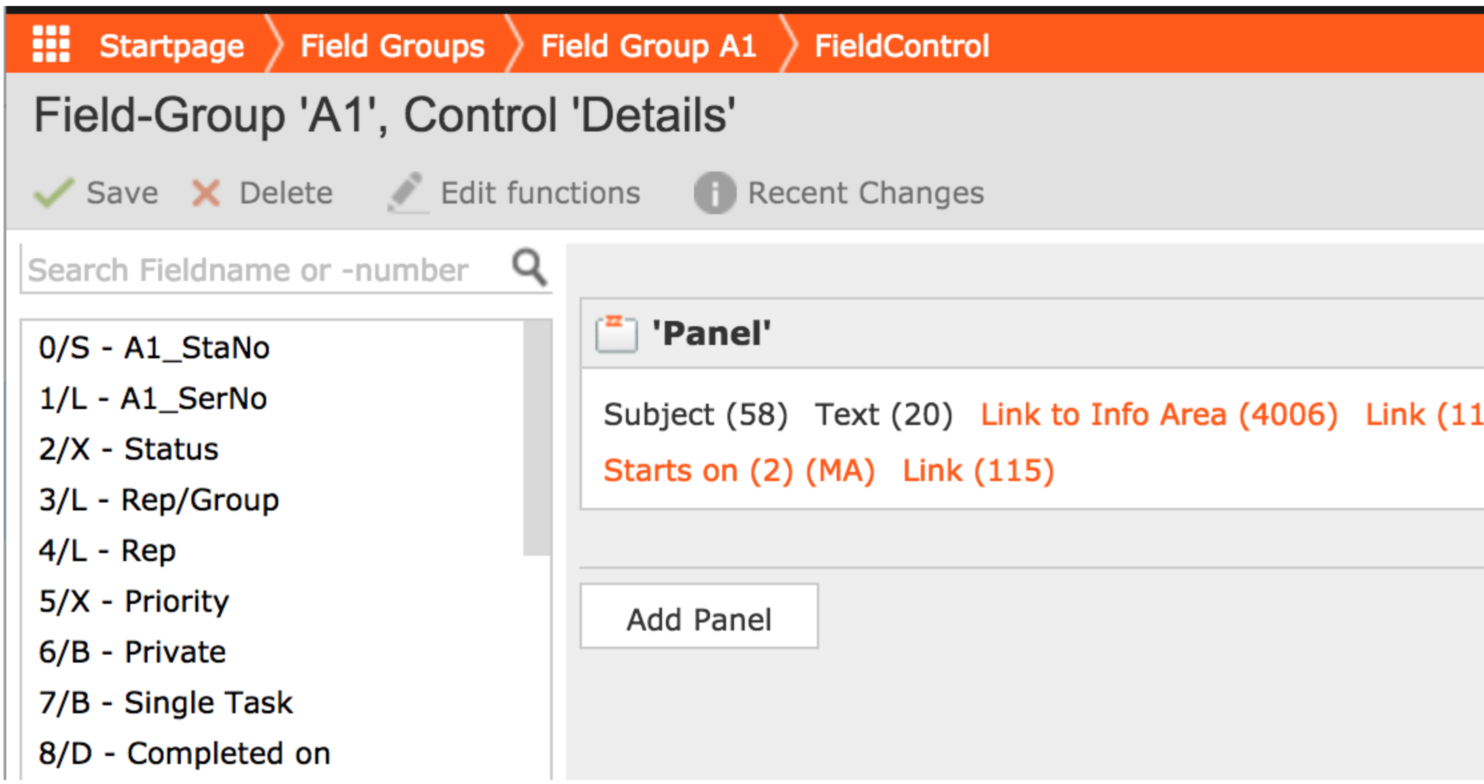
Info Area

Add a function name as `GenericLinkInfoArea` for field **Link to Info Area**, as below.

Numbers

Add Functions names `GenericStaNo` , `GenericSerNo` for **Link "StaNo"**, **Link "SerNo"** fields respectively as below.


Startpage
Field Groups
Field Group A1
FieldControl
Field-Group 'A1', Control 'Details'
Save
Delete
Edit functions
Recent Changes
Search Fieldname or -number
0/S - A1_StaNo
1/L - A1_SerNo
2/X - Status
3/L - Rep/Group
4/L - Rep
5/X - Priority
6/B - Private
7/B - Single Task
'Panel'
Subject (58)
Text (20)
Link to Info Area (4006)
Starts on (2) (MA)
Link (115)
Add Panel


Startpage
Field Groups
Field Group A1
FieldControl
Field-Group 'A1', Control 'Details'
Save
Delete
Edit functions
Recent Changes
Search Fieldname or -number
0/S - A1_StaNo
1/L - A1_SerNo
2/X - Status
3/L - Rep/Group
4/L - Rep
5/X - Priority
6/B - Private
7/B - Single Task
8/D - Completed on
'Panel'
Subject (58)
Text (20)
Link to Info Area (4006)
Starts on (2) (MA)
Link (115)
Add Panel

Non-Field Elements

Non-Field elements (LineFeed, Separator, Caption, Text) are not supported in CRM.pad to structure the output.

Combining Field Values

Learn how to combine field values.

You can combine the content of more than one field in the Details, List and MiniDetails control with the `ColSpan` field detail. For example, you can combine the First Name and Last Name fields to be displayed as Name.

To combine several fields:

1. Add the fields to the desired control (Details, List, MiniDetails).
2. Select the first field of the fields you want to combine and enter the number of fields you want to combine in the text field.

   For example:

   If you want to combine Street, Postal Code, City and Country to an address label, add these four fields to your control and set `ColSpan` to "1;2;1". In this case the field then consists of three lines: the first line contains the first field, the second one contains the next two fields and the third one contains the fourth field.

3. If you combine fields in View mode, the first field's **Label** & **Field Attributes** are applied. For example, if you combine four fields to an address label, the field's label is "Street", and if you combine two fields to a person's name, the label is "First Name". You can either hide the label with the **No Label** field detail, or you can define a field label.
4. You can also define a delimiter if more than one field is shown in one line. The following options are available:
   - "`2:-`" delimits with a minus.
   - "`4 :b /b`" delimits with the string " / " (each b is replaced by a blank allowing to trail blanks).
   - "`3 :bcb`" delimits with string " : " (each c is replaced by a colon).

- "`1 ;2 ;1:-`" delimits the two strings in the second line with a hyphen.
- "`3 :FormatText:my:2`" or `3:Text:my:1` combines three fields. The delimiter applied is a text from the `smartbook :my` text group. Wildcards (i. e. `{1}`, `{2}`, `{3}` or `%1`, `%2`, `%3`) can be applied and are replaced by the corresponding field values.

Note: Delimiters are only applied between two or more fields in the same line that contain values.

HTML Fields

Fields can be defined as HTML fields in Aurea CRM web that store text formatted in HTML.

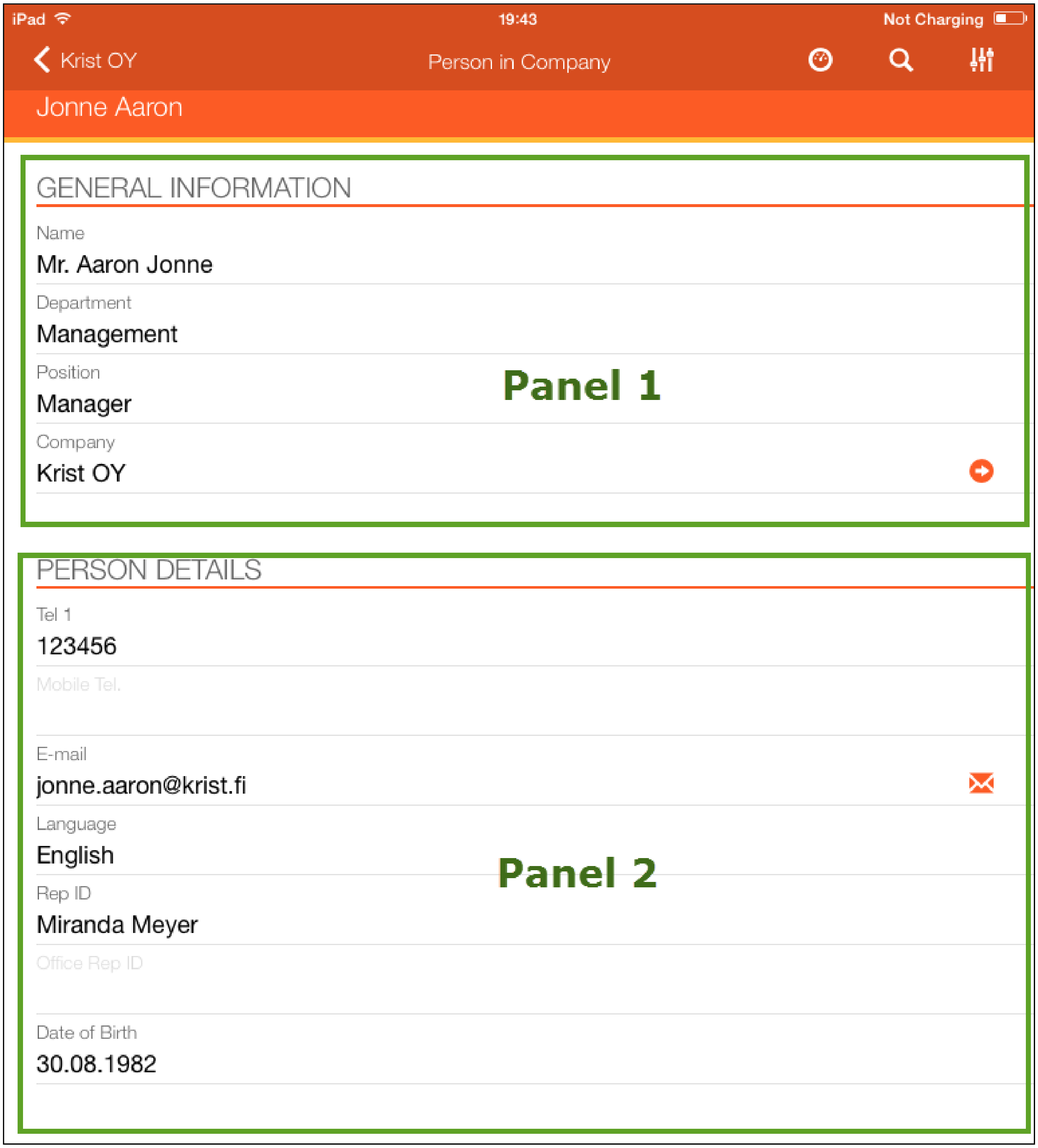
This text is also displayed in HTML format in CRM.pad. However, text that the user changes on the device loses the HTML format after saving it.

Panels

You can add panels and then add fields to these panels.

You can assign types to panels, if required, e. g. if you want to display data in a map.

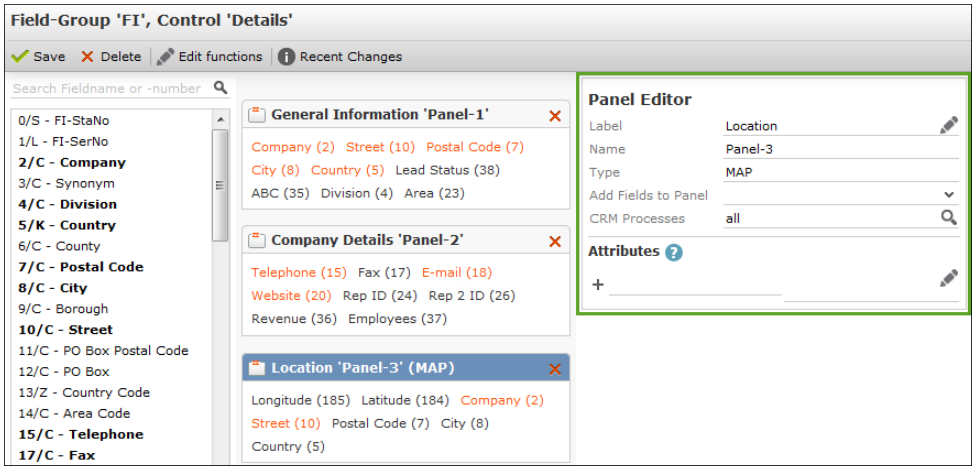
Panels, e. g. for a Person, are displayed in CRM.pad as follows:



The size of each panels is adjusted automatically to its content.

Fields defined for Details and Edit controls are organized in panels.

In the **Panel Editor** you can define the following:



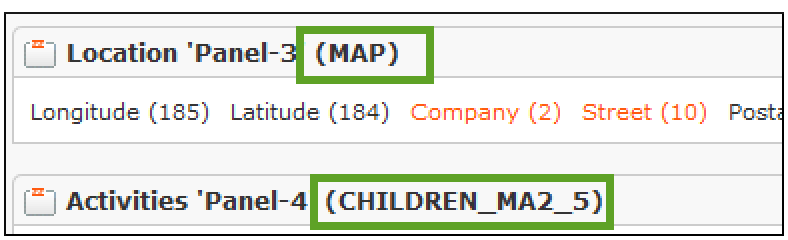
- **Label**: The text displayed as a subheader on the control.
- **Name**: The panel's name (i.e. the panel's ID).
- **Type**: You assign special types to your panel, see Panel Types on page 166.
- **Add Fields to Panel**: If you want to add the panel's fields to an existing panel, select the desired panel in the drop-down list. The current panel's fields are appended to the selected panel.
- **CRM Processes**: You can assign one or more CRM processes to the current panel.

For general information on how to handle panels, e. g. on how to add or delete them, see Configuring Panels in the Aurea CRM web Administrator Guide.

Panel Types

Learn about the panel types in aurea CRM.pad.

You can assign the following panel types to your panels, for example:



To assign a panel type the desired type in the **Type** field of the **Panel Editor**.

ANALYSES_<AnalysisName>

If you assign this type, the analysis with the name given in `<AnalysisName>` is executed and its result displayed in a list.

CHARACTERISTICS

Assign this type if you want to display contents of the Interest Group (45) and Interest (46) catalogs in a tab of an info area's Details view, see Displaying Interests/Characteristics in a Company's the Details View on page 333.

CHARACTERISTICSCONTEXT_<ContextMenu>

Assign this type if you have added Characteristics/Interests using additional info areas and you want to display the data in a panel on the Details view instead of a separate page. The action assigned to the specified context menu is executed within the panel area.

CHILDREN_<S&L>

Configure this parameter using the correspondent syntax: `CHILDREN_<S&L>#<LINKID>_<number>_<Swipe/NoSwioe>`.

Configure the parameter as follow:

- Replace `<S&L>` with the name of the desired Search&List configuration. If you assign this type, the current record's children referenced by the given Search&List are displayed in a sub-list in the Details view.
- Replace `#<LINKID>` with the linkID of the relationship you would like to add to the parameter. Add as many links as the relationship you would like to add.
- Add `_<number>`to limit the number of the displayed records under the desired **info area**.
- Use the option `_Swipe` or `_NoSwipe` to enable or disable the swipe of the children record headers on your device.

Note:

The options `_Swipe` or `_NoSwipe` refer to the children record headers and not to the **info area** panel that have swipe enable by default.

You need to set a record limit to allow the parameters _Swipe or _NoSwipe to take effect.

DOC

The user can download all documents (D3 records) that are linked to the record. If iPad supports the document type, the user can open them on the device. The documents are displayed in a list in the app.

If you assign this type, linked documents are available in a panel in the details view.

DOC.IMG

Assign the `DOC.IMG` type if records are linked to images (files in the following formats: jpg, jpeg, png, gif, tiff, tif, and bmp). The images are displayed as thumbnails in the application. Images are downloaded automatically, when the user is in online mode.

By default, the number of documents/images is limited to 100 for performance reasons. You can further limit the number of documents by adding the desired number, e. g. `DOC.IMG_<10>`.

DOC.NOIMG

If you have defined a tab for image files (`DOC.IMG`), you can add another tab/panel and assign the `DOC.NOIMG` type. Documents are displayed in lists.

By default, the number of documents is limited to 100 for performance reasons. You can further limit the number of documents by adding the desired number, e. g. `DOC.NOIMG_<10>`.

DOCSEARCH

If you assign this type to a panel, you define a document group.

Syntax: `DOCSEARCH_<Search&ListName>_<MaxResults>`

Example: `DOCSEARCH_D3FI_10`

Ensure that the `D3DocData` (or the `D1DocData`) **Field Group** is specified in the given Search&List configuration. Both field groups are contained in the `update.tablet` configuration.

The following **Functions** are defined by default for these field groups:

Field	Function
MIME type	`MIMETYPE`
Title	`Title`
Length	`Length`
Date	`Date`
Doc.Class	`Class`
Upd	`UpdDate`
Updated at	`UpdTime`

If you want to display a date or a text (e.g. the update date and the company name), replace the existing functions as follows:

Field	Function
Upd	`DisplayDate`
Company	`DisplayText`

Note: The company field is not part of the default configuration, you must add it to the field group.

GRID

By default, data is displayed view in one column for each panel in the Details view. You can define that data is displayed in two columns (= in a grid). If the Grid view is enabled, empty fields are always displayed.

You can also define a placeholder that you can add to the Grid view (e.g. if you want that a specific field is always displayed as the first field in a line, you may want to define an empty placeholder). Add a field to the desired panel and set the `Empty` field attribute. This field is then "displayed" without label and without content.

INSIGHTBOARD / INSIGHTBOARDH / INSIGHTBOARDV

If you assign this type to a panel, the Insight Board is displayed in Expand view. Syntax: `<DisplayType>_<Name>_<MaxVisibleItems>`:

- `DisplayType`: Defines the alignement of the tiles:
  - `INSIGHTBOARD` or `INSIGHTBOARDH`: Horizontal alignment and scrollable tiles.
  - `INSIGHTBOARDV`: Vertical alignment
- `Name`: The name of the Insight Board's context menu.
- `MaxVisibleItems`: Defines the number of the tiles displayed in the first line.

MAP

`MAP`, see Adding a Map on page 171.

MULTIMAP_<InfoArea>

If you assign this type, you can add a map displaying more than one record to a Details control.

Replace `<InfoArea>` with the desired info area or Search&List configuration. If you assign this type, the children of the given info area's current record are displayed in an additional list in the Details view.

You can

- limit the number of the displayed records by adding `_<number>`, e.g. `MULTIMAP_FI_5`.
- specify the link if you do not want to use the default link, e.g. `MULTIMAP_<InfoArea>#<LINKID>_<MAXRESULTS>`.

PARENT_<InfoArea>_<PanelNo>

If you assign this type, the panel with the number given in `<PanelNo>` is displayed in a list. Ensure that the given panel exists in the linked info area's (i.e. the parent record's) Details control.

You can also specify the link if you do not want to use the default link, e.g. `PARENT_<InfoArea>#<LINKID>_<PanelNo>`.

QUERY_<QueryName>_<No>

If you assign this type, the query specified in `<QueryName>` is executed and it's result displayed in a list. The number given in `<No>` limits the number of the displayed hits.

REPPARTICIPANTS / LINKPARTICIPANTS

If you assign `REPPARTICIPANTS_<S&L>` and `LINKPARTICIPANTS_<S&L>`, you can display or edit participants of the Search&List configuration specified under `<S&L>`, >> Displaying Participants.

If no Search&List configuration is found, the field group with the same name is applied (if exists).

WEBCONTENT

If you assign `WEBCONTENT_<MenuActionName>`, a web page is displayed. The web page's URL is defined in the `WebView` action assigned to the menu action, see Displaying a Web Page on page 236.

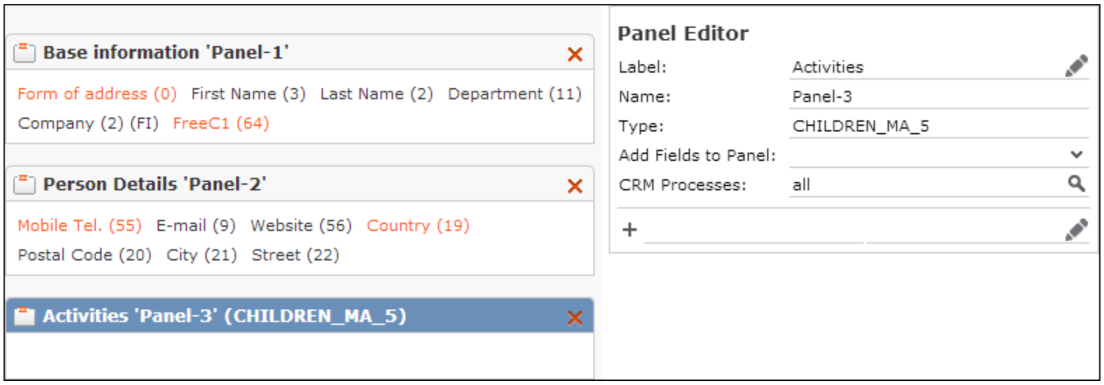
Defining Sub-Lists (Related Data)

You can define that data from child info areas is also shown in Details or Edit view by adding additional panels to the Details or Edit control.

If you add an Edit control, the user can edit the child record's data as well.

To define sub-lists:

1. Add a panel.



2. Define a Label to be used as sub-list header. If undefined, the header text of the assigned Search&List configuration is applied.
3. Assign the `CHILDREN` type, see CHILDREN_<S&L>:

   If the given Search&List configuration (`<S&L>`) cannot be found, the field group with the same name is applied as fallback.

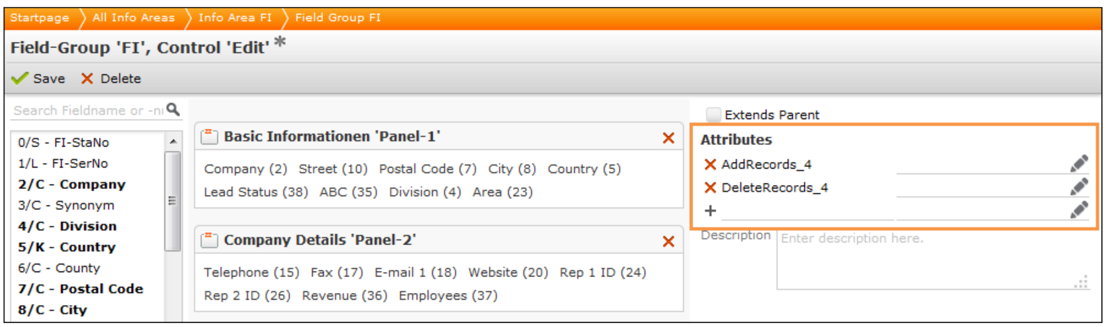
   If `<maxRows>` is omitted, the value of the `Search.MaxResults` Web Configuration parameters is applied, see Search.MaxResults.

Examples:

- CHILDREN_MA_10: Defines a sub-list containing Activity records using the MA Search&List configuration and loading 10 records at a time.
- CHILDREN_KM#2_10: This defines a sub-list containing Ticket records (using the KM Search&List configuration) where the current person is entered as contact person. Again 10 records are loaded at a time.

Add a new **Attribute** in the Edit control to define whether the user can add or delete records:

- AddRecords_<tabId>: If set to "true", the user can add records.
- DeleteRecord_<tabId> If set to "true", the user can delete records.



Adding a Map

You can configure the address information to be displayed in a map, in Details view, by adding an additional panel to the Details control.

There are two options to show items on a map:

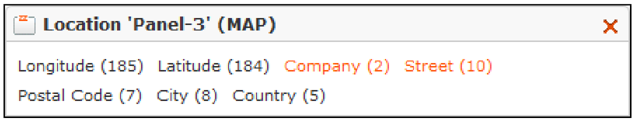
- Based on geo-coordinate fields. These fields must first be added to the Aurea CRM data model, see the *Aurea CRM win Administrator Guide*.

Note: Aurea recommends that you always provide the latitude and longitude fields to capture the geographic coordinates. This is especially useful if you wish to show all annotations in a map, see Showing All Annotations in a Map.

- Based on an address.

Configuring either option is almost identical:

1. Add a panel to the desired Details control and assign the MAP type.
2. Add the **Longitude (x)** and **Latitude (y)** fields for the two geo-coordinates.



Note: You must enter the Longitude (x) field **before** the Latitude (y) field. If you want to show the records based on the address instead, you need not add the geographic coordinate fields, you can add two dummy fields (e.g. FreeDec1 and FreeDec2) as placeholders instead.

3. Add a field that is used as header text in the Map view. For example, the Company field.
4. Add the address fields:
   - If you use the geo coordinate fields, the address is displayed below the header text.
   - If you display the address based on the address fields, they must be added to the panel. It is best practice to add the Street, ZIP and City fields and then combine them, see Combining Field Values on page 163.

Note: Apple-Geocoder cannot display addresses correctly that contain a hyphen and spaces in the building's number. For example, the address Operngasse 17 - 21 is only displayed correctly when it is entered in the database without spaces: Operngasse 17-21.

Panel Attributes

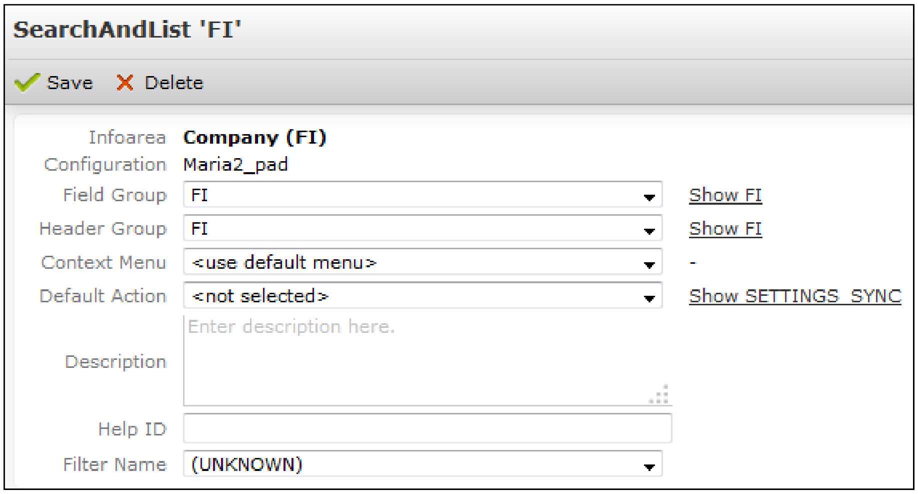
Learn about the use of panel attribute.

ParentInfoImage" = "<ImageName>

Use this panel attribute to define which image is displayed. The image must already have been registered in CRM.designer, see Images on page 220.

Search&List Configuration

The Search&List configuration defines the page that displays the search and the search results.



A Search&List configuration consists of the following items:

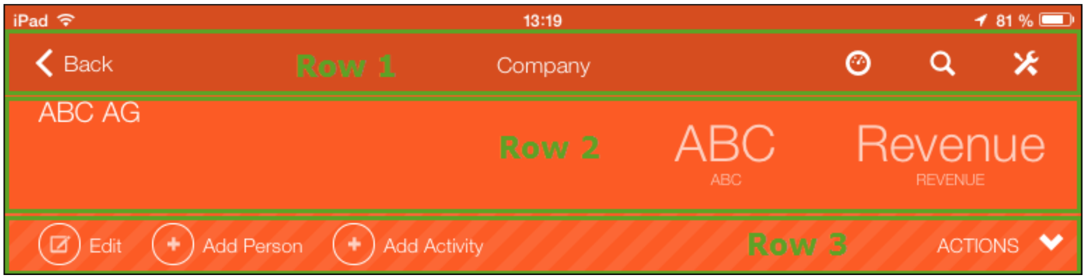
- **Field Group**: Defines the layout of the search mask and list rows. The Search and List controls of the specified field group are used, see Search Control on page 146 and List Control on page 142.
- **Header Group**: Defines the header group of the headers used on the page, see Headers on page 173.
- **Context Menu**: Not applicable für CRM.pad.
- **Description**: A description only available in CRM.designer.
- **Help-ID**: Not applicalble for CRM.pad.
- **Filter Name**: The given filter limits the number of search results.

Headers

Learn about the use of header in CRM.pad.

Each page in CRM.pad contains a header that can contain up to three lines displaying the following:

- First row (always displayed): Info area name, navigation button (e.g. Back) and the Home, Search and Control Center button.
- Second row: Tiles (if available).
- Third row: Quick actions and Actions menu.

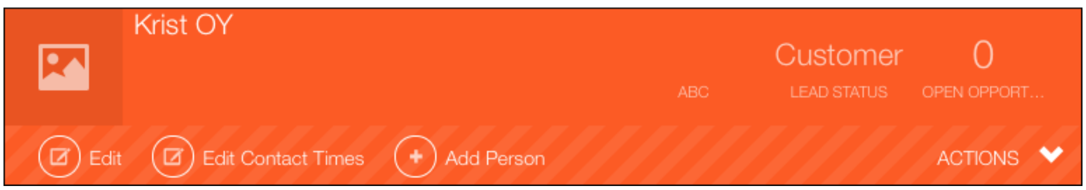


Two types of headers are available:

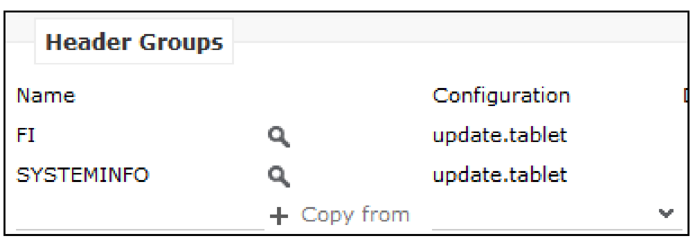
- Standard headers providing the standard functions for the user. Standard headings are info area-specific.
- Special headers providing different/additional functions. They can be info area-specific or info area-independent.

Header Group

You can define the contents of the headers displayed in CRM.pad, for example:



Info area-specific headers are grouped in the **Header Groups** area of an info area.



If no header group was defined, the corresponding header from the DEFAULT header group is applied. In the `update_tablet` configuration, the default header is the header with the info area ID as name, e. g. `FI` is the default header group for the Company info area.

If you add headers for specific functions, use the following naming conventions:

`< infoAreaID >.<Name>`

Examples:

`MA.Calendar`

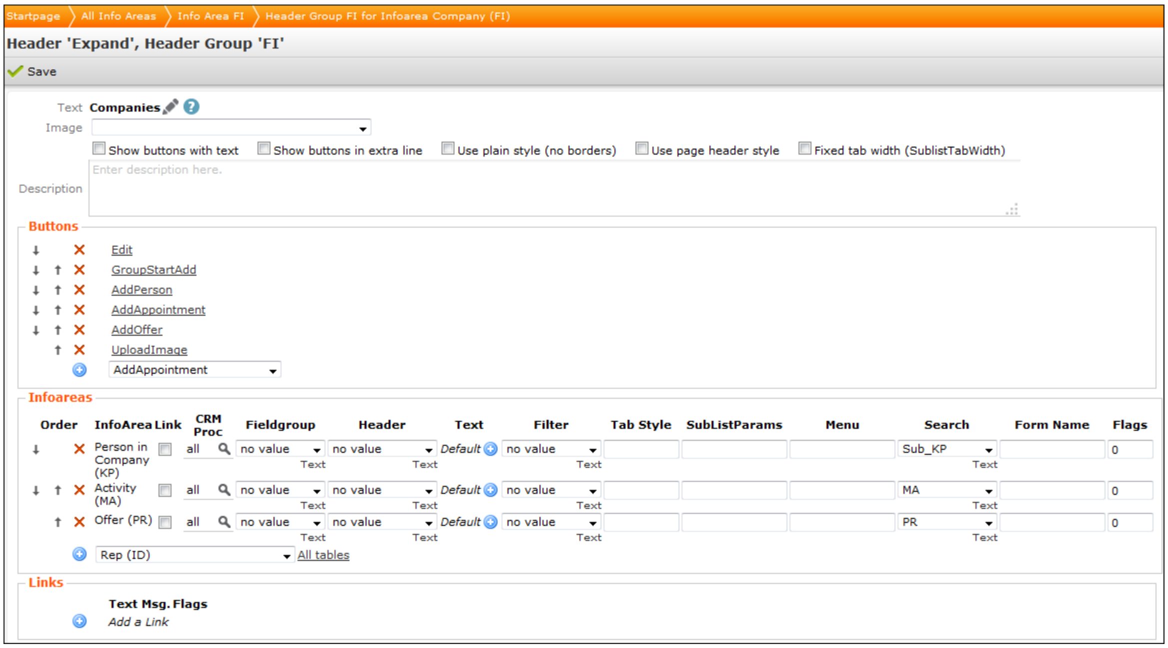
`AU.SerialEntry`

Defining a Header

To define a header group:

1. Switch to the desired header group.
2. In the **Headers for Header Group [info area]** window, click on besides the desired view, e. g. **Expand.**

   The header group of the selected header (in this example the Expand header) is displayed.



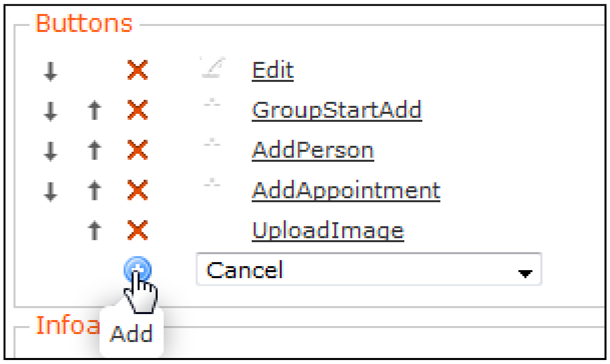
Only the following items are used in CRM.pad

- In the upper area you can define the **Text**, i.e. the text displayed in the header, whether buttons are displayed with text, what format is used, etc.
- In the lower area, you can define the following:
  - **Buttons**, see Adding Header Buttons.
  - **Infoareas, see** Adding Info Areas (Tabs).
  - **Links**: Not available in CRM.pad

Adding Header Buttons

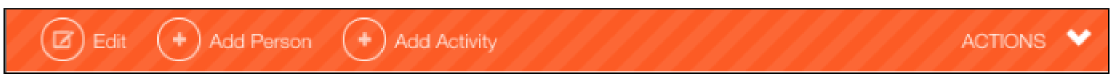
To define which functions are available in the application add the desired buttons:

1. Select a button from the drop-down list (the following example shows the FI header):



2. Click on (**Add**).
3. You can click on a button to change its settings, see Buttons on page 63.
4. You can change the order using the and buttons.

The buttons are displayed in CRM.pad as follows:



Note: If no quick actions were defined, the first three buttons are displayed as quick actions on the left of the header's function row. Grouping is ignored when no quick actions were defined. All other actions can be accessed in the **Actions** menu. If quick actions were specifically defined, all buttons defined in the header are available in the **Actions** menu, see Quick Actions.

Adding Info Areas (Tabs)

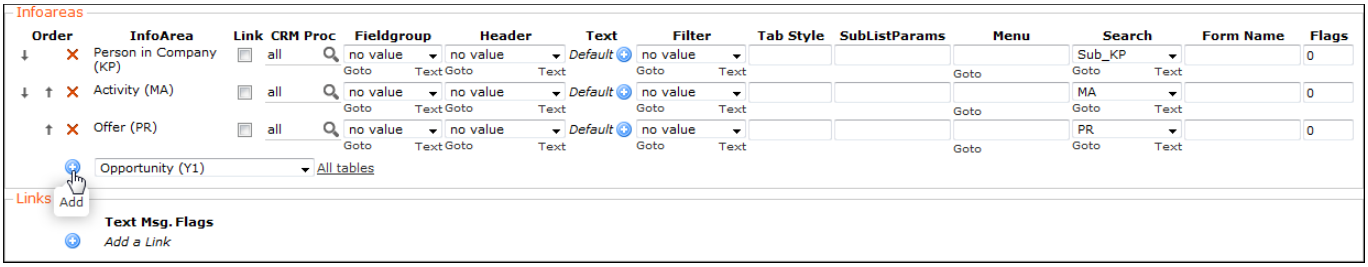
You can define that related data is displayed in a separate tab in CRM.pad by adding the desired info area in the header's **Infoareas** area.

Note: You can only use info areas with a data model link to the info area to which the header definition applies.

You cannot add a tab for info area-independent headers.

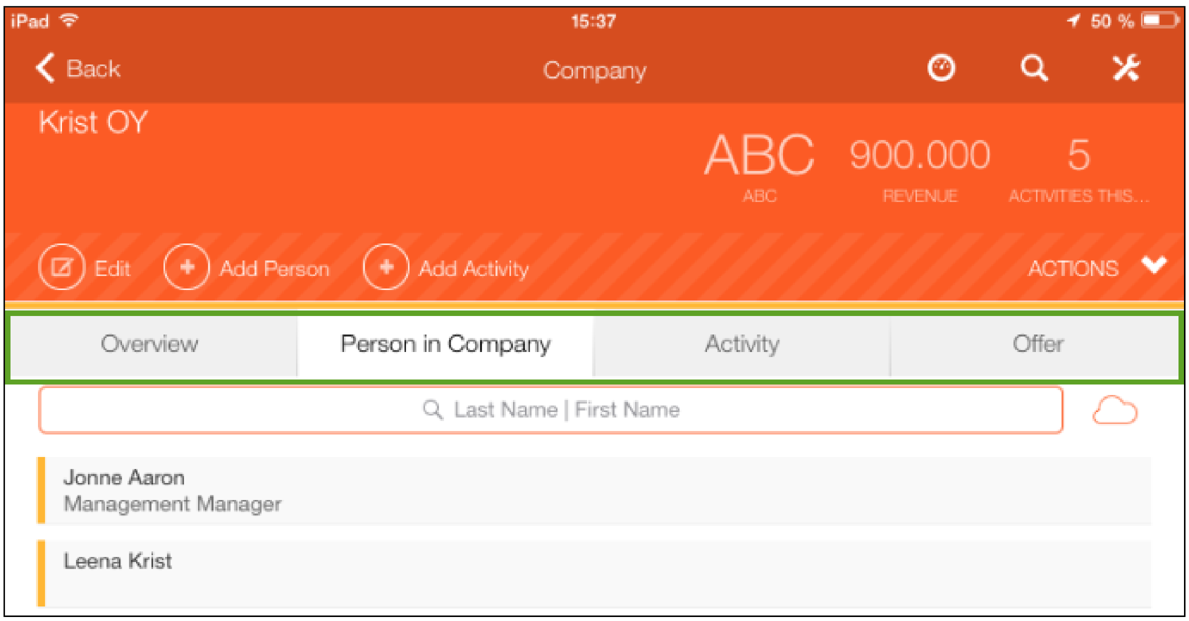
To add an info area (tab):

1. Select a button from the drop-down list (the following example shows the FI header):



2. Click on (**Add**).
3. You can change the order using the ↑ and ↓ buttons.

The related data is displayed in CRM.pad as follows:



The user can tap on the desired tab to display the data.

Alternative Header Captions

Following texts of the header group referenced in the Expand configuration are displayed in CRM.pad:

- The defined header label (a header's **Text** field). As a fallback, the info area name is displayed if no header label was defined.
- Edit view: You can define that a different text is displayed in Edit view: Specify the header group holding the desired text in the applied Expand configuration.

Header Tiles

You can define that up to three tiles are displayed in the header.

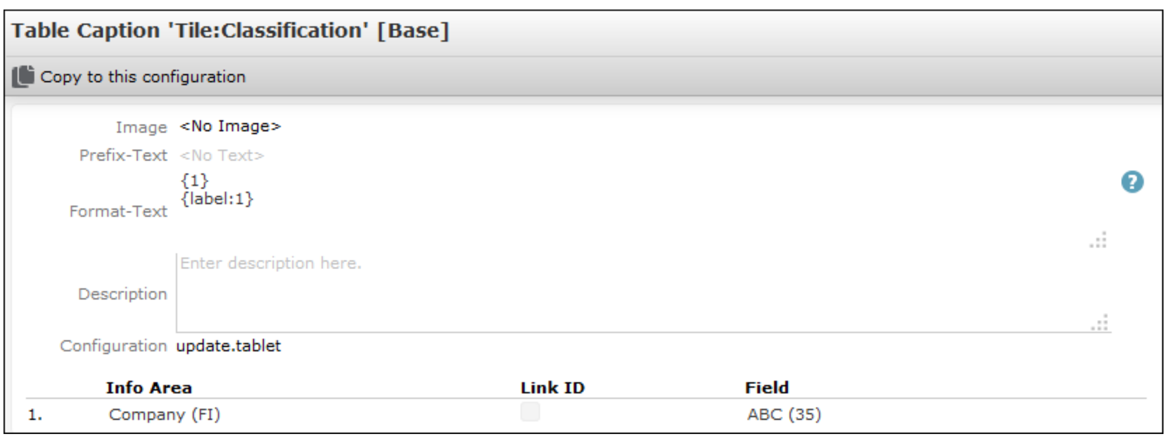
You need to define the following:

1. A context menu with the `Tile:` prefix in the name, e. g. `Tile :Company` (included in the default configuration).
2. Add the context menu actions (max. three) for each tile you want to display. The names of the context menu actions also must begin with the prefix `Tile:`, e.g. `Tile:CompanyClassification` (assigned to the `Tile:Company` context menu in the default configuration)

   You can only assign the following two actions (all the other Tiles: <name> included in the default configuration are only used in CRM.mobile).

   - `Tiles:Record`, see Tile:Record on page 138.
   - `Tiles:Search`, see Tile:Search on page 139.

3. Add a table caption specifying the fields that are displayed in the tile, e.g. the `Tile :Classification` table caption (included in the default configuration).



4. Assign the context menu as control attribute (see Control Attributes on page 156) to the desired field group, e.g. FI field group:

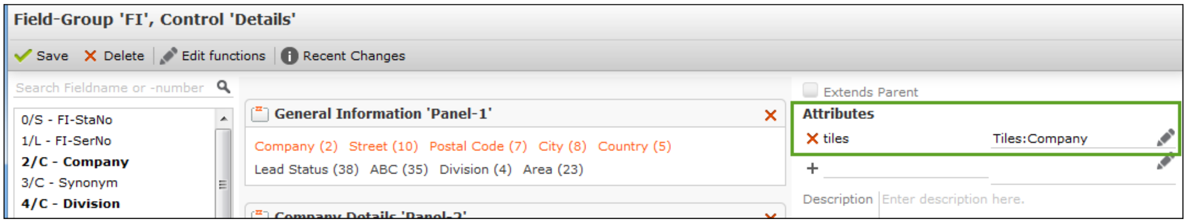


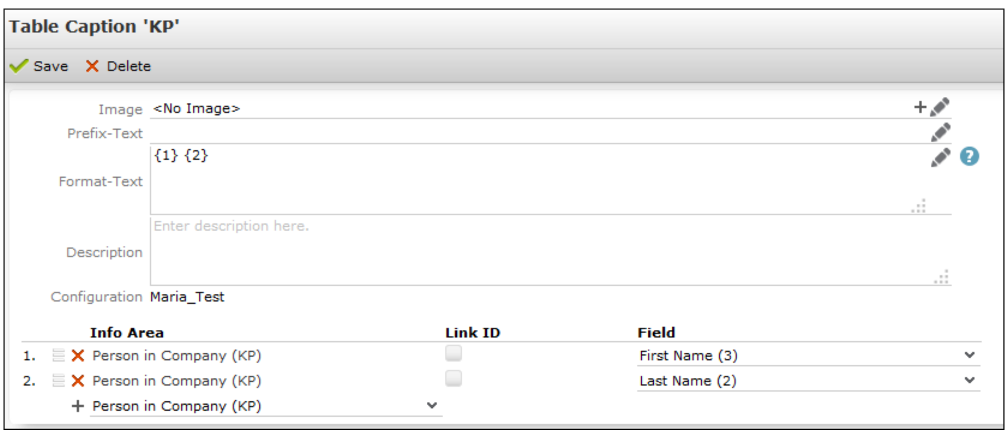
Table Captions

You can define with a table caption which data of a record is displayed in its header, e. g. first and last name for a person or the company name.

To define a table caption:

1. You can:
   - Switch to the desired info area and click on `Table Captions`.
   - Click on `Table Captions` (**Views** area) on the CRM.designer start page

   Example:



2. You can define the following:
   - **Image** (optional): Name of an image that is displayed on the left of the table caption. The image must already be registered, see Images on page 220.
     - Click on + (**Add**) to select a fixed image.
     - Click on ✎ to define conditions for an image to be displayed.

     Example:

     `CAT_AQ _{ 1}` refers to the first field defined under **Field**.
   - **Format-Text:** You can define which fields added under **Field** are displayed in which order using placeholders. For example, in the screenshot above, First Name (1.) and Last Name (2.) were added. `{1} {2}` displays John Doe, `{2}, {1}` displays Doe, John.

     If you do not define a **Format-Text**, no Table caption is displayed.

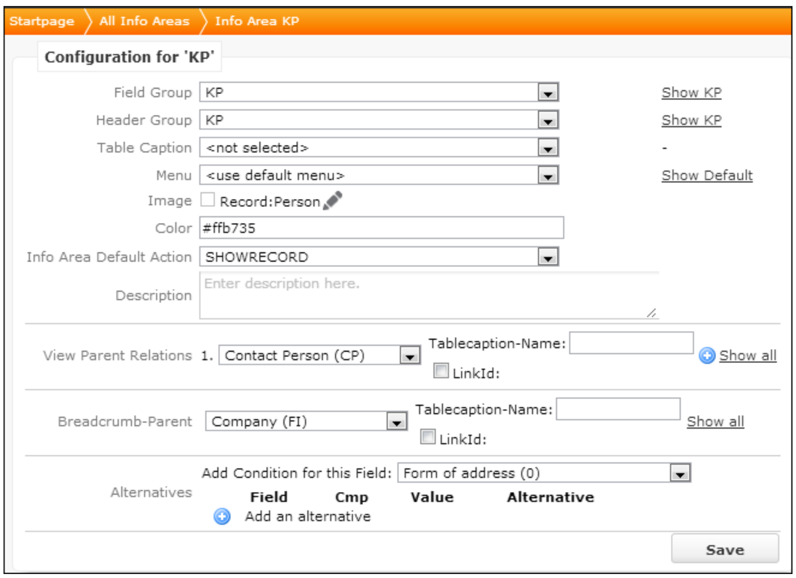
Note: If you want to display **Format-Text** without formatting (e.g. display numbers without digit grouping symbol), enter the placeholder as follows `{r1}`

- **Description:** The given description is only available in CRM.designer.

3. In the lower area you can add the **Fields** you want to display in the table caption. The fields are numbered automatically. Use these numbers in the placeholders in the Format-Text field, e.g. `{1}` `{2}` etc.

Expand Configuration

The Expand configuration of an info area contains the field group and header group that configure the Details view of a record.



The following options are available:

- **Field Group:** Defines which Details control is used.
- **Header Group:** Defines which `Expand` and `Expand.QuickActions` header is used.
- **Table Caption:** Defines the record's table caption.
- **Color:** The info area's color. Used for Record tiles, in lists and on Details views. If undefined, light gray is used.
- **Info Area Default Action:** Defines the default action that is executed when a user taps on a link.
- **View Parent Relations**: Not applicable in CRM.pad.
- **Breadcrumb-Parent:** Not applicable in CRM.pad.
- Use **LinkId** to specify another than the default link, see Link IDs in the *Aurea CRM web Administrator Guide*.

Virtual Info Areas

Virtual info areas are defined in Aurea CRM web.

Therefore, see the Aurea CRM web Administrator Guide on detailed information on how to configure and use them.

Only the function to display different data based on virtual info areas for one info area is available in CRM.pad.

Example:

You can define three different virtual info areas for the FI info area called VFI1, VFI2, and VFI3. You can assign different Details pages to each virtual info area. You can then display these three Details pages within the same FI list.

To use virtual info areas:

1. Create a `RecordView` action and assign `<ID>[Identifier]`:

   `<ID>` is replaced with the record's info area ID (FI, VFI1, etc., `[Identifier]` can be replaced with a string that determines the Expand configuration you want use for this feature.

2. Create a new context menu action and assign the `RecordView` page call. Set the `ConfigName` to `# #[ Identifier]`.
3. Create a separate Expand configuration for each info area named e.g. `FI[ Identifier]`, `VFI1[Identifier]`, `VFI2[Identifier]`, and `VFI3[Identifier]`, and assign the desired field group, header group, image etc. to each Expand configuration.

   If you do not define Expand configurations for the virtual info areas, the Expand configuration from the physical info area is applied.

4. Assign the previously created context menu action as default action to the corresponding Search&List configuration.

Record Selectors

You can copy values from other records using a so called "RecordSelector". Two action templates are available for this purpose:

- `Selector` for record selectors of type address, see Selector on page 126.
- `RecordSelect` for record selectors of type record, see RecordSelect on page 121.

Note: Edit Trigger Filter for Record Selector is not supported for Aurea CRM.pad.

Record Selector Type Address

This record selector allows the user to select a record and then copy values from this record to the current record.

For example, several addresses can be entered as additional addresses. A user can select the desired address and then copy the address data to the current record (e. g. to an offer record).

For this record selector type you need the following:

- The List controls of the source and target info areas must contain the fields you want to copy. These fields need the copy functions assigned.
- The Edit control of the target info area defines the fields that are displayed in the record selector. Add a new tab and add a FreeC field for each field you want to copy. Add an additional FreeC field to call the record selector. Assign the required functions to these fields, see Required Functions.
- A Search&List configuration for the target info area.
- A context menu action with the `Selector` action assigned, see Selector on page 126.

Note: Currently, no default "copy field group" exists in the `update.tablet` configuration. Define a field group as needed.

Required Functions

In the Edit control of the target field group, assign the following functions to the FreeC fields:

Field	Function	Label
First FreeC field	`=`	Calls the record selector.
All other FreeC fields	`Copy1` to `Copy<n>`	Enter the copy function and number it sequentially.

In the List control of the source info area:

Function	Description
`Name`	Add this function to the field that you want to display as the record selector's heading.
`Copy1` to `Copy<n>`	Add this function to the field you want to copy to the record selector.

Required Field Attributes

You need to set the following field attributes:

- For the field that calls the record selector (i.e. that has = assigned as **Function**):

  Don't save

- For fields that you want to copy but are not be displayed in the record selector:

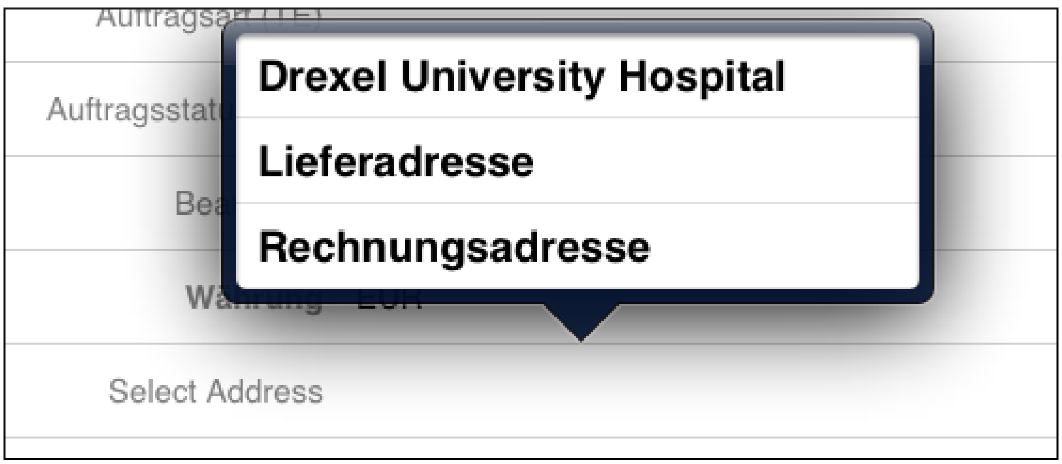
  Read-only

- If the result of a record selector contains more than one record of the same name, by default only the first hit is displayed. To show all hits of the same name enter the following option in the **Select-Function** field attribute:

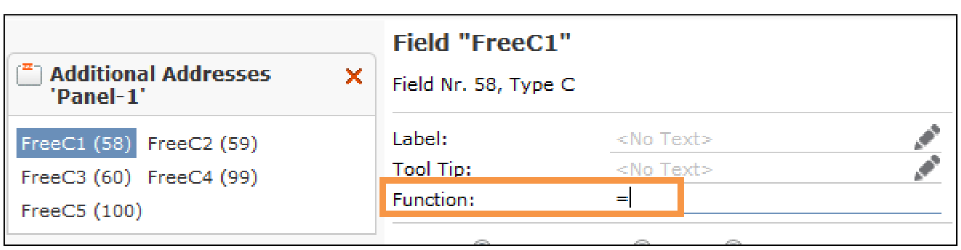
  ```
  {"DuplicateNames":true}
  ```

Example: Record Selector of Type Address

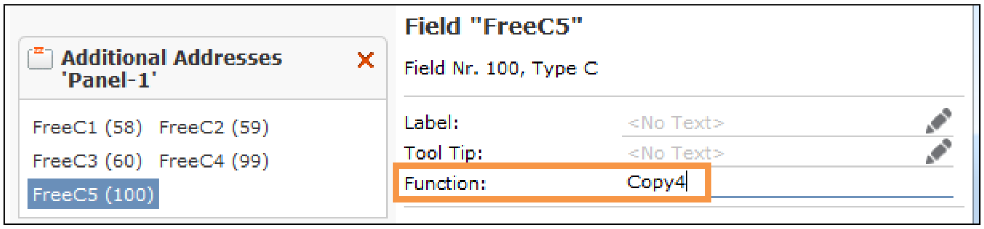
The following example shows how to define that a user can select one of the available addresses from the Company (FI) or from the Additional Addresses (ZA) info areas by tapping on a field in the Offer (PR) info area. In this example, the FreeC fields are used to copy the desired values.



1. Add FreeC fields (e.g. FreeC1 to FreeC5) to the Edit control of the Offer (PR) info area's PR field group. These fields are needed to display record selector. Add one additional field that is used to start the record selector.
2. The FreeC1 field is used to fire the record selector, therefore, enter = into the **Function** field and set the required field attributes, see Required Field Attributes.



3. Assign `Copy1` to `Copy[n]` as **Function** for the other FreeC fields:



4. Add the FreeC fields to the Details control of the PR field group.
5. Add the following fields to the List control of the desired field group of the PR and the ZA info area and assign the corresponding **Functions**:

   Both info areas:

Field	Function
Street	Copy1
Postal Code	Copy2
City	Copy3
Country	Copy4

   ZA info area:

Field	Function
Type	Name

   PR info area:

Field	Function
Company	Name

6. Define a Search&List configuration (e.g. `FIAddress`) for the two field groups added in the previous step.
7. Add a context menu action (e.g. `PRAddress)` and assign the `Selector` action, see Selector on page 126:

   For this example, you need to define the following input arguments as follows:

Input Argument	Value
Type	Address
SearchAndListName	FIAddress
ListConfigs	ZA
LinkInfoArea	FI

Record Selector Type Record

This record selector type copies field values from any info area to the current one.

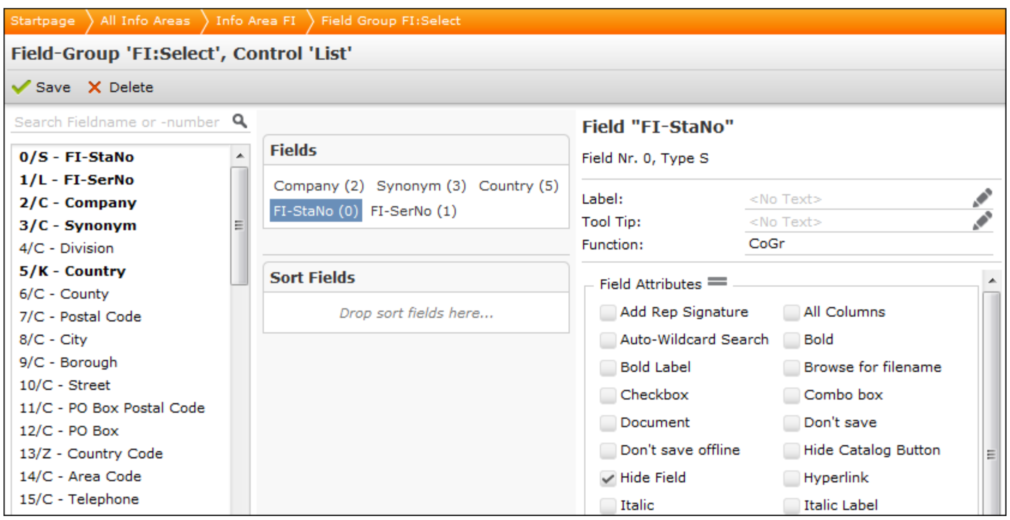
The user can search for a specific record and copy values by selecting the record. You can also use this function to link records.

The following example shows how to define that different persons or companies are linked to relationships:

1. Define a List control for your source field group in FI and KP. The source field group is the field group from which the values are copied from and defines the fields that are shown in the Record Selector's result list. If you also want to copy fields that are not be displayed in the list (e.g. FI-StaNo, FI-SerNo, KP-StaNo, KP-SerNo.), enable **Hide Field** in the **Field Attribute** area.

   - For the FI info area:

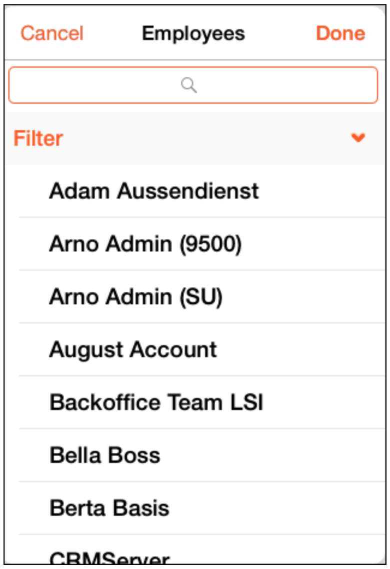
     1. Define a List control, in the following example FI:Select.



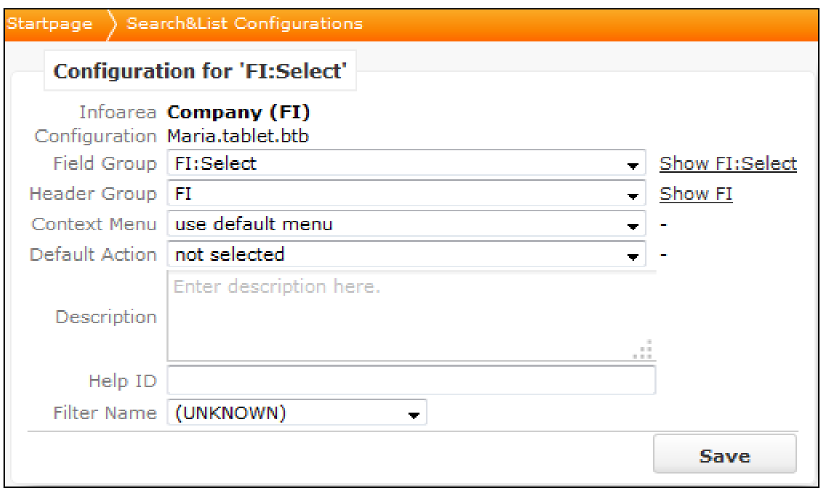
     2. Enter the following functions in the Function field of the following fields:

Field	Function
Company	Company
FI-StaNo	CoGr
FI-SerNo	CoNo

     3. In the Search control of the same field group you can define the fields that are searched in the database (i.e the fields displayed in "ghost text" in CRM.pad's search field).



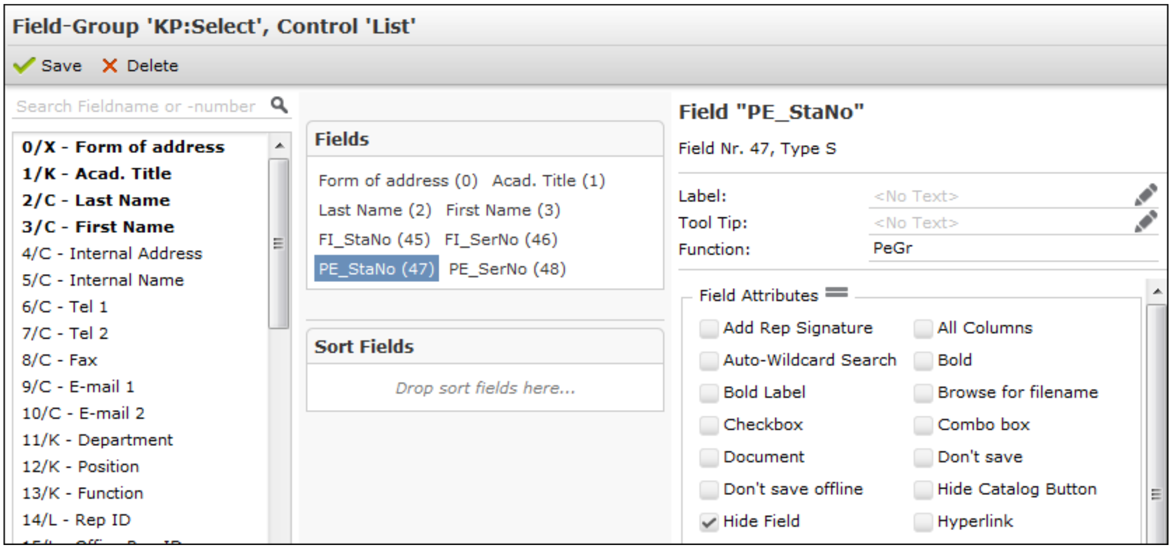
4. Add a Search&List configuration for your source copy field group:



The name of this Search&List configuration is used in the record selector's definition.

- Define the same items for the KP info area:

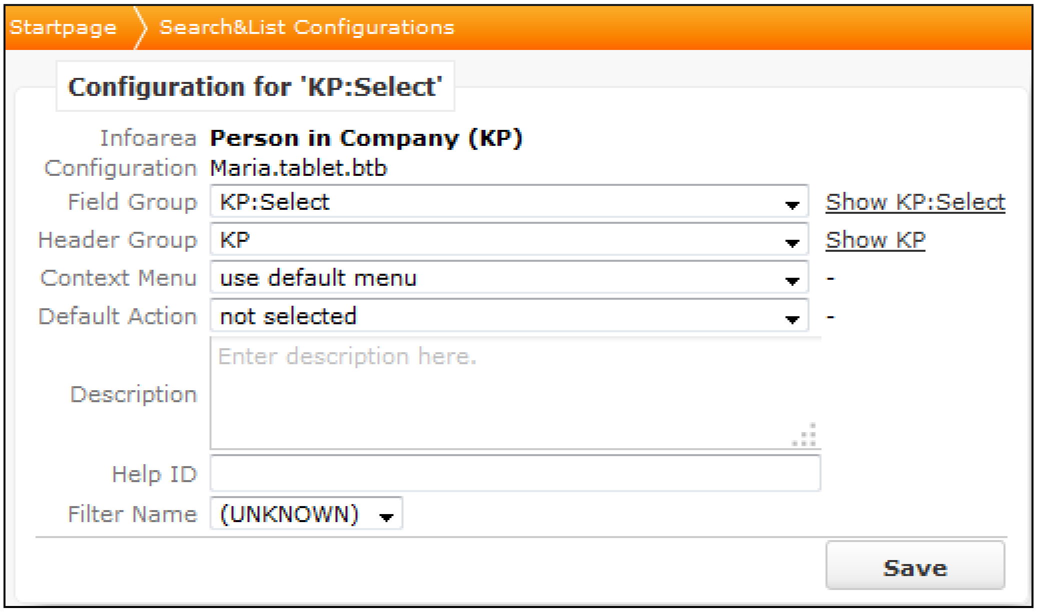
1. Define a List control, in the following example KP:Select.



2. Enter the following functions in the Function field of the following fields:

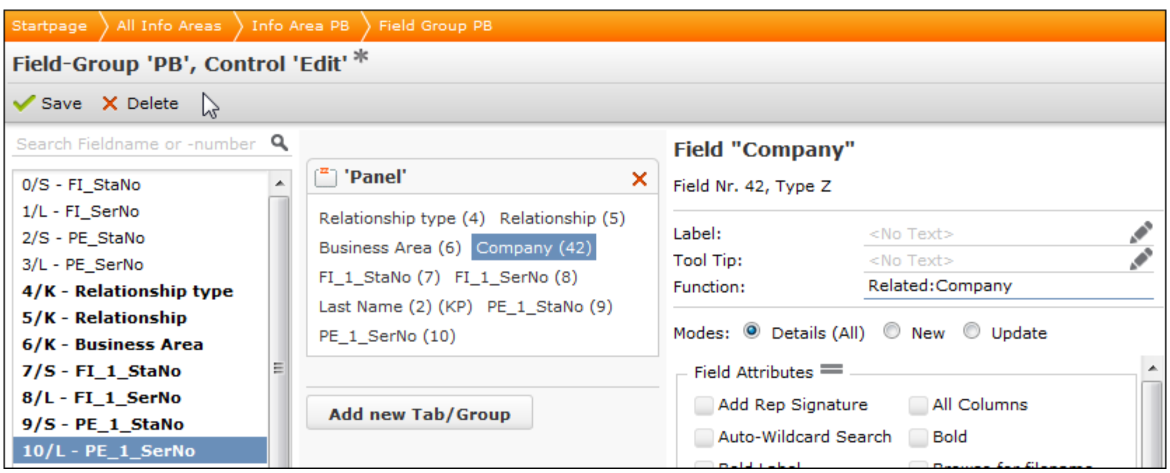
Field	Function
Last Name	LastName
FI-StaNo	CoGr
FI-SerNo	CoNo
KP-StaNo	PeGr
KP-SerNo	PeNo

3. Search&List configuration:



2. Define the Edit control of the target field group. The user can call the record selector from this control.

   In this example, the target field group is the PB field group of the PB info area:



3. Enter the following functions in the **Function** field to define the field values that are copied from the source field groups. Use the following syntax:

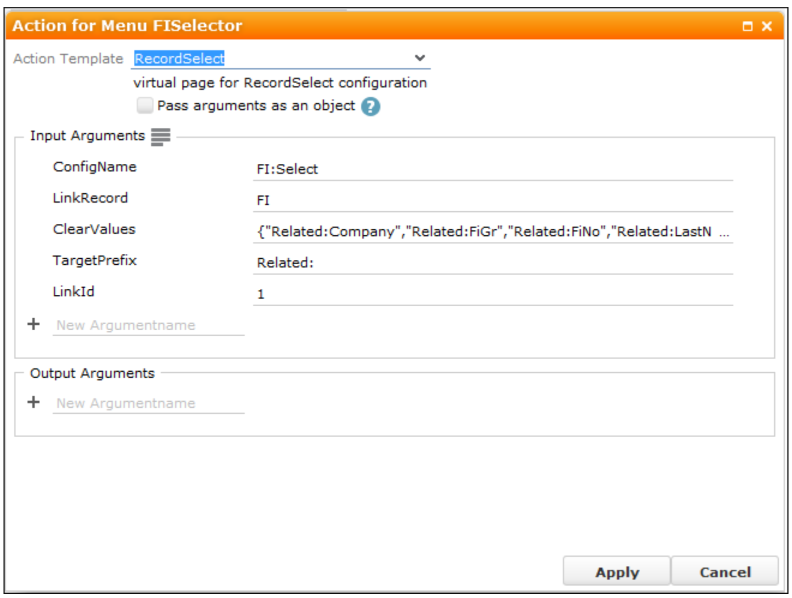
   ```
   Related:<FunctionName>
   ```

   Replace `<FunctionName>` with the **Function** ame you entered in the corresponding source field group.

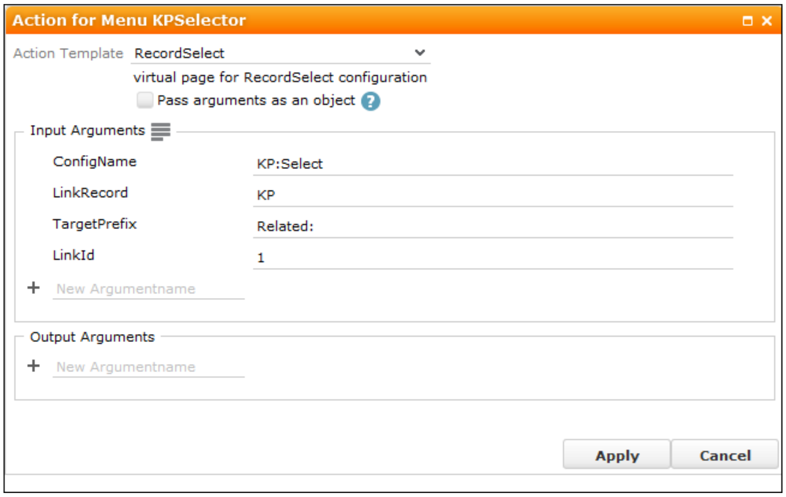
Field	Function
Company	Related:Company
FI_1_StaNo	Related:CoGr
FI_1_SerNo	Related:CoNr
Last Name	Related:LastName
PE_1_StaNo	Related:PeGr
PE_1_SerNo	Related:PeNo

4. Define the record selector by adding two context menu actions (one calling the record selector for FI, the second one for KP) and assign the `RecordSelect` action, see RecordSelect on page 121.

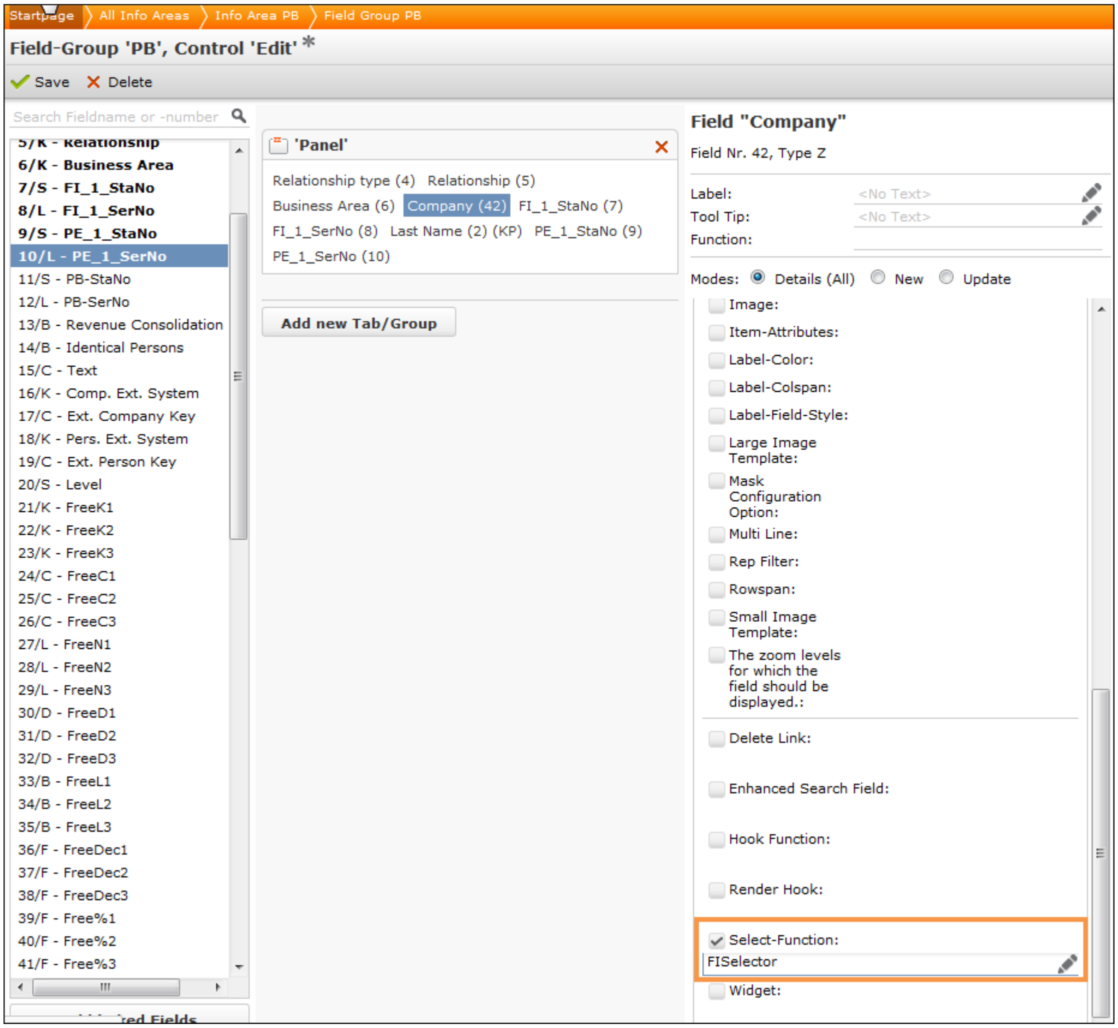
   - Context menu action for companies (FI):



   - Context menu action for persons (KP):



5. Switch to the Edit control's field group (you previously created) of the target info area (in this example, PB). Enter the desired context menu action name in the desired field's the `Select-Function` field.



Links

Links between parent and child record are automatically set in online mode.

In offline mode they are set when the user saves a record and:

- The key fields (StaNo and SerNo) exist for a records, these fields are filled when the link is set.
- Fields that exist in the CRM.pad data model in the linked info area and in the current info area are also filled.
- Fields that exist in the current info area but are links in the linked info area are also filled.
- Fields that are either entered or filled by the three options previously mentioned the link is also filled.

Example: If a KP link containing FI fields exists, the FI link is also set automatically.

Indirect Links

You can define indirect links, e.g. link KP (Person in Company) and VF (Sales Area) using the FI-StaNo/FI-SerNo fields. The VF fields then are not queried anymore using the KP link, but using the direct FI link.

To link indirectly:

1. Add the child fields to the desired field group control of the parent info area.

   Example: Add the VF fields to the desired KP field group control.

2. Define the fields' **Exended Option** using the following JSON syntax:

   `{"parentLink":"<Indirect Link InfoareaShort>"}`

   Example: `{"parentLink":"FI"}`

Header Swipe

You can define that the user can switch from one record to another by swiping to the left (previous record) or to the right (next record) in the details view header.

You can define, whether this option is enabled for all info areas or for some info areas only.

To enable the header swipe:

- Globally:

  Set the `View.RecordSwipeEnabledDefault` Web Configuration parameter to `true`, see View.RecordSwipeEnabledDefault.

- Only in some areas:

  - `RecordListView` action: Set the `SwipeDetailRecords` input argument (`true|false`), see RecordListView on page 117.
  - `Legacy Form`: Switch to the **Legacy Forms** area, add a form called `RecordList` and define the `SwipeDetailRecords` (`true|false`) in the **Options** column in JSON syntax:

    `{"SwipeDetailRecords":"true"}`

  - Related area panel, see CHILDREN_<S&L>.

Rep Lists

Learn how to define rep lists.

You can define how rep lists are displayed:

- Containing reps set to `Interal=true` (Field 103)
- Ignoring these reps

Two field groups exist to list the available ID records:

- `ID` (default): Includes the `Internal=true` setting.
- `IDSystem`: These records are not listed, alphabetical sorting is included.

Assign the desired field group in the Search&List configuration.

Filters

Learn where to add filters.

You can add filters to

- filter data in queries (client and server),
- define the offline data set (server only),
- set initial values for new records ("Template filters"), see Predefined Values for New Records,
- determine whether editing or adding records is allowed ("Rights filter"),
- filters that define which values are applied ("Decision filters"), see Decision Filters on page 203.
- editTrigger filters that update the dependent fields when other field values are updated,

While offline data set filters are only executed on the server all other filter types must also be executable on the client.

Sometimes, custom filters defined for some records can prevent the CRM.pad application from synchronizing these records with the CRM.pad server, when these records are linked from other infoareas.

For example, you may have set a filter to prevent the loading of a specific company record. And if the CRM.pad user adds this company to his favourites and then triggers a synchronization with the server. The Favourites infoarea is synchronized with the CRM.pad device, but the Company records are not loaded and the favourites points to an empty record.

Variable Expressions (Filter Tokens)

Filter tokens (variables) are automatically replaced with the current value when the filter is applied. The following filter tokens are replaced by:

- `$curRep`: The `repId` of the current rep
- `$curTenant`: The current rep's tenant (not additional tenants!)
- `$curOrgGroup`, `$curDeputy`, `$curSuperior`: The current rep's organizational group, deputy and superior
- `$curGeoLongitude` or `$curGeoX`: The current longitude coordinate
- `$curGeoLatitude` or `$curGeoY`: The current latitude coordinate
- `$curDay`: The current date

- `$curfdWeek`: The date of Monday of the current week
- `$curfdMonth`: The first day of the current month
- `$curfdQuarter`: The first day of the current quarter
- `$curfdYear`: The first day of the current year
- `$curTimeSec`: The current time in the format `HHMMSS`
- `$curHour:` The current hour
- `$cur$$$Y:` The current year
- `$????####`: The current day, to be used in birthday queries, see Birthday Filter.

You can add or subtract time periods like +1d, -5d, +2w, -3m, +1y for the following tokens: `$curDay`, `$curfdWeek`, `$curfdMonth`, `$curfdYear` and `$????####`.

You can use these expressions to hand over values to functions as variables with the `$par` prefix, for example:

- `$parDate:fdWeek`: the first week day of the date in `$parDate`
- `$parTime:Hour`: the hour of the time in `$parTime`

You can also add or subtract time periods, for example:

- `$parDate:Day+2d`: two days are added to the date in `$parDate`
- `$parDate:Hour+2h`: two hours are added to the hour of the date in `$parDate`

You can use the following expressions to hand over the current geographic coordinates to Template filters (e.g. when you create a new company record on the client's premises, the device's current location is applied):

- `$curGeoX`: the current longitude coordinate
- `$curGeoY`: the current latitude coordinate

Using Values from a Rep (ID) Record as Filter Tokens

You can also use values from ID records as filter tokens. For example, the `$parRepCountry` filter token is replaced by the rep's country.

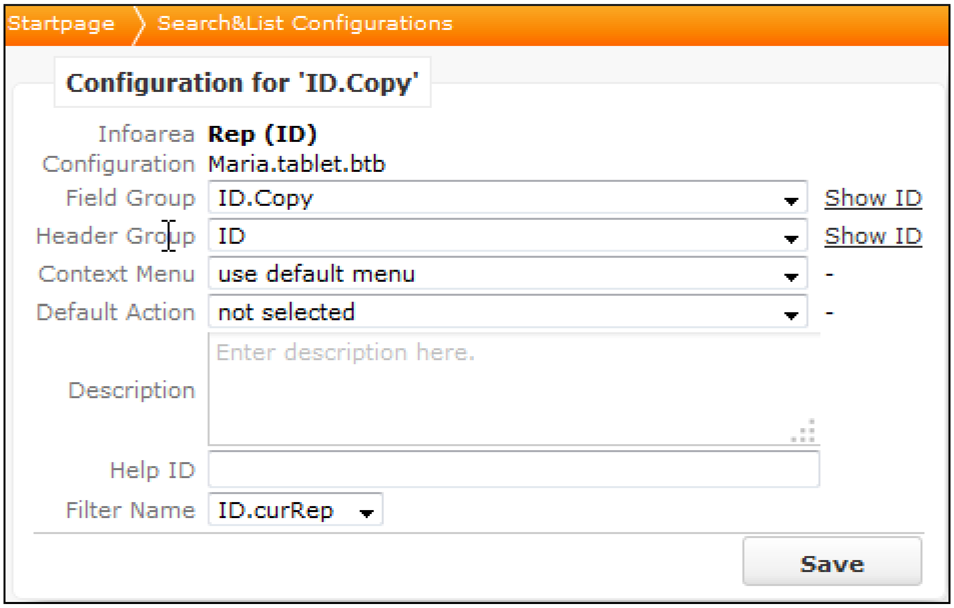
To configure which tokens can be used:

1. Create a field group (e.g. `ID.Copy)` and add the fields that you want to use as filter tokens to the List control.

   Enter the desired function names into the **Function** field of each field, for example:

Field	Function
Form of Address	RepGender
Country	RepCountry
Region	RepRegion

2. Create a Search&List configuration with the same name as the field group and select the **Field Group** (in the example **ID.Copy)** and the **FilterName** (in the example **ID.curRep**). If this filter does not exist, the Filter Token always references the values of the first rep found in the ID table:



3. Enter the name of the Search&List configuration in the `RepCopySearchandLists` web configuration parameter, see RepCopySearchAndLists:

You can apply these filter tokens in any query or filter by entering the following syntax: `$par<FunctionName>`.

Using Values from a Record as Filter Tokens

For `NewView/EditView` action templates and for the quick add you can also define values from a specific record as filter tokens.

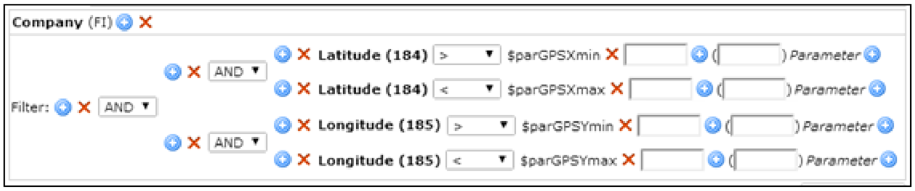
To define which tokens can be used:

1. Switch to the List control in the desired field group and add the fields that you want to use as filter tokens. You can also add records from other info areas that are linked (1:1 or n:1) to the current record.
2. Enter the desired token names in the **Function** field of the desired fields.
3. Set the `CopySourceFieldGroupName` input argument of the `NewView` and/or `EditView` action calls to the previously defined field group, see NewView on page 106 and EditView on page 93.
4. You can use these filter tokens to copy values from the current record when you open a new page, e. g. in Template filters.

Location Filters

A location filter can be used for searches to return only data that is located within a certain distance from a specific geographic point (Vicinity Search):

1. Define the following filter:



2. Set the `DistanceFilter.MaxValue` Web Configuration parameter to the desired value, see DistanceFilter.MaxValue.

Destination Filter

With destination filters you can filter destination records belonging to a root record, if these records are not directly linked.

Example from the OTC version:

If a rep issues samples to a person during a business call, the sample records are linked to the KP record, but no link exists to the MA record. Therefore, you can define a filter that you can use e. g. in the quick add to limit the number of records displayed.

Range Filter (From - To)

You can define a range filter with parameters, e. g. for dates.

The user can then define the range to limit the number of records displayed to records within the desired range. You can define range filters for any number fields, date fields, and even text fields.

Example: `MA.FromTo` filter contained in the `update.tablet` configuration.

Template Filter

A template filter defines default values that are automatically inserted when a user adds or edits a new record.

Example: `FI.ParamLand` filter contained in the `update.tablet` configuration.

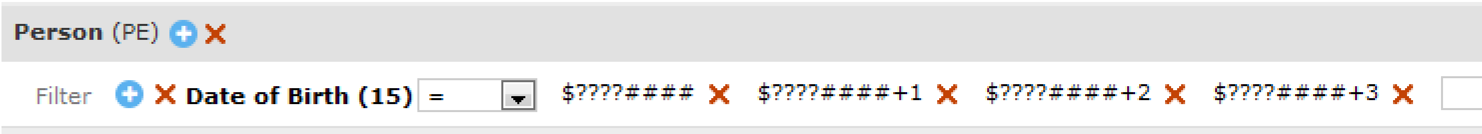
Special Parameters for Template Filters used in Record Selectors

If you define a template filter for a record selector, you can also add:

- Parameters using **Function** names (with the prefix `$par` or `$cur`) that have been assigned in the source field group.
- `Parameter :Key` function that is assigned to the new child record. If a user calls the record selector again before saving, CRM.pad does not add another new record but replaces the previous (unsaved) record.

Birthday Filter

If you want to define a filter that returns all birthdays within the next feww days, you cannot define a time period in one criterion but have to add every day as follows:



Delete Warning Filter

The **DeleteWarningFilter** allows user to use the existing filters for showing delete warning messages.

Based on any custom filter condition user can show custom message while deleting. This works side by side with delete `RightFilter`. If `RightFilter` and `DeleteWarning-Filter` used together. First `RightFilter` is evaluated then `DeleteWarningFilter`.

When using `DeleteWarningFilter` with `Delete.Ask` web configuration. Following are the behaviour:

DeleteWarningFil-ter	Delete.Ask	Warning Popup	Warning Message
Match	True	Shown	Message from DeleteWarningFilter
Not-Match	True	Shown	Generic warning popup message
Match	False	Shown	Message from DeleteWarningFilter
Not-Match	False	Not Shown	N/A

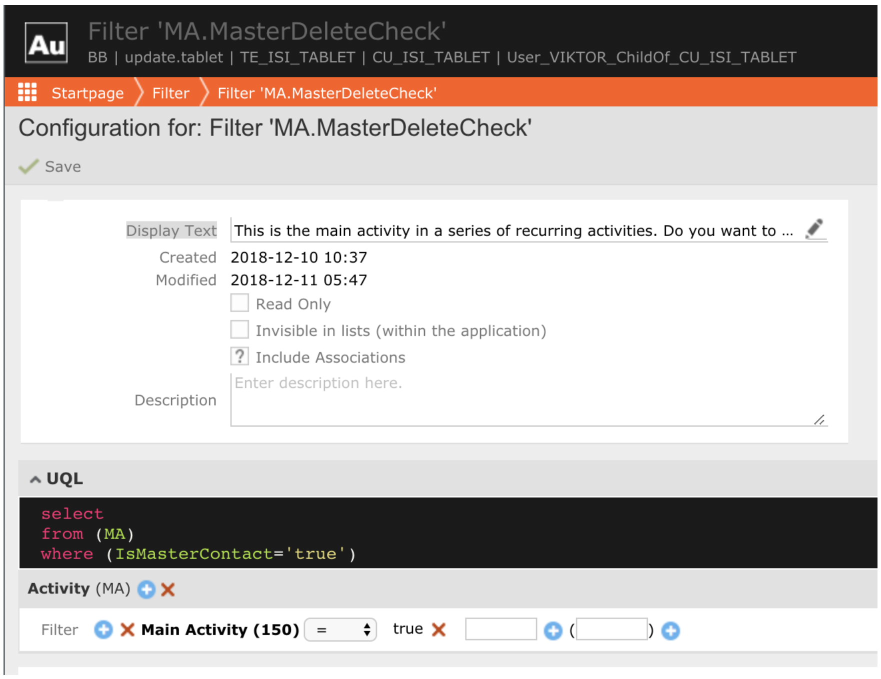
Following is a example usage of **DeleteWarningFilter** feature, to enable confirmation popup while deleting the main activity in a recurrence series.

Here **IsMasterContact** field in MA is used for the filter, need to do the following configurations

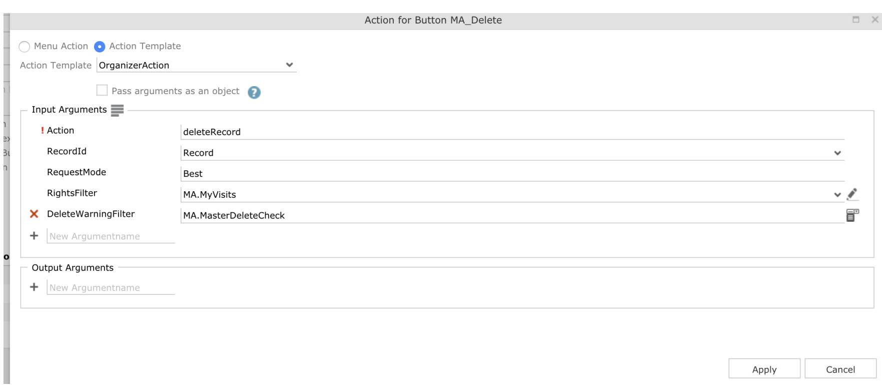
1. In the designer create a filter for MA as below:

   ```
   select from (MA) where (IsMasterContact='true')
   ```

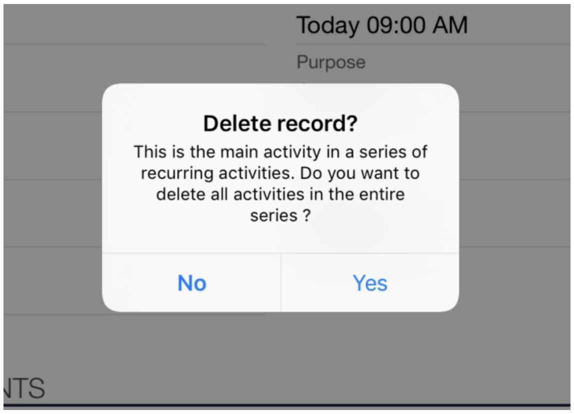
   Set the Display Text as the message "This is the main activity in a series of recurring activities. Do you want to delete all activities in the entire series ?" in the filter.



2. In the Delete Button need to add the Input Argument **DeleteWarningFilter** and set the newly added filter name. Use the modified delete button in the MA.Expand header



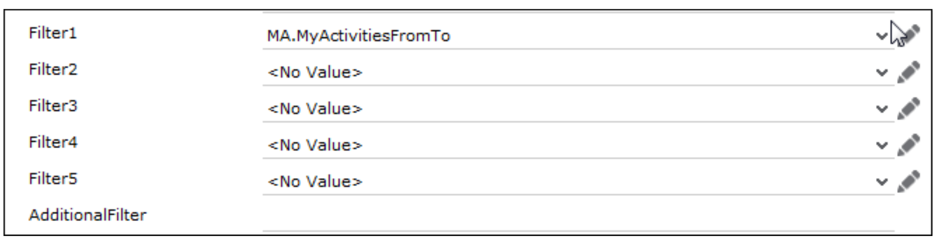
3. Make sure to do the Client Data Model update.
4. Do a full sync in CRM pad and try to delete a Main Activity. User gets the configured error message in a confirmation popup.



Adding Filters to the Action Call

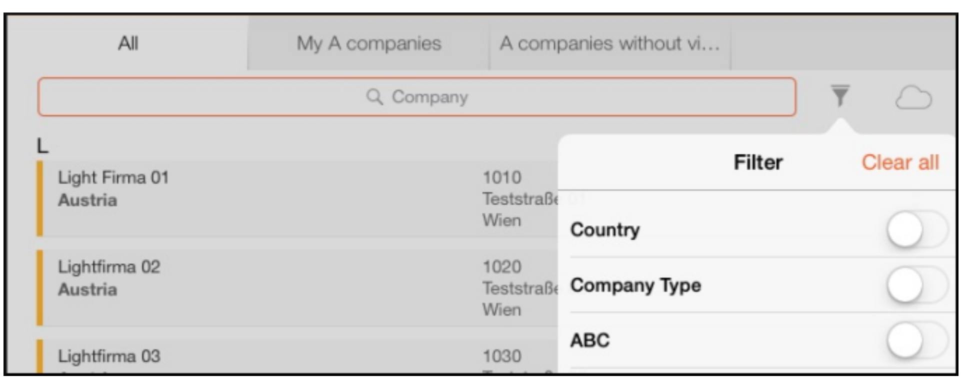
Learn how to add filters to action call.

Up to six (depending on the action template) filters can be assigned to an action call. If you need more filters, you can add them in the `AdditionalFilter` field (separated by semicolon).

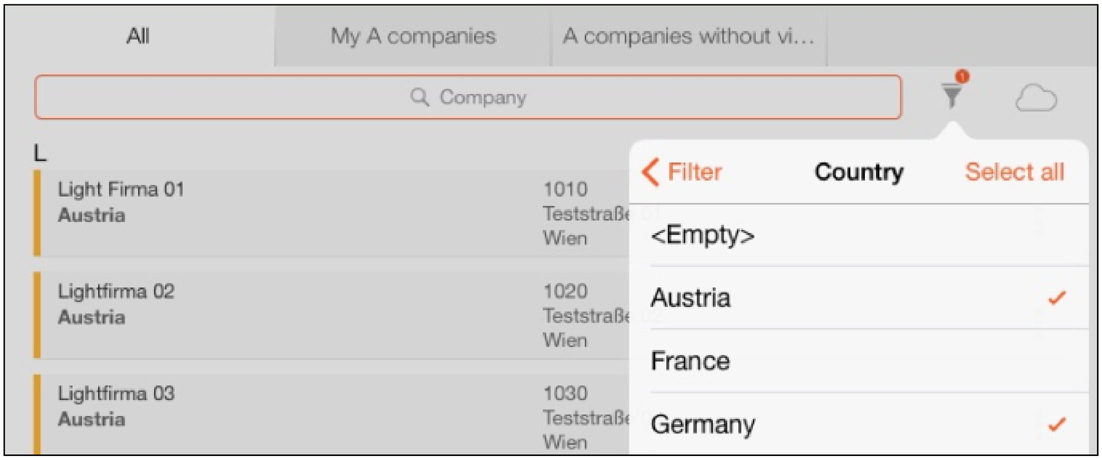


A (**Filter**) button is displayed in the application and the user can add and combine filters that are applied when searching the data.

If a filter has no parameter, it can be enabled or disabled (on/off switch as shown in the following screenshot).



If a filter has a parameter, the user must select one or more values from a list (check marks as shown in the following screenshot).



Note: If a selected filter is disabled, the selected values are still displayed but **not applied** anymore.

Creating Child Records Automatically

You can create child records automatically with `ChildCreateTemplate` filters.

These filters are automatically used if they exist in your configuration.

Syntax: `<InfoAreaID>.ChildCreateTemplate` (where `<InfoAreaID>` must be replaced with the info area's ID).

You can use this filters to create records in offline mode that would otherwise be created by triggers or the business logic on the server.

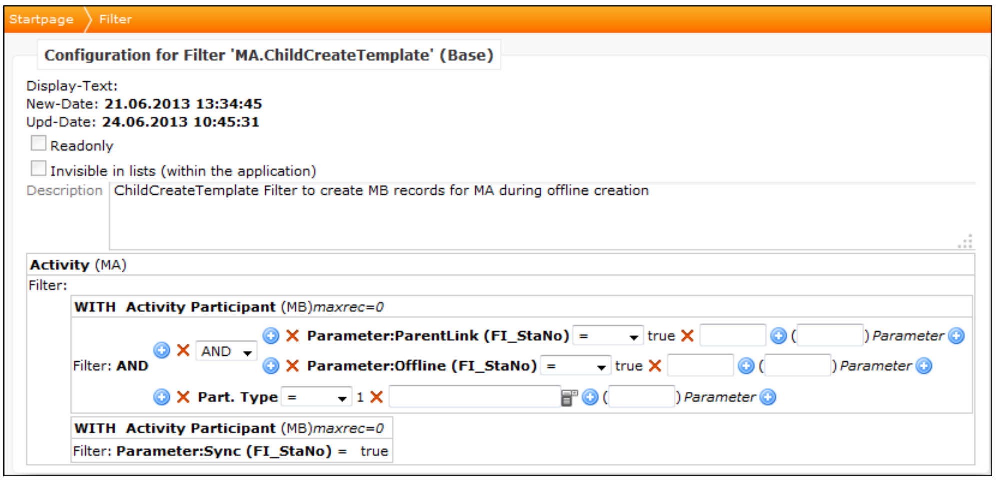
These filters can contain the following parameters:

- `Parameter :Offline :` The record is only created if the device is offline and is discarded if the device is online again. It is assumed that a server routine creates this record.
- `Parameter :Yield :` The record is only created if no other record exists for this info area.
- `Parameter :Sync :` Child records that were created on the server (e.g. by a trigger) for a record that was created on the device are also synchronized during the following incremental synchronization, see Synchronization Problems on page 380.

For example, if a MA record is created on the device, the server automatically creates MB records for it. Without this parameter, these MB records are not synchronized unless the user carries out a full synchronization. If you set this parameter, these records are included in every synchronization request. However, this may cause performance issues.

Note: You cannot combine `Parameter Sync` with the other options but must enter it in a separate condition on its own due to technical reasons (see example below).

- `Parameter :ParentLink :` Defines the link to the parent record. Possible values:
  - `true`: Uses all links
  - Info area ID or `Infoarea#LinkId`: To set the desired links manually.



Conditional New/Edit Rights Using Rights Filters

Learn about the use of Rights filter.

The `RightsFilterName` parameter is available for the `NewView` and the `EditView` actions. If you enter a filter name, it is applied to the current record (e.g. in edit mode, the record to be edited, and in New mode, the linked parent record, i.e. the record from where the New mode was started). Only if the record fulfills the criteria, the operation is carried out.

Example:

A user can create a person for a linked company, but only if the company meets the defined filter criterion (e. g. must be based in Austria). If the filter criterion is not met, an error message is displayed ("Action is not allowed. Data content prohibits this action.").

You can define your own message text by entering a **Display-Text** in the filter. This text is then displayed instead of the standard text:

Note: Depending on where the checked record is stored (server or client), the filter is either applied on the client or remotely. If the filter does not only check conditions on the current record but also e.g. parent or linked records, ensure that all records needed to check the rights filter are also synchronized to the client.

Some linked records may not be synchronized if there are custom filters defined on the record.

Filtering Items from the iPad Calendar

Learn how to filter items from ipad calendar.

If you define a date filter, e. g. to display the list of the current day's appointment on the dashboard, items from the iPad calendar are only filtered, if you enter the `Date` function name into the **Func** field.

Decision Filters

Learn about the use of decision filters.

Decision filters can be used for the following purposes:

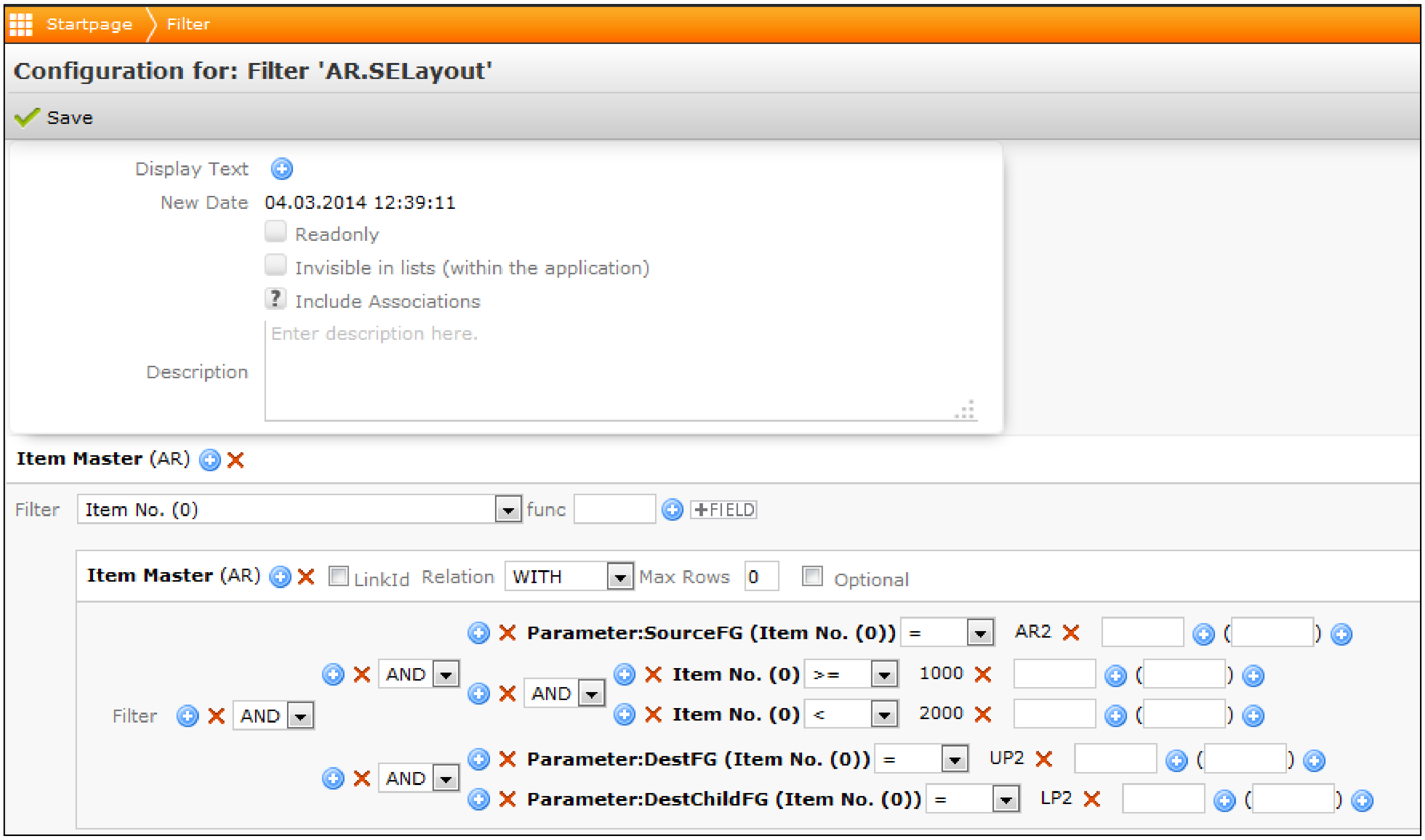
- To use different field groups in the quick add based on the source data of a single article.
- To use different MiniDetails actions or default actions based on the record's data in a `RecordListView`.
- To define quick actions for objectives.

A Decision filter has several sub-info area nodes of the same info area, each with conditions and properties.

Example: Decision Filter for the Quick Add

You can specify a Decision filter in the new `FieldGroupDecider` input argument (`Serialentry` action call).

In the following example, items with an Item No. between 1000 and 1999 does not apply the field group defined in the `SerialEntry` action call but the ones defined in the following Decision filter (i. e. AR2, UP2, and LP2).



The `SourceFG` (AR2), `DestFG` (UP2) and `DestChildFG` (LP2) parameters are applied to all items within the Item No. range from 1000 to1999. These parameters override the "default" field group assigned in the `SerialEntry` action call.

You can also define multiple sub-info area nodes for multiple conditions.

You can define the following parameters:

- `Parameter:SourceFG`: Overrides `SourceConfigName` (List-Control)
- `Parameter:DestFG`: Overrides `DestinationConfigName` (Edit-Control)
- `Parameter:SourceChildFG`: Overrides `SouceChildConfigName` (List-Control)
- `Parameter:DestChildFG`: Overrides `DestinationChildConfigName` (Edit-Control)

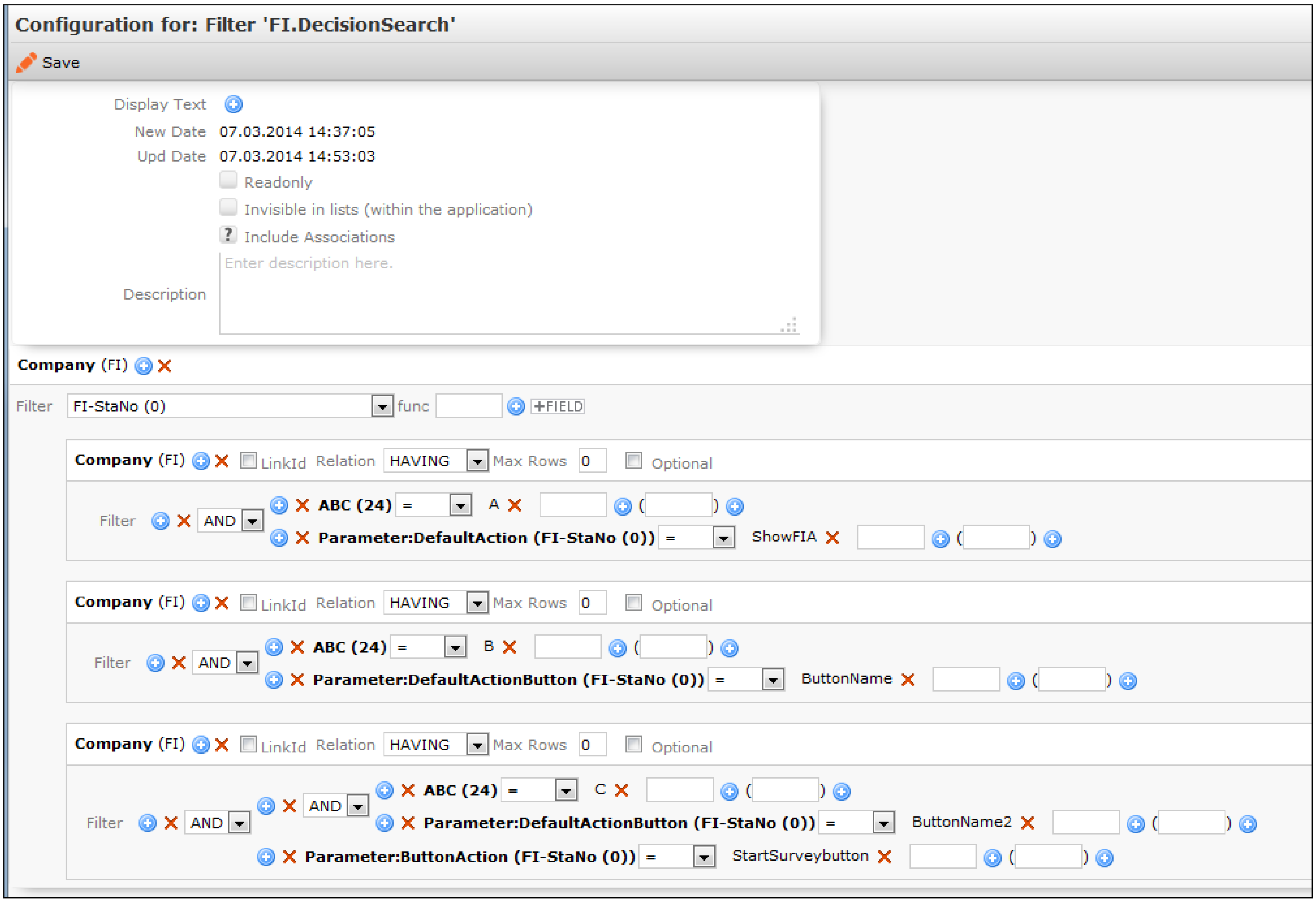
Decision filters only affect the display of data, the quick add business logic remains unchanged and is still defined in the field groups and in the action calls:

- All fields must exist in both, the field groups referenced by the filter and the ones assigned in the action. Fields that do not exist in the field group assigned in the action call are ignored.
- If fields are only needed for the filter, you can set the **Hide Field** field attribute in the field group.
- **Function** names and the **Don't save** and **Don't save offline** field attributes **cannot** be overridden by a Decision filter.
- You can define different (alternative) labels.

Example: Decision Filters for RecordListview

Decision filters can be specified in the `DetailActionSwitchFilterName` input argument (`RecordListView` action call).

In this example, the record's default actions are based on the ABC field's value. The defined default actions used are applied to the corresponding records. Additionally, for "C" companies an additional quick action is available in the MiniDetails.



You can define the following parameters:

- `Parameter :DefaultAction`: Defines the alternative default action (context menu action) that is executed instead of the standard default action.
- `Parameter :DefaultActionButton`: Defines the alternative button instead of the default button. The action assigned to this button is executed.
- `Parameter :Action`: Defines one or more context menu action that are displayed as additional quick actions for the MiniDetails.
- `Parameter :ButtonAction`: Defines the buttons that are displayed additionally in the MiniDetails.

Decision Filters in Quick Actions for Objectives

If you use Decision filters in quick actions for objectives, define them as described in Example: Decision Filters for RecordListview.

However, **only** the following **two parameters** are available:

- `Parameter:Action`
- `Parameter:ButtonAction`

Assign the Decision filter to the `ExecuteActionFilterName` input argument of the `ObjectivesEditView` action call, see ObjectivesEditView on page 107.

Allowing Adding/Editing Records Only in Online Mode

You can define that a user is only allowed to add or edit records when in online mode.

You can:

- Set the `RequestMode` input argument for the desired `NewView` or `EditView` page call, see General Input Arguments on page 70.
- Set the `Edit.DefaultRequestMode` Web Configuration parameter to `online` to enable this option throughout the application, see Web Configuration Parameters and Layouts on page 23.

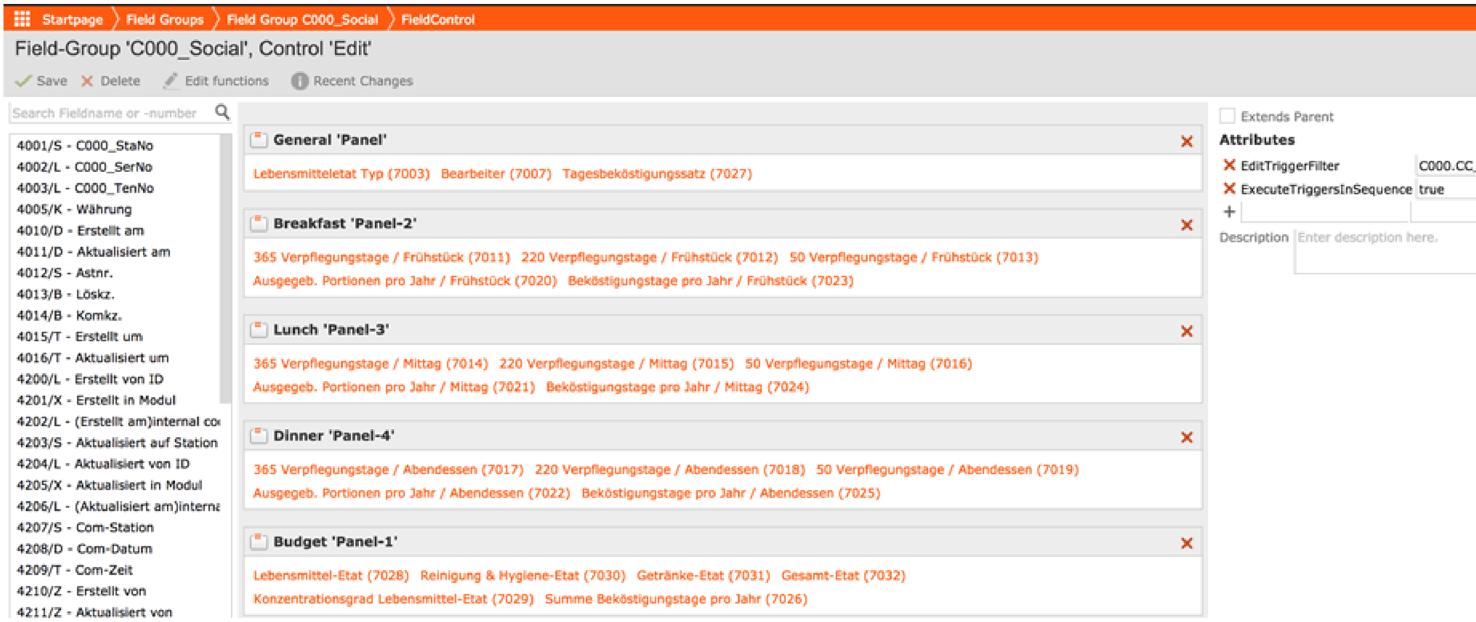
Sequential Execution of Edit Trigger Filters

The EditTriggerFilter attribute allows you to configure multiple triggers in sequence separated by a semicolon.

In certain cases a field value might be dependent upon another field that was updated by a previous editTrigger in the sequence. In such cases the dependent field value is not evaluated. For example, if a field A = B + C and field D = A + E. Then setting an edit trigger in sequence for the two updates only updates field A but not field D that is dependent on field A.

To ensure that field D is updated with newly updated value of field A, set the ExecuteTriggersInSequence attribute to true.

The following screenshot shows the two attributes set for a field group.



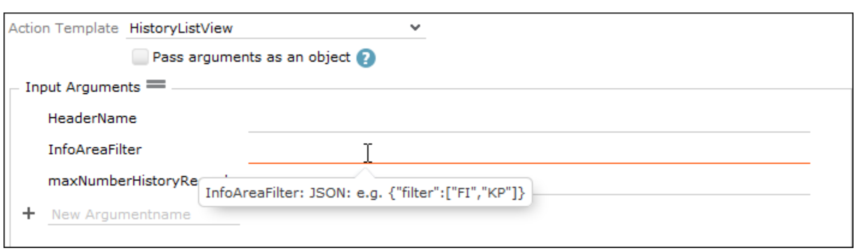
Adding a History View

Learn how to add a history view.

To add a history:

1. Switch to the **Menus & Actions** area and add a new **Application Menu Action**, e.g. `History`.
2. Switch to the **Headers** area, add a new header (e.g. `History`) and assign `all Infoareas` as **Infoarea**.
3. Open this header, add a new **Special Header** (e.g. `History`) and add a **Text** to this special header.

   This text is displayed in CRM.pad as header for the history list.
4. Assign the `HistoryListView` action call to this menu action and define the following input arguments, see HistoryListView on page 100:



- `HeaderName`
- `infoAreaFilter`
- `maxNumberHistoryRecords`

Array-Fields

Some catalogs are defined in the database as array fields.

For example, in CRM.pad's LSI template, the fields **Qualification**, **Qualification2** and **Qualification3** read their data from the same catalog.

In this case you can set the `colspan` attribute for the first field (e. g. **Qualification**). This results in only one field being displayed in CRM.pad. The user can select multiple entries, max. the number of fields that have been combined with the colspan attribute.

Downloading Resources

You can add the resource files of a configuration to a zip file and then add this zip file to the **Images** area in CRM.designer.

This zip file can be loaded instead of single resource files. You can decide for each existing configuration whether the resources are available as individual files or in a zip file.

Each zip file added to the **Images** area must be named with `ZIP`**:** as prefix and the according rights must not only be set for the directory where the zip file is stored but also the file itself.

Defining E-Mail Fields

You can define that fields from info areas (e.g form an MA record) is automatically passed to your e-mail client.

You need to assign functions to the required fields in the field group's List control. The value that is passed to the e-mail client is defined by the field's label. These labels can contain placeholders (Syntax: `{$FunctionName}`) that refer to functions assigned to other fields in the control.

Example 1 (MA info area):

Field	Function	Label
FreeC1 (56)	`ToText`	`{$ToAddress}`
E-mail (9) (KP)	`ToAddress`	

This results in the e-mail address stored in the (linked) E-mail field in the KP info area being transferred to the **To :** field of the e-mail client.

Note: Enterprise version of Aurea.CRM allows CRM.pad users to add multiple email addresses from activity participants (MB) info area by adding the linked E-mail field from the Persons (PE) Info area. For more information on sending emails to multiple activity participants, see Sending Email to Multiple Activity Participants.

Example 2 (MA info area):

You can also use multiple placeholders in a label. You could also add a fixed string but no multiple language support is available for these fixed strings.

Field	Function	Label
FreeC2 (57)	`ToText`	`Message: {$Text} Best wishes, {$Rep}`
Text (23)	`Text`	
Rep (115)	`Rep`	

If the Text field contains the text "Thank you for your message." and the Rep field the name "John Doe", the defined label results in:

Message: Thank you for your message. Best wishes, John Doe.

The following **Function** names are available to pass data to the e-mail client:

Function	Field in the e-mail client	Description
`TOTEXT`	**To:**	**Allows you to add participant email ID.**
`TOTEXT2`	**To: Company Email ID**	**Allows you to add the Activity company email ID.**
`SUBJECTTEXT`	**Subject:**	**Adds the default subject matter provided in this field to the subject of the email.**
`CCTEXT`	**Cc:**	**Allows you to add participant email ID to the CC field.**

Function	Field in the e-mail client	Description
BCCTEXT	**Bcc:**	**Allows you to add participant email ID to the Bcc field.**
BODYTEXT	**The e-mail's body**	**Adds the default text matter provided in this field to the body of the email.**

Control Center

The control center contains functions that the user can call throughout the application such as the synchronization options or the system information.

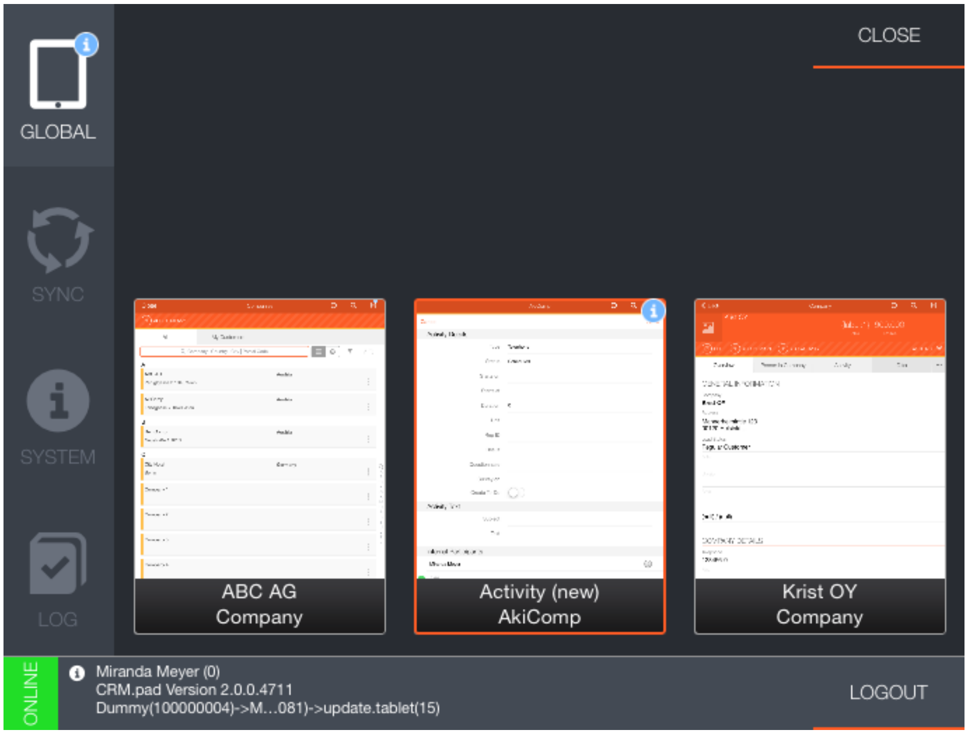
You can define global actions that are available in the **Global** area of the control center, see Global Actions.

In the control center the user can:

- Switch between the active organizers by tapping on the desired one, see Multiple Organizers on page 211.
- See, which mode (online or offline) is currently active.
- Log out from the application
- Switch to the other options (**Global**, **Sync**, **System**, **Log**)
- Close the control center

Multiple Organizers

Multiple organizers are enabled by default. Open organizers are displayed in the **Global** area of the Control Center.



Due to technical restrictions only one New or Edit view (e. g. edit, quick add, or survey edit page) can be opened at the same time.

CRM.pad manages automatically when organizers are created or dropped. For example, if the user edits a company and then navigates to a person (using the global search), the person record is automatically opened in a second organizer. This organizer is automatically dropped, when the user taps the **Close** button.

The following rules are applied for dropping organizers:

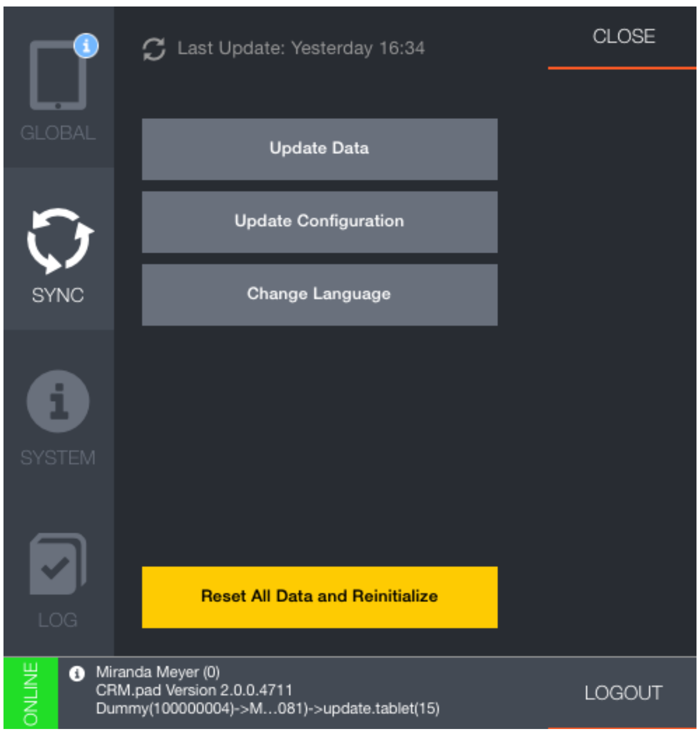
- An organizer containing a record in Edit mode cannot be dropped.
- An organizer is dropped if it was started e. g. from the search and the user navigates back to the initial page.
- If an organizer is started by the system, and two organizers that are not in Edit mode already exist, the one that was not used the longest is dropped.

Synchronization

The user can synchronize data in the **Sync** area of the control center.

The user can synchronize either:

- All data (full synchronization), i. e. the data and the configuration, see Full Synchronization.
- Only the data that has been changed since the last synchronization (incremental synchronization), see Incremental Synchronization.
- Only the configuration (all the data from the designer database including resources), see Configuration Only.
- Change the application's language.



Note: Data is not synchronized when the device switches to sleep mode, i. e. synchronization processes are interrupted (even if a full synchronization is being carried out). Therefore, Aurea software recommends to adjust the delay until the device switches to sleep mode accordingly.

You can use the following Web Configuration parameters to disable options:

- Sync.ConfigOnOff.
- Sync.LanguageOnOff.

Full Synchronization

The full synchronization transfers the data model, catalog values, the configuration, record data, configuration resource files, and documents as defined beforehand.

Note: Records deleted on the server are not transferred to the device (i.e. are not deleted on the device) unless a full synchronization is carried out.

Data is transferred in several requests that are sequentially processed:

- At first, the data model, catalog and configuration data is transferred from the server to the client.
- A request for each offline data set fetches the record data.
- All configured resource files are transferred.
- All document requests are sent to the Document Download Manager who downloads them asynchronously.

The application is blocked during synchronization, other actions cannot be carried out.

Data is fully synchronized when a user accesses the application for the first time. Afterwards, the user is reminded to fully synchronize data regularly. After a certain time period, users cannot access the application anymore without carrying out a full synchronization. You can define the interval, when a user must carry out a full synchronization with the following web configuration parameters:

- Sync.FullSyncWarnIntervalDays.
- Sync.FullSyncIntervalDays.
- Sync.FullSyncBlockIntervalDays.

If the application is terminated during the synchronization, the state is stored on the device. If the user logs in again, the synchronization continues.

Note: Records that are not included in the offline dataset are removed from the device when the next full synchronization is carried out, see Offline Record Set / Offline Data Set on page 51.

Incremental Synchronization

An incremental synchronization transfers all record and catalog data changed since the last synchronization (full or incremental). It is executed in one request, usually only takes seconds and can be carried out frequently.

You can define that data is incrementally synchronized at specific time intervals, see Sync.Config.

After synchronizing data is up-to-date, **except**:

- Data downloaded with the application but not belonging to an existing record set.
- Data that does not belong to an offline data set anymore because the rights definition or the record set itself was changed.
- Data belonging to an offline data set after the rights definition was changed but the data itself was not changed. This data is only included in a full synchronization.

Note: Records that have been marked as deleted (DelCd. field) in the Aurea CRM database are also synchronized. However, physically deleted records are not considered. They are only removed from the device when carrying out a full synchronization.

Configuration Only

It is recommended to use this option only during the project development phase. Only the application's configuration data is transferred.

UpSync

The UpSync is started automatically when the application switches from offline to online mode and synchronizes only requests that were created while working offline.

System Information

System information is available in the control center's **System** area.

The following configuration units are included in the `update.tablet` configuration:

- Context menu action called `INFO_SYSTEMINFO`. This context menu action calls the `SystemInfoPage` action, see SystemInfoPage on page 138.
- Company (FI) info area:
  - Field group called `SYSTEMINFO`
  - Header group called `SYSTEMINFO`

  Note: This header group does not apply to version 2.x but exists for compatibility purposes for earlier versions.

The **System** area gives an overview of information on the current set-up (such as the version number, server URL, assigned rights, logged-in user, configuration, etc.) and facilitates user support.

You can add additional information as follows:

1. Create a **Web Configuration Layout** named `SystemInfo`. Every parameter in this layout is automatically appended at the end with its name. See the see Aurea CRM web Administrator Guide on how to create a Web Configuration Layout.

   Note: You cannot change modify but only display these additional parameters.

2. If you want a different label displayed for a parameter, add an additional (dummy) field to the `SYSTEMINFO` field group, enter the name of the desired Web Configuration Layout parameter in the **Function** field and define the **Label**.

Default SYSTEMINFO Field Group

The data displayed in the System Information is defined in the List control of the FI info area's `SYSTEMINFO` field group. The default is configured as follows:

These fields are used as placeholders to assign **Function** names to them. Ensure that the following data is given in the **Label** and **Function** fields for each field:

Field	Label	Function
FreeC1	Device Model	`devicemodel`
FreeC2	Operation System	`system`
FreeC3	App Version	`appversion`
FreeC4	Aurea CRM web Version	`webversion`
FreeC5	Server URL	`serverurl`
FreeC6	User	`user`
FreeC7	Roles	`roles`
FreeC8	Rights	`rights`
FreeC9	Configuration	`configuration`

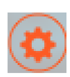
Note: If you are using a RAS application and you also want to display the RAS application ID in the **System** area of the control center, add another field and assign the `rasapplicationid` as **Function**.

Log Files

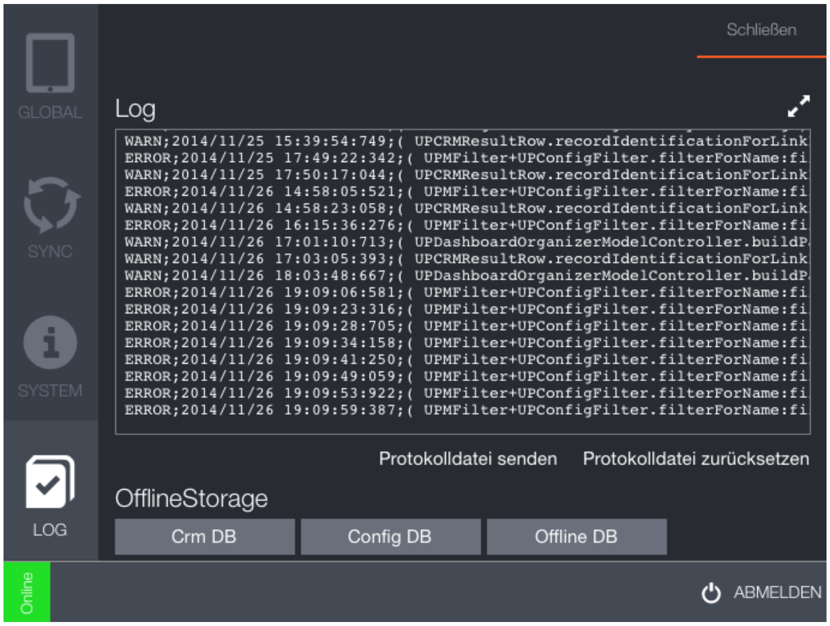
Logging information is displayed in the **Log** area and can be sent by e-mail to your administrator.

Logging is enabled by default after installing CRM.pad, the log level is automatically set to `Warning`.

Log files can be viewed as follows:

- In the **Log** area of the control center. You can send the log file by e-mail, reset the log file or query the available databases (SQL statements).
- If login was not successful, by tapping on  (**Settings**) and selecting **View Logs.**

The content of the log file is displayed:



The user can reset the content or send it e-mail.

You can define with the following Web Configuration parameters whether

- after an error occurs, a button is displayed in the pages header with which the user can send the e-mail, see SendErrorReport.
- additional information such as screenshots etc. is included in the sent data, see SendErrorContextInfo.

Overruling Default Log Settings in the iPad Settings

In addition to the **Log** area in the CRM.pad's control center, more logging options are available in iPad's **Settings** area. These options are available for each user and can help to detect problems. For example, the Support Center can ask the user experiencing problems to change the log level (e. g. to track a specific error or when no access to the CRM.designer database is available) and then to e-mail the log file to the Support Center agent.

Only ask to set these options for testing purposes or to locate the cause of errors, as each enabled item increases the data transfer and, therefore, can lead to performance problems.

The user must switch to the iPad settings () and tap on **CRM.pad** icon to enable these additional settings:

- **Application Information:** Shows the version number.
- **Logging**: The user can define the log level (default level: `Error`) and which log files are written, e.g.:
  - **Override Server Log Level**: If set to `ON`, only options set to `ON` of the available log options are logged.
  - **Log Level**: Defines which occurrences are written to the log file.
  - **SQL Log**: SQL statements of the local database are logged.
  - **Request Log**: Server requests (sent and received) are logged
  - **Config Log:** Configuration items (loaded or used) are logged.
  - **Network Log**: General network communication is logged.
  - **Result Log**: Each result row is logged.
  - **Questionnaire Log**: Questionnaire information is logged.
  - **Quick Add Log**: Quick add information is logged.
  - **UpSync Log:** UpSync requests are logged.
- **Development Settings**:
  - **Full Data Sync after Login**: If enabled, data is fully synchronized when the user logs in. This may be useful e. g. when problems occur while CRM.pad tries to display the start page unsuccessfully. This option is automatically switched off after logging in successfully.
  - **Debug Console**: If enabled, the user can tap on **Debug Console** to start an input control. The user can enter SQL statements and execute them.
  - **Advanced**: The user can enter an alternative configuration, e. g. to display data in the previous "outfit" of version 1.x or to try out configurations that are not yet available.
  - Example:

`{"oldRasHandling":"true"}` can be used as fallback when when problems occur after an automatic migration. This options defines that the "old" logic is applied to local databases.

- **Layout Settings**:
  - **Show Quick Add Overview on the left**: If enabled, the Quick Add Overview is displayed in landscape mode on the left side, if disabled it is displayed on the right.

URLs and URIs

Learn how to configure URLs and URIs for CRM.pad.

URL Hand-over

You can define that an URL is called and handed-over including the current record's parameter with the `OpenURL` action template, see OpenUrl on page 109.

Calling records Using URIs

You define that an URI is generated and, for example, if it is sent per e-mail, the recipient can click on the URI. The application is automatically started/opened and the defined default action executed. In most cases, a record is displayed in Expand view.

The URI can be given in the following formats:

- `crmpad://executeAction/InfoArea={INFOAREAID}&RecordId={RECORDID}`
- `crmpad://executeAction?InfoArea={INFOAREAID}&RecordId={RECORDID}`
- `crmpad://executeAction?Identification={RECORDIDENTIFICATION}`

The `RecordIds`can also be given in decimal or hex code, e.g.:

- RecordIddecimal:`crmpad://executeAction/InfoArea=FI&RecordId=429496729665`
- RecordIdhex:`crmpad://executeAction/InfoArea=FI&RecordId=x0000006400000041`
- RecordIdhexlong:`crmpad://executeAction/InfoArea=FI&RecordId=429496729665`
- RecordId Hex leading 0 `crmpad://executeAction/InfoArea=FI&RecordId=0x0000006400000041`
- Identification by hex `crmpad://executeAction/Identification=FI.x0000006400000041`

Note: When providing a `RecordId` in hex code, the characters must be in lowercase.

Starting CRM.pad from Another Application

You can, for example, fill the shopping cart in an external application and then access the order's quick add directly:

1. You need to define a button with the syntax `external: <ActionName>` calling the desired action.

   Note: A button only named `<ActionName>` is `not` executed. The `external` prefix is required for the button, but must not be given in the URL.

2. You need to hand-over an URL in the following syntax:

   `crmpad://action?name=<ActionName>&recid=<RecordId>[&FurtherParameters]`

   Replace `<ActionName>` with the desired action's name.

   Example:

   `crmpad://action?name=Order&recid=AU.x000027127f8dcf17`

3. If no `<ActionName>` is given, the default action for the info area is executed. All parameters that are supported by this default action can also be given using the syntax `<ParameterName>=<Value>`

   Examples:

   `crmpad://action?recId=FI.x00000064000004ab` calls the FI info area's default action.

   `crmpad://action?recId=FI.x00000064000004ab&RequestOption=Online` calls the FI info area's default action and the `RequestOption` set to `Online`.

Images

Learn how to add images to CRM,pad.

You can add your own images in PNG format to CRM.pad, for example, a logo, calendar pictures, etc.

To add images:

1. Copy the image file to the `.. /web/tabletresources` directory of your Aurea CRM web installation.
2. Register the image file in the **Images** area of CRM.designer, for example:

3. Carry out a full synchronization. The images can only be displayed after being synchronized to the device.

Retina Displays

Starting with iPad 4, the display supports high pixel density (Retina display). Therefore, add your images twice:

- "normal" resolution (`*.png`) for normal resolution devices
- high resolution (`*@2x.png`) for retina devices

For example, if you have an image called `logo.png` that is 320x40px, then the file `logo@2x.png` should be 640x80px.

Legacy Forms

Legacy Forms are forms that were originally defined in update.seven designer and were migrated to CRM.designer.

The `update_tablet` configuration contains the following Legacy Forms:

- StartPage on page 224.
- StartPage.LegacyMenuTemplate on page 226.

Row Types

Learn about row types in legacy forms.

You can assign the following types to a row in a Legacy Form.

DatePicker Type

Displays the calendar on the start page.

- **Func** argument:

  You can set a specific date or use the `$curDay` variable (default) to define the day displayed.
- **Valuename** argument:

You can define a name that you can use as a parameter in the **Func** or any other argument within the same Legacy Form. See the example in see RecordList Type.

- **Options** argument:

  For example, you can define the date picker's behavior, see Configuring the Date Picker's Behavior on the Start Page.

RecordList Type

Displays records like child records in the Details view.

You can add more than one `RecordList` row to a Legacy Form. The record lists are searched for content in the order they are added. If no data can be found for the first `RecordList`, data is searched for the second one, and so on.

- **Func** argument:

  You can define the following options:

  - The name of the Search&Listconfiguration from where the field group name is taken.
  - A link record: You can either leave it empty or enter `ID .$ curRep` if you want to display records linked to the current rep.
  - A filter name
  - `Param1` (optional)

  Example:

  The default configuration contains a `DatePicker` row that has `Calendar` as **Valuename** assigned. You can therefore reference the date picker with `$Calendar`.

  The default configuration contains `MA; ;MA.Date ;$Calendar` in the record list's **Func** field resulting in the date picker's date (defined in its **Func** argument) being handed over as `$parParam1` to the given `MA.Date` filter. The record list is then only displayed for the current day.

  Further available parameters: `FI ;$ MARecord` or `KP;$MARecord`.

  You can add a `RecordList` row and assign `FI ;$ MARecord` or `KP;$MARecord` in the **Func** field. If the user taps on a record in the MA list, the corresponding FI or KP record is displayed in a list.

- **Options** argument:

  You can define the following options in JSON syntax:

  - `MaxResults`: Limits the number of records shown.
  - `RequestOption`: Defines how the data is searched for, e. g. `{"RequestOption": "Best"}`
  - `NoDetails`: If set to `true`, the user cannot switch to the Details view.
  - `DisablePaging`: If set to `true`, paging is not available if records in a list are formatted in row or classic layout, see ListStyle Input Argument.

StaticText Type

You can only add a `StaticText` row after a `RecordList`. The text entered in the **Valuename** column is displayed in the application if no data is found.

Map Type

The **Func** arguments for the `RecordList` type also apply to the `Map` type, see RecordList Type. But the data is displayed in a map.

Note: If the Search&List configuration given in the Func column has also a Default Action assigned, this action is executed when the user makes a long-tab on a pin in the map.

The configured List control must return four fields:

- GPS y-coordinate
- GPS x-coordinate
- Title (shown in the map)
- Address

You can set the ColSpan field detail to combine the values of more fields, see Combining Field Values on page 163. If a row does not have GPS coordinates, its location is determined by the address field.

HTML Type

You can define that, for example, a client's web page is displayed. Web pages can only be displayed in online mode.

- Enter the URL in the Func column.
  - You can add the current rep's name or ID to the URL with the $curRep or $curRepId variables, for example, `http://www.google.at/search?q=$curRep;`
- Adjust the height by entering JSON into the **TD-Attributes** column.

ActionForm and TriggerExecution Types

The parameters and functions of these two forms are described in the topic, see Executing Server Workflows with Parameters on page 367.

InsightBoard Type

Displays an Insight Board.

Example: By default, the SMARTBOOK application menu is displayed on the start page in the out-of-the-box version, see StartPage.LegacyMenuTemplate on page 226.

Analysis and Query Types

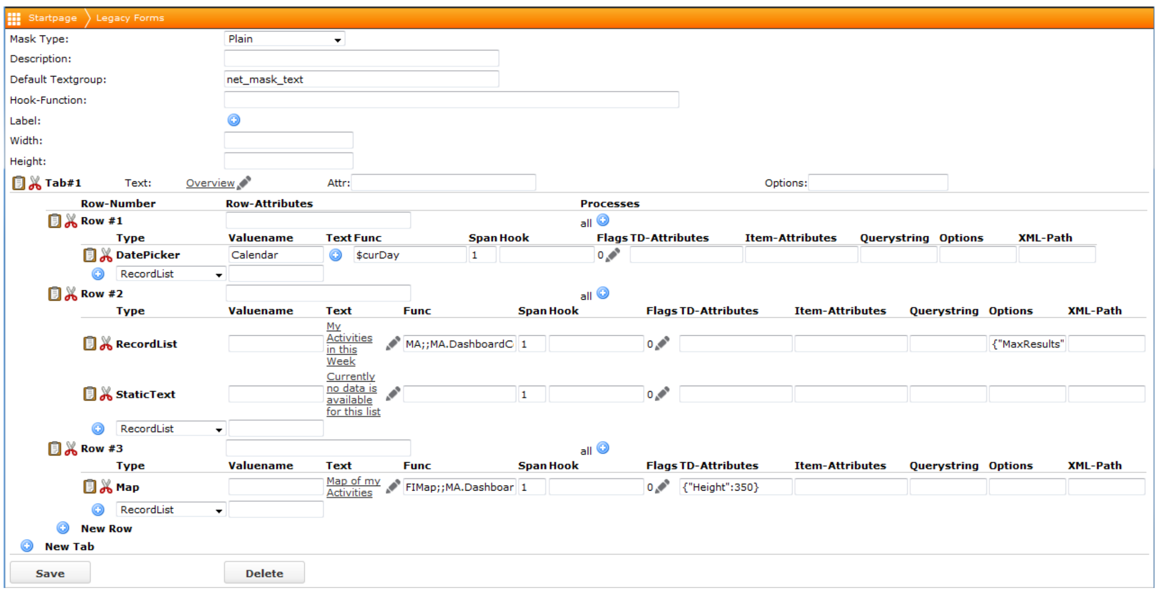
You can add an analysis (as of CRM.pad 2.1) and a query (as of Version 2.1.1) to a Legacy Form.

Enter the analysis or query name in the **Func** argument. You can also add a link record (e.g. the name of a `RecordList` row).

StartPage

This Legacy Form defines the content of the start page displayed after the user logs on to the CRM.pad app.

The following items are included in the default `StartPage`:



- Date Picker, see DatePicker Type.
- Activity list with a filter applied that displays the current reps activities, see RecordList Type.
- Map limited in size, see Map Type.

You can add further rows. For each row you must define

- The **Type**, see Row Types on page 221.
- The **Func** content (a list of values separated by semicolon) that depend on the assigned type.

You can enter additional options in JSON syntax in the following fields:

- **Options** column: Additional options
- **Item-Attributes** column: Item-specific options

Example: A color bar is displayed below a page's header. By default, the info area's color is assigned.

For the start page, the Company info area's color is displayed. You can define a different color for the start page in the **Item-Atrritubes** field of the Legacy Form's item (**Row #1**), e. g. `{"Color":"#FF0000"}`.

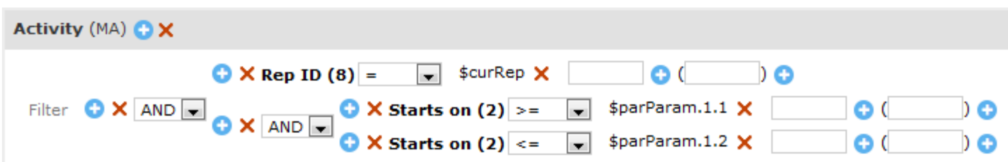
Configuring the Date Picker's Behavior on the Start Page

By default, the user's activities for the current week are displayed after the user switches to the start page. And if the user taps on a specific date, the activities for the week of the selected date are displayed.

You can change the default behavior, so that the user can list either the selected day's or the selected week's activities.

To configure the date picker's behavior:

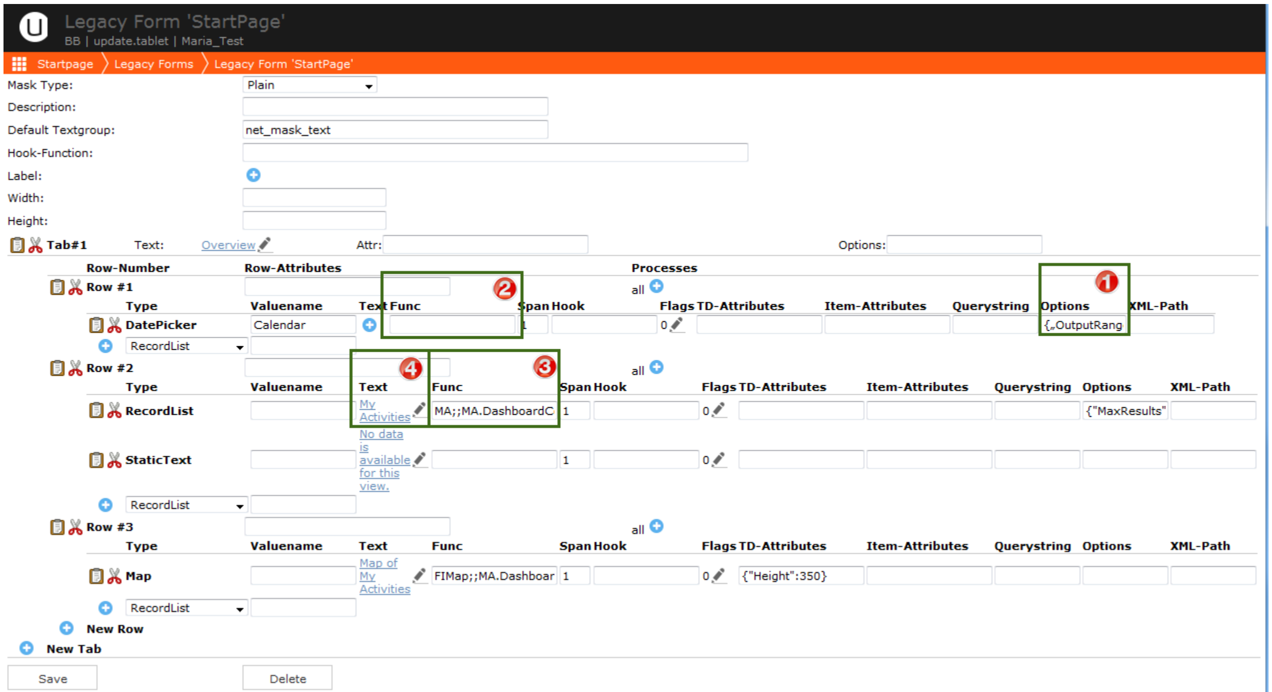
1. Define a filter as follows:



Note: Instead of defining a new filter, you could also adjust the existing `MA.DashboardCalendar` filter accordingly.

---

2. Switch to the **StartPage** Legacy Form:



3. Specify the following:

   a. Enter `{"OutputRange" :true }` in the **Options** argument field of the `DatePicker` row.

   b. In the `update.tablet` configuration, `$curDay` is entered in the **Func** field by default. Delete this value.

   c. In the **Func** field of the `RecordList` and the `Map` rows, the filter applied is referenced (by default `MA.DashboardCalendar`). If you defined a new filter, replace this value with the name of your filter.

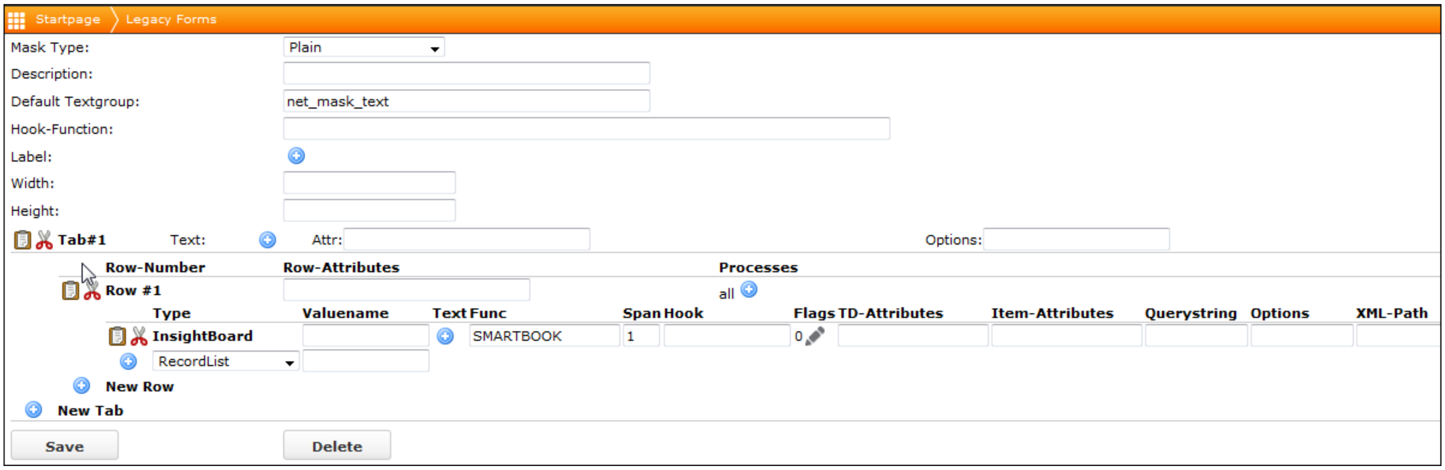
   Note: The filter referenced must contain the following condition `Date>=$parParam.1.1 AND Date<=$parParam.1.2 AND Rep ID=$curRep`

   d. Change the `RecordList`'s default **Text** (`My Activities this week`) to match your configuration.

StartPage.LegacyMenuTemplate

The `update.tablet` configuration also contains the **StartPage.LegacyMenuTemplate** Legacy Form.

If this Legacy Form exists, the defined **SMARTBOOK** application menu is displayed on the start page as an Insight Board on top of the first item defined in the **StartPage** Legacy Form.



By default, the **Options** field contains the following JSON definition:

`{"SkipMenus":["LOGOUT","SYSTEM_INFO", "DASHBOARD_STARTPAGE", "SYNCHRONIZA-TION", "FEEDBACK","GLOBALSEARCH"], "useAllKindOfItems":true}`

Description:

- `" useAllKindOfItems ":true` defines that the SMARTBOOK menu items are displayed in the Insight Board on the start page (otherwise it is displayed empty).
- `SkipMenus`: The given menu items given in this JSON array are **not** displayed in the Insight Board.

Forms

Learn how to configure forms in CRM.pad

You can also use CRM.designer's **Forms** area to clone CRM.pad's Generic Forms (that were available in update.seven).

Note: See the Aurea CRM web Administrator Guide for information on Forms.update.seven's Generic Forms and Aurea CRM's Forms share the same namespace when synchronizing. Ensure they have different names assigned.

The following widgets are supported:

- ActionForm
- ClientReport
- CoreReport
- DatePicker
- Details

- InsightBoard
- Label
- MapView
- Query
- RecordList
- TriggerExecution
- WebContent

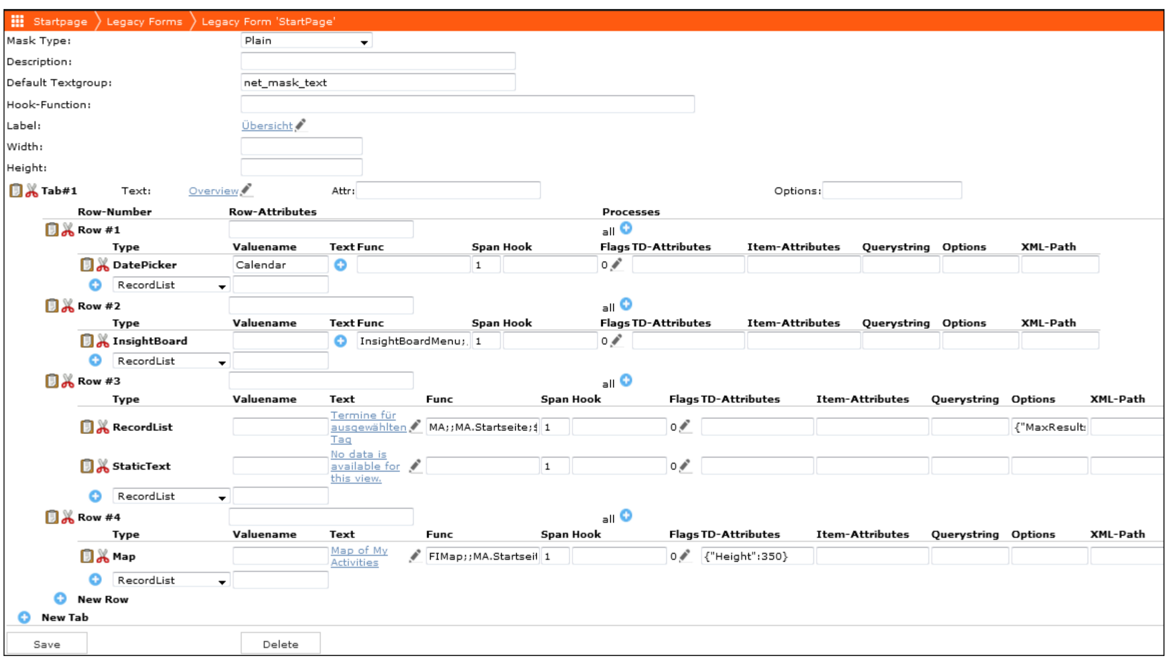
Note: CRM.pad widgets are only shown as placeholders.

Example: Start Page

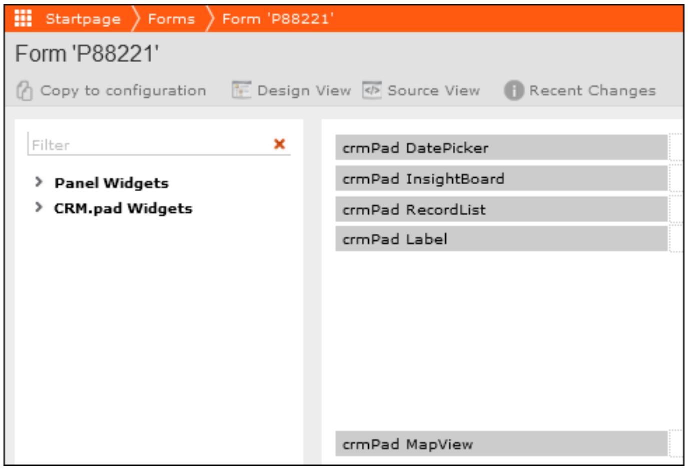
The following example illustrates, how to define the `StartPage` Legacy Form using **Forms**.

`StartPage` Legacy Form:

8

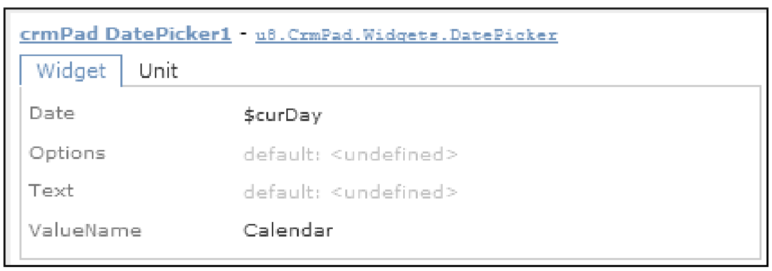


The above Legacy Form results in the following Aurea CRM form:

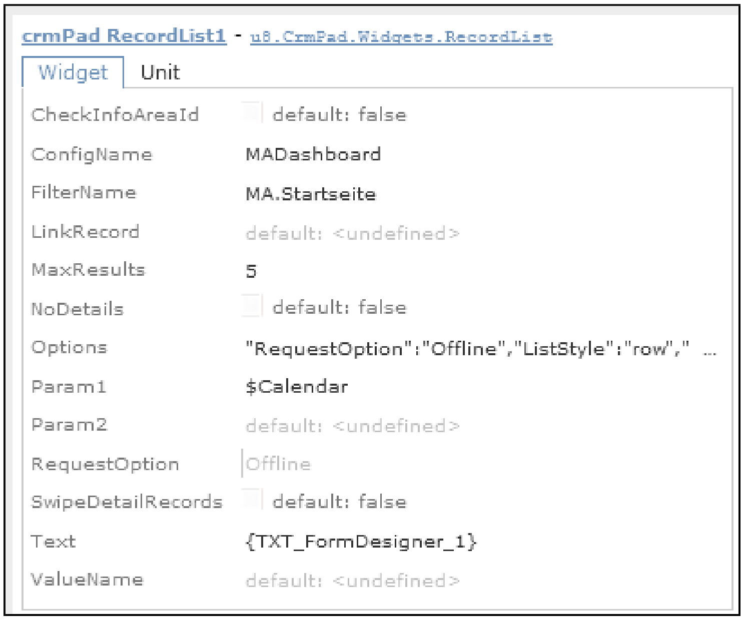


The attributes of the items (= the rows in the Legacy Form) are defined in Aurea CRM Form as shown in the following screenshots:

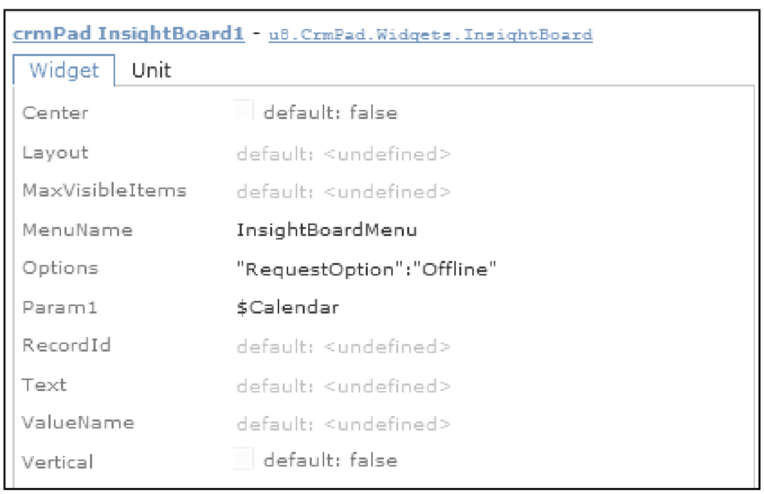
DatePicker:



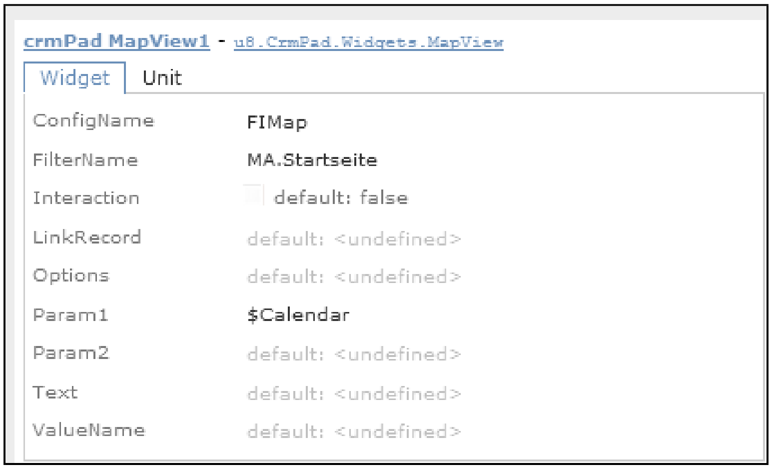
RecordList:



InsightBoard:



Map:



Documents

Documents can be stored in the Aurea CRM database (as binary data) and a record in the D1 or D3 info area is created for each document.

Downloading Documents

You can define how the documents are available in CRM.pad:

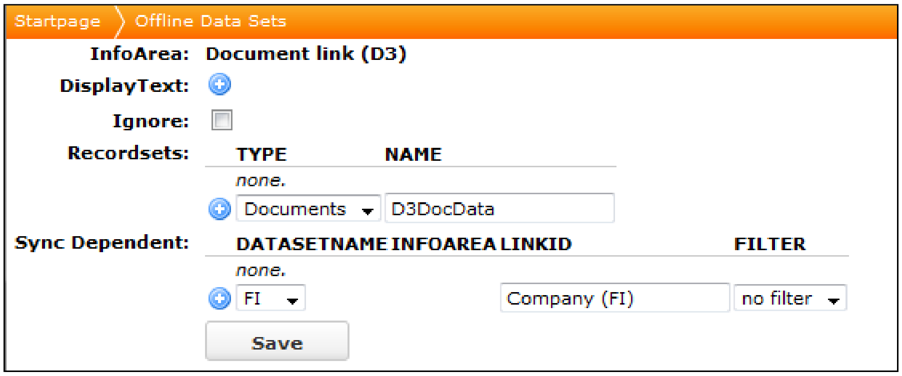
- In the default configuration, only the document info areas (D1 and/or D3 records) are transferred (i.e. the D1 and D3 info areas are included in the offline data set, see Offline Record Set / Offline Data Set on page 51).

  Users can see that a document exists and can download the document if a connection to the server exists.

- If both, the link and the document, are transferred to the device the user can access the document in offline mode as well.

  Note: The volume of the data transferred during synchronization increases and can cause performance problems.

  You need to define the record set for the D1 and D3 offline data set. Switch to the data set and add a new **Recordset** of **TYPE** `Documents` and enter the **NAME** in the following syntax: `[InfoareaID]DocData`, e.g. `D3DocData` or `D1DocData`:



You can define, how documents are transferred using the `Sync.DocumentSync` Web Configuration parameter, see Sync.DocumentSync.

- You can also add documents to the quick add, see Displaying Documents in the Quick Add.

Displaying a Documents List

You can define that a list of documents is displayed. This list can be called with an application menu or as a tab in an info area listing all related documents.

To define that a document list is displayed:

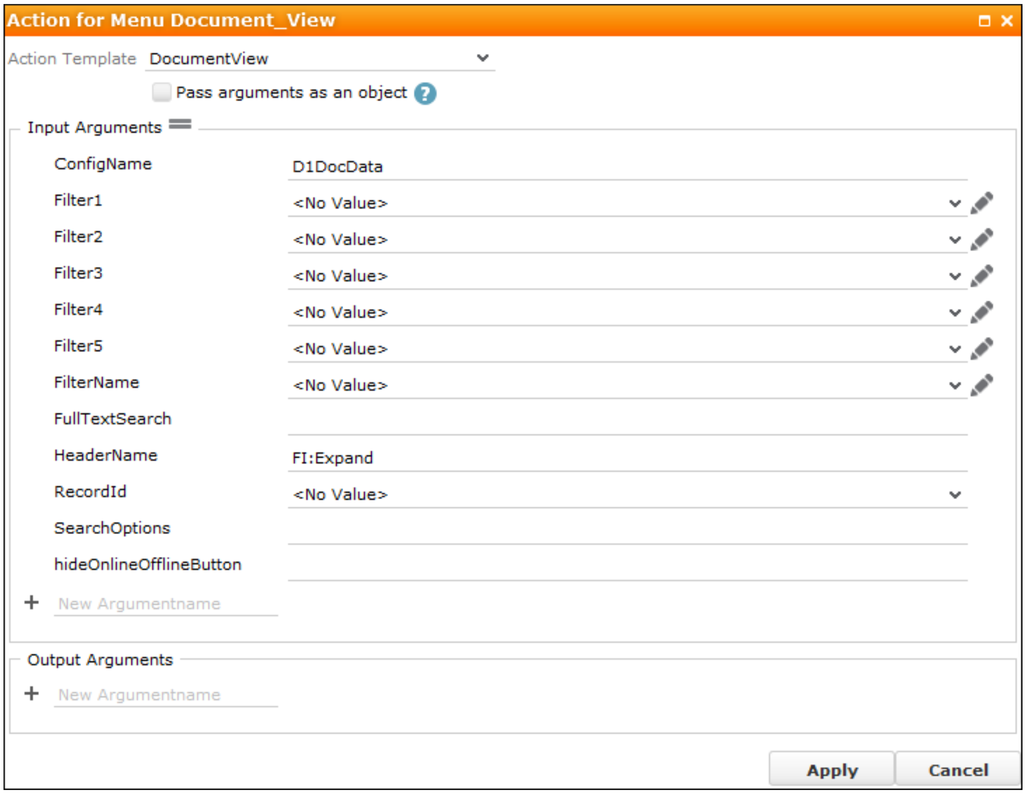
1. Assign `DocumentView` as action template to the application menu action or to the tab.

   For a description of the parameters see Search on page 237.

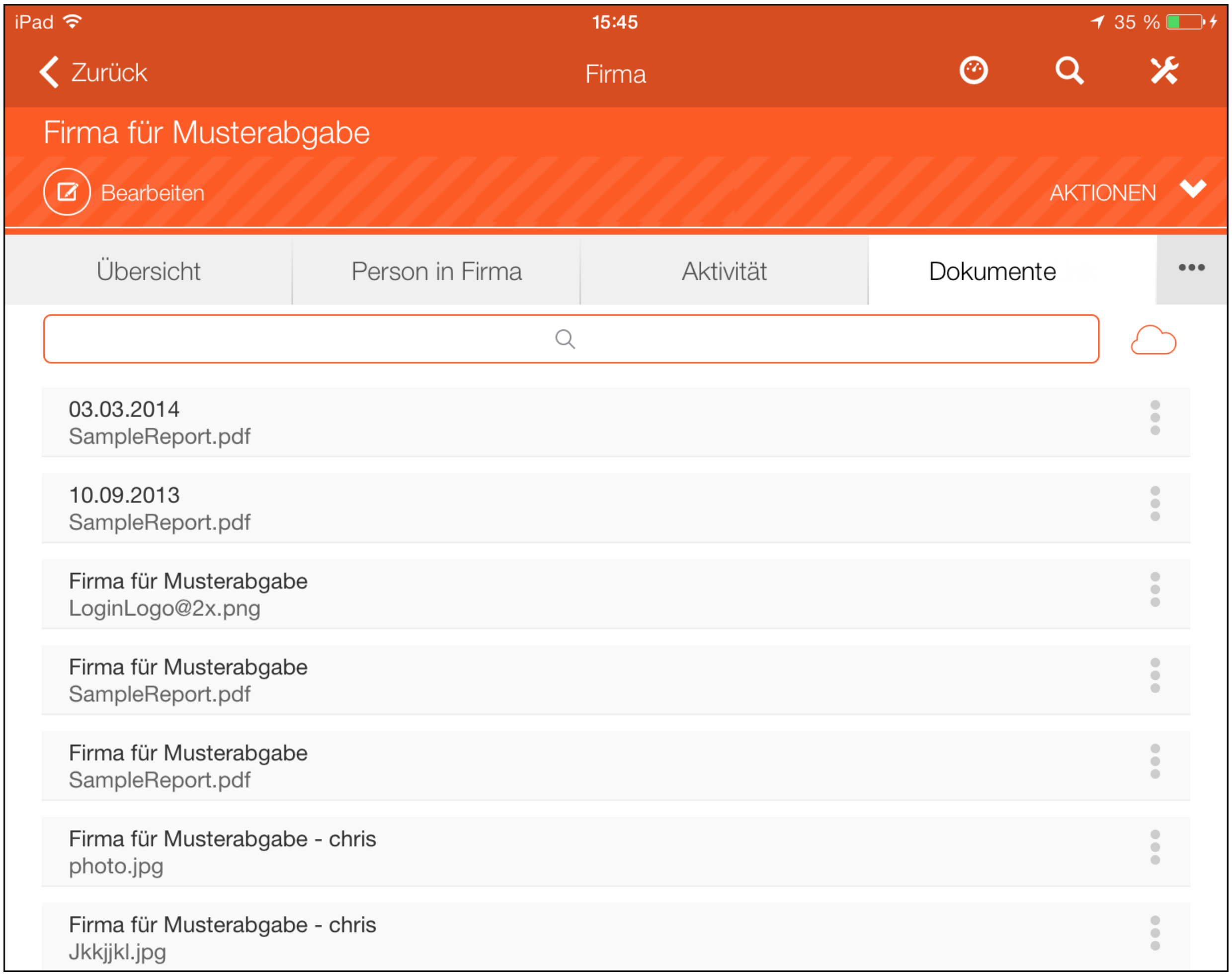
2. You can group the documents. Add a field to the field group that is referenced in the List control and enter `groupingKey` in the **Function** field. For example, if you assign the `groupingKey` function to the **Mime Type** field, the documents are grouped by extension (e. g .jpg or .doc files).

3. If you want to display a different header, define the desired header by specifying the `HeaderName` input argument.

   The following graphic shows how to assign the action template to an application menu item:



This definition results in a document list being displayed for the user:



Displaying Documents in the Quick Add

You can also add documents to the quick add edit pane to display, e.g. the description or a photo of a product.

To add documents to the quick add edit panel:

1. Switch to the `SerialEntry` action call and enter the desired documents and filters in the `DocumentsDefinition` input parameter field in JSON syntax:

   ```
   "[{"name":"<infoarea>","filter":"<filtername>"}]"
   ```

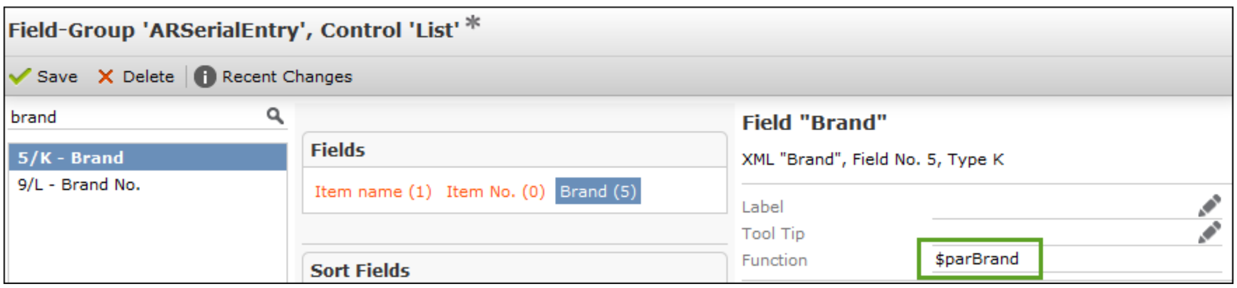
   Example:

   ```
   "[{"name":"D1","filter":"D1.Brand"}]"
   ```

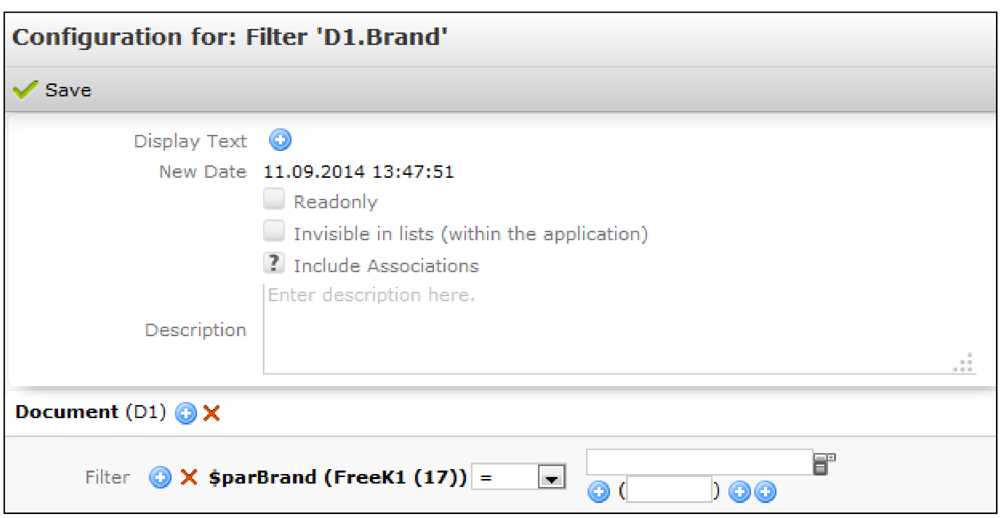
The following JSON string parameters are also available:

- `Style` (optional): Defines the document type and can contain the following values:
  - `IMG` (only images)
  - `NOIMG` (all files except images)
  - `DEFAULT` (all files)
- `addPhotoDirectButtonName`: The name of a button that has the `PhotoUploadAction` assigned, >> Uploading Pictures. If set, the user can upload photos in the Quick Add.

2. Switch to the List control of the Quick Add's field group (e.g. `ARSerialEntry`) and assign the required **Function** that you need for the filter, for example `$parBrand` for the `Brand` field:



3. Define a filter (e.g. `D1.Brand`) and add the function name from the previous step to use the field's content as filter criterion:



Document Inbox

You can define that the user is able to upload documents to the Document Inbox (that is available in the Control Center). The user can then link these documents to a record.

To configure the Document Inbox and linking contained documents to records:

1. Define the `DocumentInbox` Web Configuration parameter, see DocumentInbox. This parameter controls, whether the Document Inbox is displayed and the document types that can be added.
2. If you want the user to be able to edit data (e. g. specifying a file name, entering a description) when they link a document with a record, define an Edit control in the D3 info area containing the desired fields .

   If you assign `Filename` as **Function** to a field, the file's name is automatically entered into this field.
3. Add a button and assign the `FileUploadAction` action, see FileUploadAction on page 96.
4. Add the button the desired info area.

Displaying a Web Page

Learn how to open a web page on CRM.pad.

You can open a web page in CRM.pad, for example, the built-in tutorial for the application. The `WebView` action template is available for this purpose, see WebView on page 140.

You can define that the web page is displayed

- in a separate browser, see Displaying a Web Page in the Web Browser.
- or in a panel on the Details view, see Displaying a Web Page in the Details View.

Displaying a Web Page in the Web Browser

To open the tutorial in a separate web page:

1. Switch to the **Application Menus** area.
2. Add the pre-defined **Tutorial** menu action to the `Smartbook` application menu.
3. Ensure that the following input arguments are defined as follows:

Input Argument	Value
HeaderName	`Tutorial.Tutorial`
Url	`index`
localWebArchiv	`true`

Displaying a Web Page in the Details View

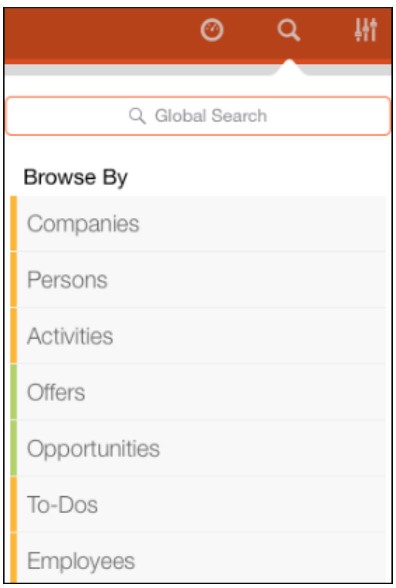
To add a web page in a details tab:

1. Switch to the **Application Menus** area, create a new menu action and assign the `WebView` action call; see WebView on page 140.
2. Add this new menu action to the `Smartbook` menu.
3. Switch to the Details control and add a panel of type `WEBCONTENT_<Con-textMenuName>`, see WEBCONTENT.

Search

Learn how to configure search on CRM.pad.

The user can open the Search area by tapping on the button. The Search button is available throughout the application. The Search area contains the global search as well as all available info area-related searches from the SMARTBOOK menu by default.



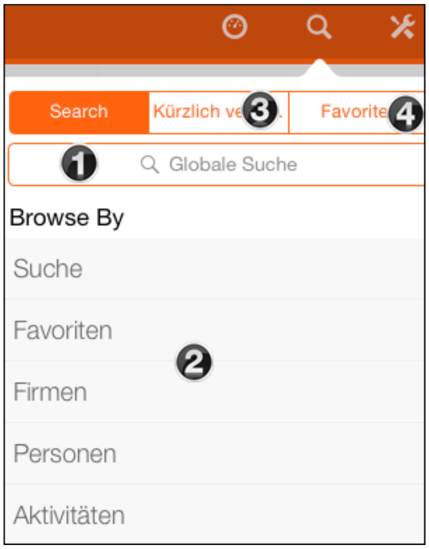
Find as you type is activated for all searches by default. The user can tap on the desired record in the search result to switch to the record's details view.

Configuring the Search Menu

If desired, you can replace the SMARTBOOK application menu with a self-defined search menu.

In this case you need to define the info area-related searches in an application menu named `$AppSearchMenu`.

A self-defined menu can consist of the following items:



1	The global search: Data is searched in all info areas on the device (offline mode).
2	Info area-related searches: You must create a context menu action with the `RecordListView` action template assigned for each info area. All other action templates are ignored.
3	History search: The recently viewed records are listed.
4	Favorites: The records you defined as Favorites are listed.

To define your own search menu:

1. Add an application menu named `$AppSearchMenu`.

   Add the desired items and assign the following action templates:

Application Menu Action Name	Action Template
`$GlobalSearch`	see GlobalSearch on page 99.
`$HistorySearch` **Note:** In addition, you need to add another application menu action named `HistoryView` having the same content structure as `$HistorySearch`.	see HistoryListView on page 100.
`$FavoriteSearch`	see MultiSearch on page 104.
For each info area an application menu action	see RecordListView on page 117.

Vicinity Search

Use the `GeoSearch` action template to configure the Vicinity Search, see GeoSearch on page 98.

The Vicinity Search is a location-based search. Max. three different info areas can be searched. The search starts at the current location by default, however, the user can enter a different starting point.

You need to define the following:

- **Application Menu Action:** Create a new **Application Menu Action** and assign the `GeoSearch` action.
- **Search&List:** Ensure that the GPS coordinate fields were added to the List control that is referenced by the assigned Search&List. The GPS coordinate fields must be defined as described in Adding a Map on page 171.
- **Filter:** The Location filter is applied, see Location Filters.
- **Details control:** If you want to start the Vicinity Search from a company or person record (i. e. the position of that record is used as the starting point for the search), define a button and assign the **GeoSearch** action.

  1. Enter the following JSON in the **AdditionalParameters** input argument:

     `{"GPSX":"$GPSX$","GPSY":"$GPSY$"}`

     If no GPS coordinates are available for your current location, you can define that the address is used as fallback:

     ```
     {"GPSX":"$GPSX$","GPSY":"$GPSY$","GpsCity":"$GpsCity$",
     "GpsStreet":"$GpsStreet$", "GpsCountry":"$GpsCountry$"}
     ```

  2. You must also define the following **Extended Options** for the corresponding fields in the List control (otherwise they cannot be applied):

     - `{"GPS":"City"}`
     - `{"GPS":"Street"}`
     - `{"GPS":"Country"}`.

  3. Make sure that the MAP field configured in CRM Designer for the Company details also has the extended options set for the following fields:

     - `Latitude:{"GPS":"Y"}`
     - `Longitude:{"GPS":"X"}`
     - `Street:{"GPS": "Street"}`

   - `City:{"GPS": "City"}`
   - `Country:{"GPS": "Country"}`

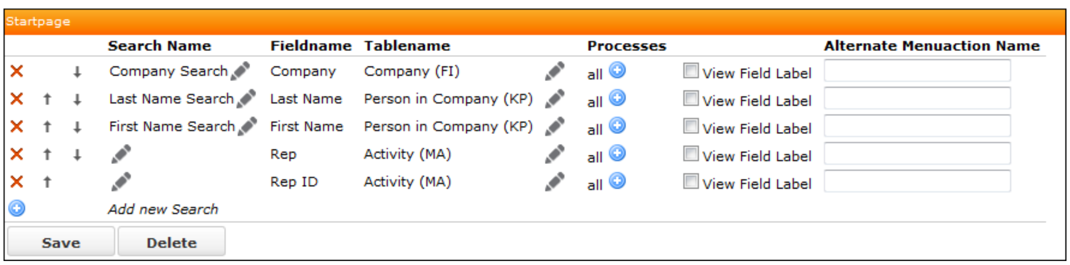
- Further settings:
  - The user can define the radius of the search area with a slider. You can define the max. radius with the `DistanceFilter.MaxValue` Web Configuration parameter, see DistanceFilter.MaxValue.
  - The label displayed are taken from the filter's Display text.
  - If multiple info areas are used, the user can de-select certain filters by clicking on them. The display text of the filter is used as label.

Searching Several Info Areas (Global Search)

The Global Search is included in the default configuration. This search includes more than one info area. You can define the info areas to be searched in the **Quick Search** area. The user can access the Global Search from the application menu.

To define the Global Search:

1. Click on **Quick Search** on the CRM.designer start page:



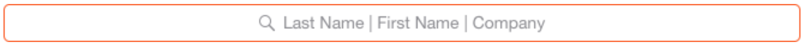
2. Define the fields in the info areas to be searched:
   - Click on (**Edit**) besides an item to change the existing setting.
   - Click on (**Add**) at the end of the list to add a new info area, and then define the settings for the search.

Note: The `FullTextSearch` web configuration parameter does not apply to the global search for performance reasons.

Configuring the Search&List Page for Info Area Searches

After searching for data, the hits are listed in the Search&List page. For the Search&List page define the following configuration units for the desired info areas in CRM.designer:

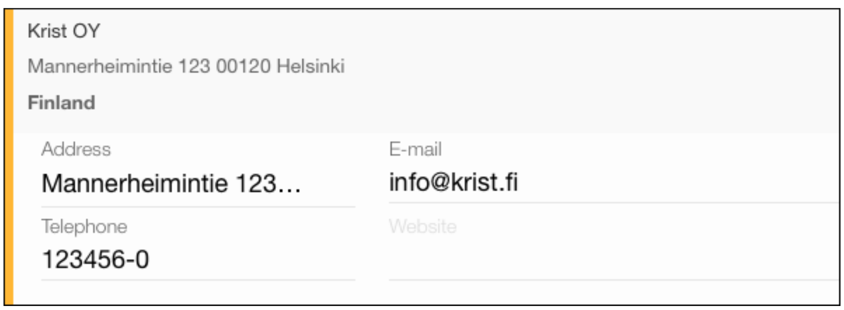
- Search control: Defines the fields that are searched in the database. These fields are shown in the application's search field as ghost text.



- List control: Defines the data displayed in lists.

  You can also define that an icon is displayed for the info area. Assign an image with the same name as the info area ID (e.g. FI for company) in the Expand Page configuration:

- MiniDetails control: Defines the layout of the box that is displayed after the user clicks on .



- Search header: The header is defined in the header group and then assigned in the Search&List configuration. You can add to a header group:
  - The page's title (header text, defined in the Text field).
  - Buttons: If you add buttons, the first three buttons are displayed as quick actions on the left side of the header, the other buttons can be accessed by the **Actions** menu on the right side.

  Further tabs shown (Info Areas). The data of these (sub-)info areas can be accessed using the tabs, see Displaying Data in Additional Tabs.

Displaying Record Details

Learn how to configure displaying record details.

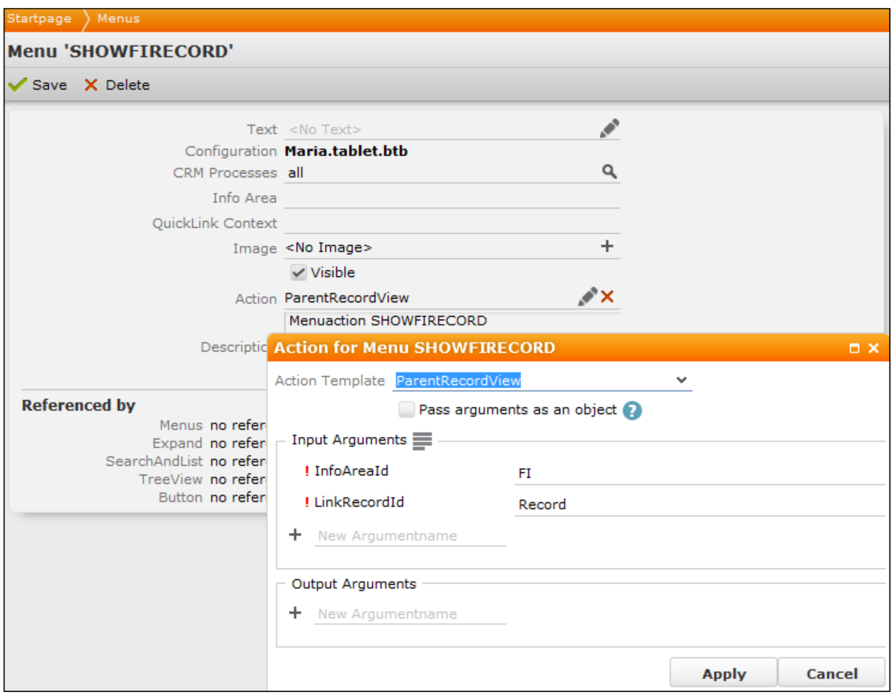
How records' details are displayed is defined in the `RecordView` page. You can add a `RecordView` page only to a record's context-specific menu action in the **Context Menus** area. The `RecordView` page is usually called by the predefined `SHOWRECORD` menu action that displays the record clicked in a list with the default Expand Page Configuration.

The Expand configuration defines which field group and header group is used.

To define the Details page:

1. Switch to the **Context Menu Actions** area.
2. Select `SHOWRECORD` and copy it to your configuration.

   The menu's configuration is displayed.
3. Click on (**Edit**) besides **Action**.



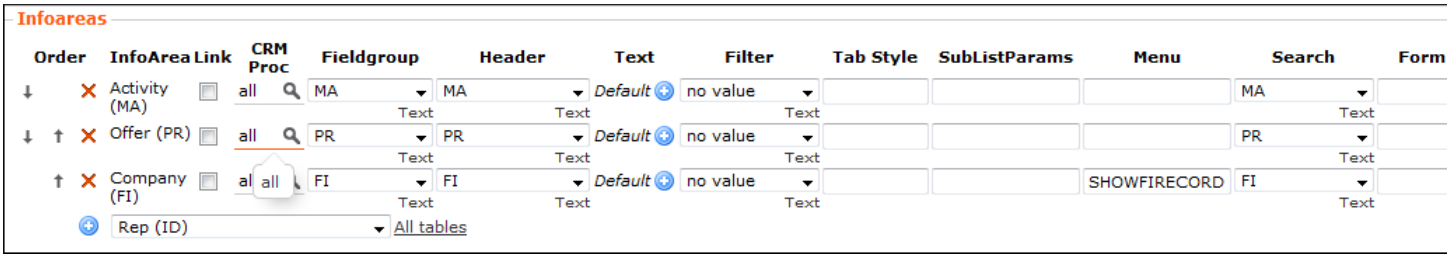
Displaying Data in Additional Tabs

You can define that linked data is available in CRM.pad in additional tabs on the right side of the screen. You can add additional tabs to lists or to Expand views. For example, in Expand view they can contain child info areas, and in lists, as tab could contain the current user's appointments.

To add additional tabs to display data:

1. Switch to the header group of the desired info area.

   The following example shows the Person (KP) info area:



   This Person Details view contains three additional tabs:

   - The first tab contains the person's activities (i.e. a search page with a link record).
   - The second one contains the person's offers (i.e. a search page with a link record).
   - The third one displays the page that was configured in the `SHOWFIRECORD` menu action. You can add this context menu action on any page. This `SHOWFIRECORD` page displays a Details view with the context record as link record and a fix target info area.

2. You can enable **Link** if the info area is not linked to the current info area with the default link.
3. Enter an existing context menu action in the **Menu** field to call the defined action when the tab is displayed.

   Note: You can add a filter to the tabs. Define a context menu action that calls the desired filter and enter this context menu action's name in the **Menu** column.

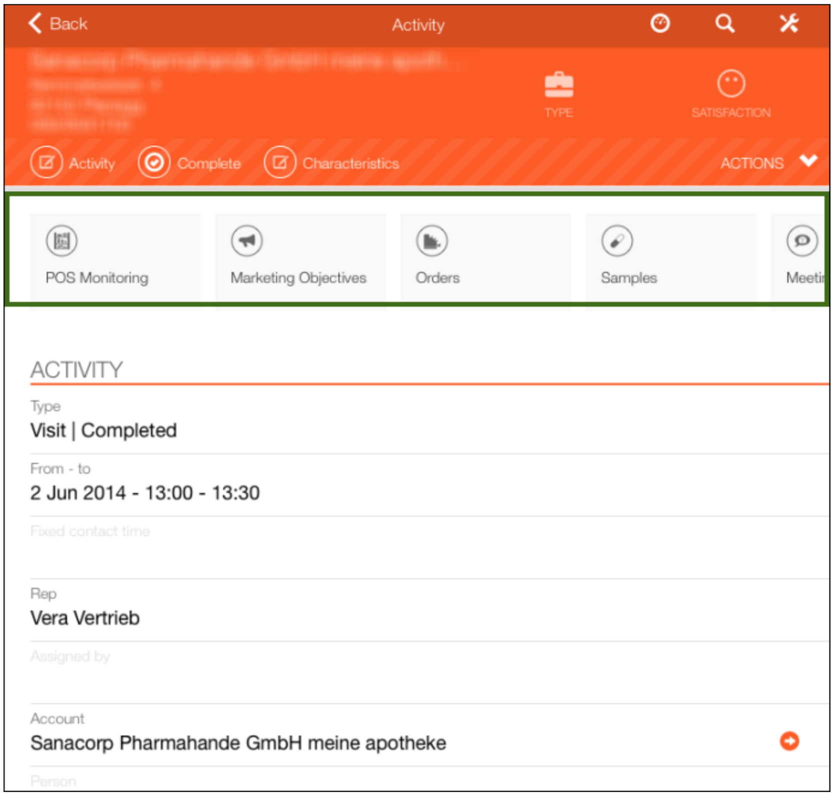
4. Enter the text that is displayed on the tab in the **Text** column
5. To change the text of the first tab, add a sub-info area row on top and set **TabStyle** to "#" in that row.

   Note: **Fieldgroup** and **Headergroup** are not used. These groups are defined by the **Search** field.

Switch Tab Index

You can define that a separate tab (sub-list) is opened when the user tabs e.g. on a button.

For example, the **Process** panel is implemented in the LSI template uses function. In these processes, if a user creates a report by clicking on a button, this report is displayed this in a separate tab (reports can only be displayed in a separate tab).



To configure this action:

1. Assign the `SwitchTabIndexAction` action template to the desired button, see SwitchTabIndexAction on page 137.
2. Define the `TabIndex` input argument. For example, if you enter "2", the tab is displayed as the second tab on the right side.

Time Zones

Learn how to configure timezones on CRM.pad.

As of CRM.pad 2.0.1 you can set a specific time zone with the `System.iOSServer-TimeZone` Web Configuration parameter, see System.iOSServerTimeZone. The time zone is applied after the next full synchronization.

By default, CRM.pad applies the time zone of the last synchronization.

If you change the client's time zone, iPad appointments are displayed based on the "new" time zone, other appointments such as Aurea CRM activities are still displayed based on your last session. This might result in confusion when displaying both, data from your iPad and Aurea CRM activities in the calendar.

After a time zone change all date and time fields are converted and displayed in the selected time zone. In addition, they are also converted to the server time and applied accordingly on the server.

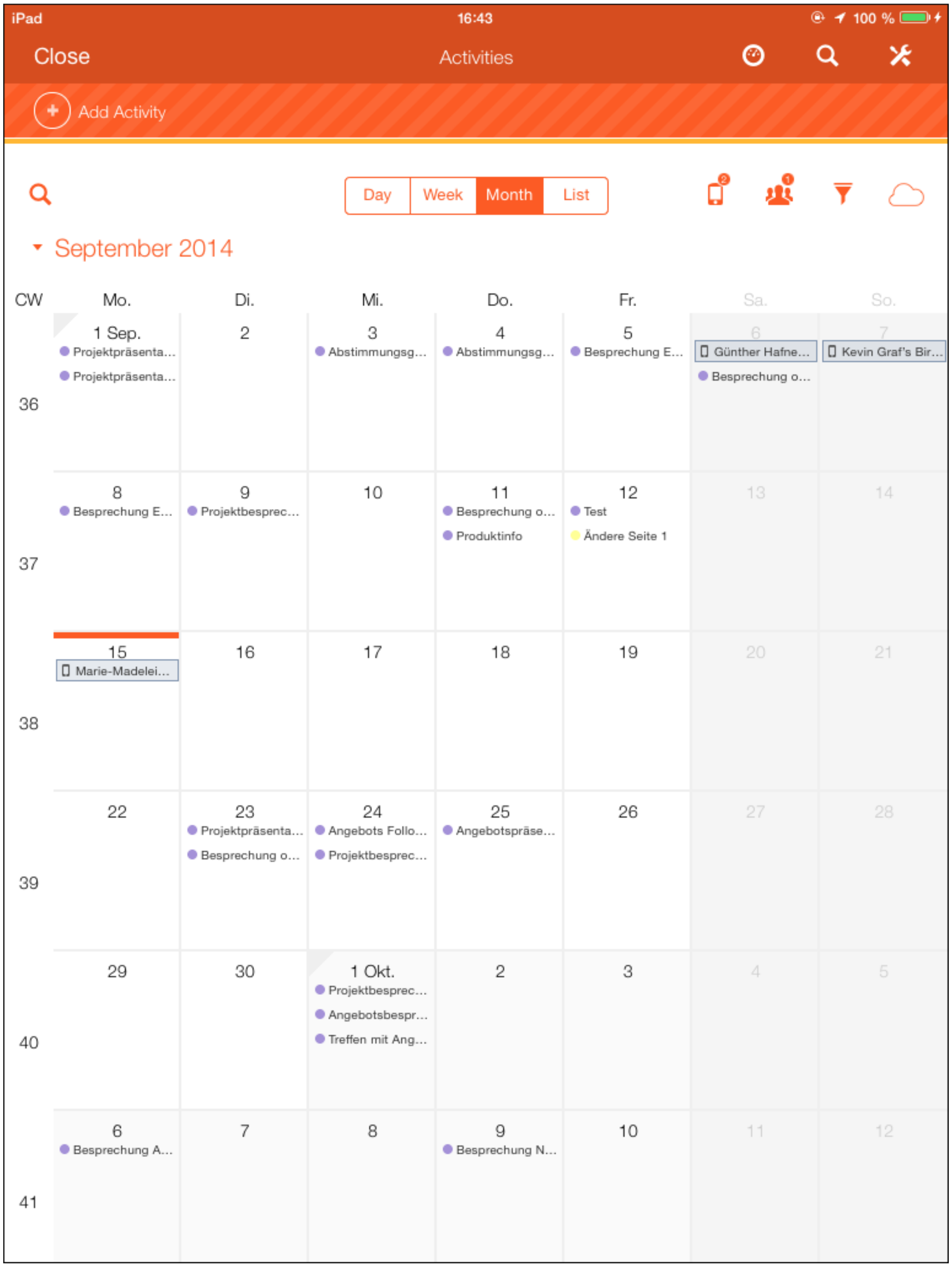
4

Basic Features

CRM.Pad is easy to use and offers numerous features to facilitate your daily activities.

Calendar

By default, activities are displayed in a calendar in CRM.pad.



If the user taps on an item in the calendar the details are displayed in a pop-up.

You can define a calendar view for appointments that includes filters and search options and different calendar views (day, week, month).

The user can also view his appointments in list view.

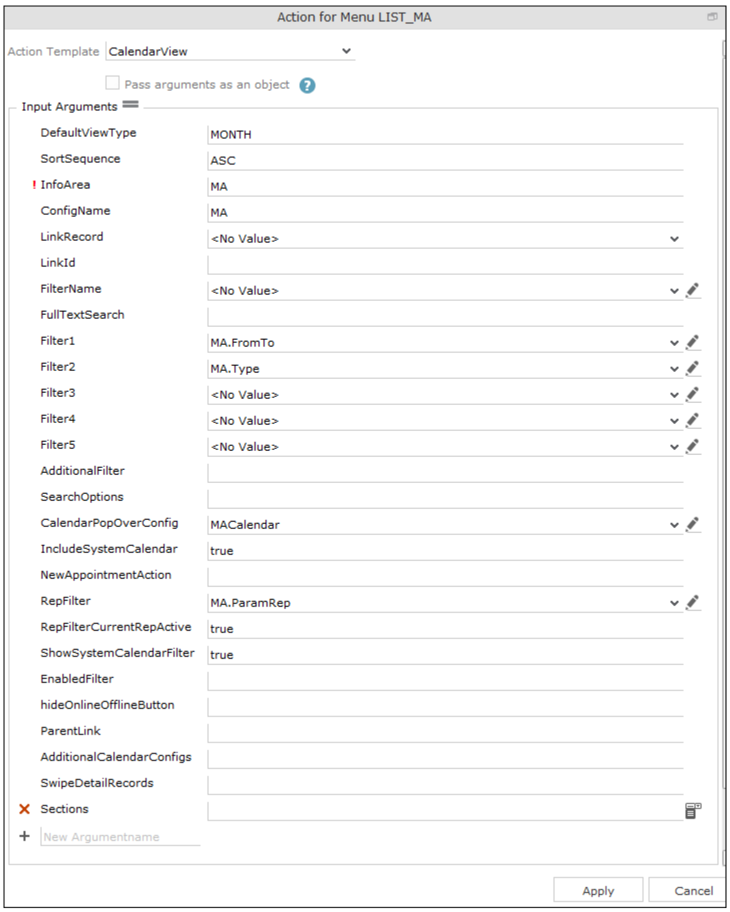
Calendar items created in the iPad calendar are displayed in black formatting. These items cannot be changed in CRM.pad.

Enabling and Configuring the Calendar

You can assign the `CalendarView` action to the desired menu item to display the calendar.

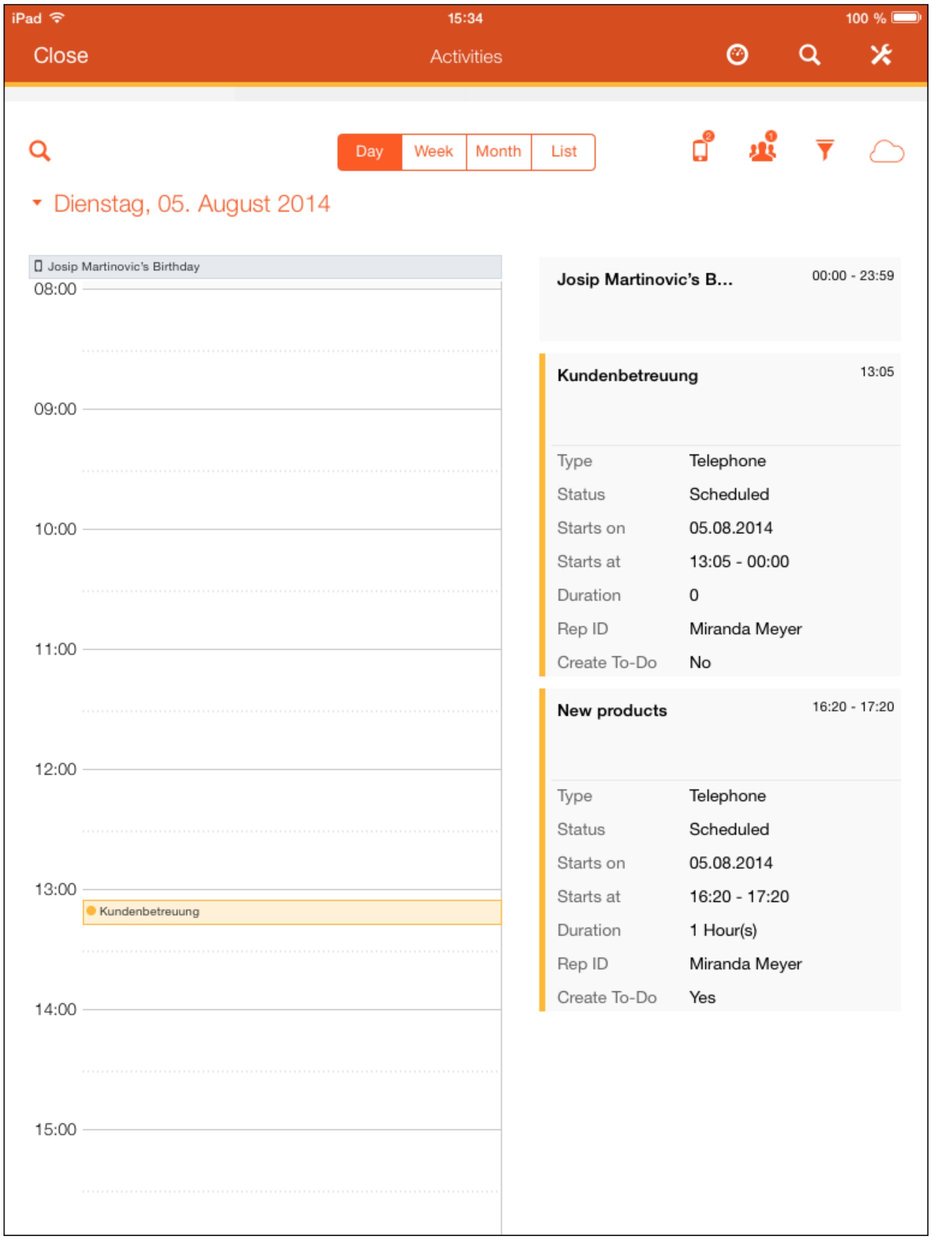
For further details, see CalendarView on page 74.

The following screenshot shows the default configuration included in the `update.tablet` configuration where the `CalendarView` action is assigned to the `LIST_MA` context menu action:



Defining the Calendar Day View

The Day view is included in the default configuration and shows all appointments of the selected day.



To configure the Day view's details:

1. The first line ("heading") is applied from the `MA.Calendar` table caption (same as the Calendar's pop-up).
2. Further fields displayed above the separator (gray line) are defined by the **Function** names in the field group's List control. This field group must be given in the `CalendarView` action's `ConfigName` parameter (default value `MA`), see CalendarView on page 74. The following **Functions** can be assigned:

   `Date`, `Time`, `Type`, `Status`, `RepLabel`, `EndTime`, `EndDate`, `CompanyLabel`, `PersonLabel`

   If some of these functions were not assigned in the field group, the corresponding area is left empty.

3. The fields below the separator are defined in the MiniDetails control of the same field group. If no MiniDetails exist, the Details control's first panel is applied.
4. You can use the `SortSequence` parameter in the `CalendarView` action to define how the appointment list is sorted, see General Input Arguments on page 70.

Adding a New Appointment from the Week or Month View with a Long Tap

Learn how to add a new appointment from week or month view.

To add new appointments in the week or month view with a long tap on a specific date/time, you need the following configuration units:

1. A filter, default: `MA.TemplateCalendarNew`.
2. A button (not included by default) to assign the `NewView` action. In addition to the default parameters, ensure the following input arguments are set to the following values:

Input Argument	Value
AdditionalParameters	{"Date":"$Date$","Time":"$Time$"}
ParentLink	KPFI
SavedAction	Return
TemplateFilterName	MA.TemplateCalendarNew

3. Switch to the `CalendarView` action call and enter the button's name in the `NewAppointmentAction` input argument.

Note: A new appointment can only be added for the current and future days.

Calendar Field Group

Learn about the calendar field group functions.

The MA field group has the following functions in its List control assigned by default:

Field	Function
Date	`Date`
Time	`Time`
Activity Type	`Type`

Field	Function
End Time	`EndTime`
End Date	`EndDate`

Note: The calendar cannot display any records if these functions have not been assigned.

If you also want to display data from other info areas in the calendar, define the List control for the desired field groups.

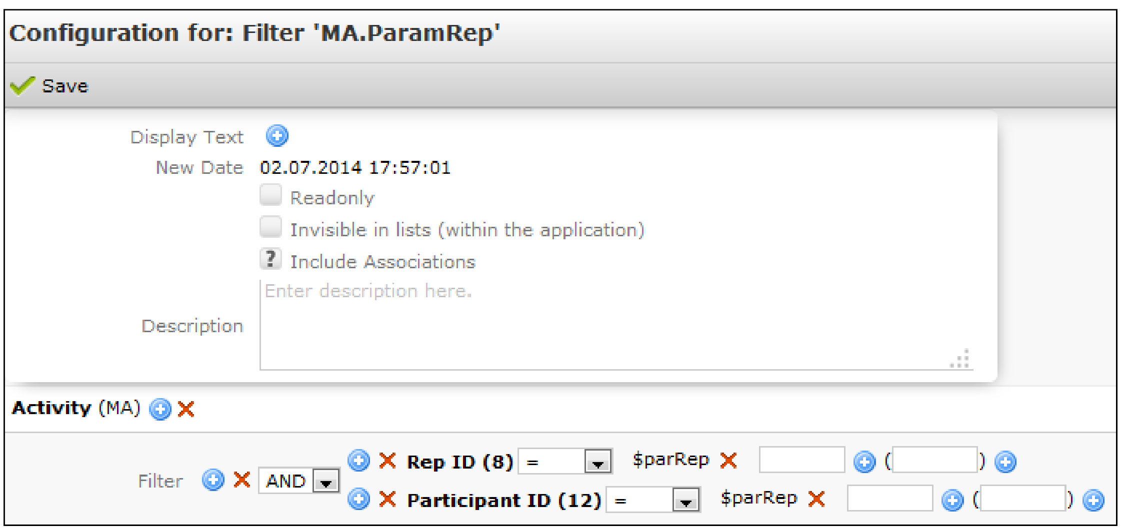
Calendar Rep Filter

You can define a rep filter that you can assign to the `RepFilter` input argument in the `CalendarView` action call.

For further details, see CalendarView on page 74.

The filter needs at least one parameter (**Rep ID** = `$parRep`).

The following example shows the default `MA.ParamRep` filter as included in the `update.tablet` configuration:

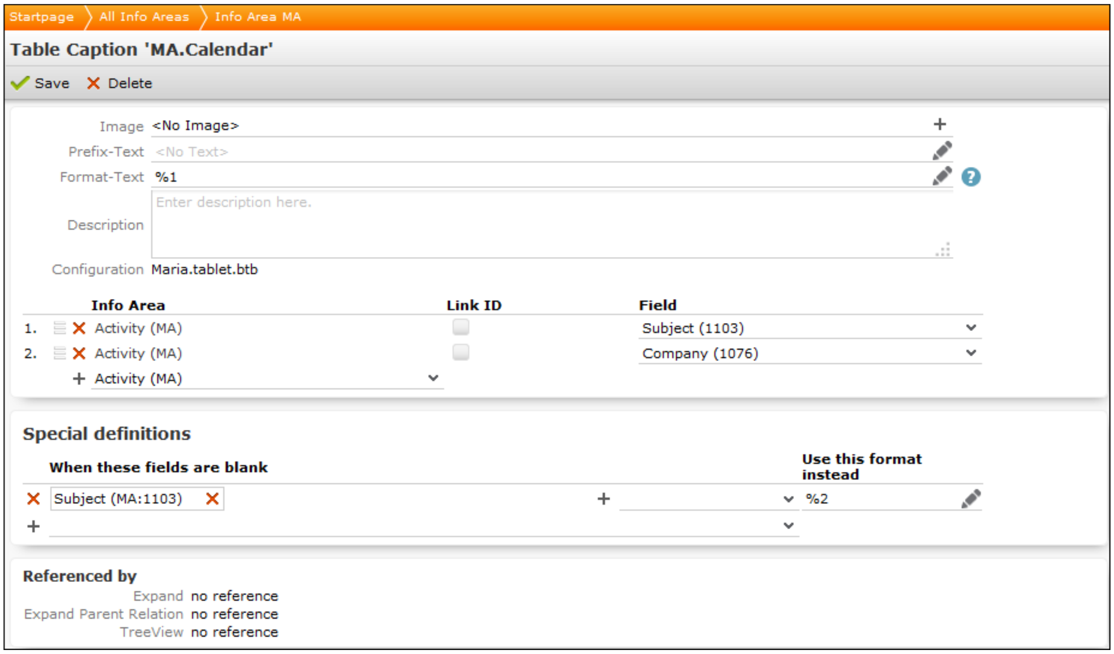


You can add more parameters to further limit the number of reps.

Calendar Table Caption

The table caption defines which information is displayed in the calendar for appointments.

Example:



By default, the calendar tries to read the table caption with the name `<InfoAreaID>.Calendar` (e.g. `MA.Calendar`). If this table caption does not exist, the table caption `<InfoAreaID>` (e.g. `MA`) is used.

Configuring Calendar Pop-ups

Learn how to configure calendar pop-ups.

The `update.tablet` configuration contains the following calendar pop-up configuration:

- The Details control of the `MACalendar` field group in the MA info area containing the following fields:

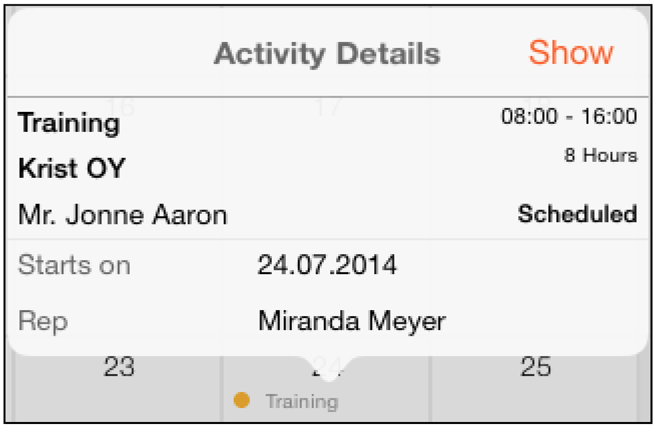
  Company, Duration, Unit, Person, Status, Starts on, and Rep.

  You can add or remove fields or use another field group.

- The `CalendarPopOverConfig` input argument of the `CalendarView` action call, see CalendarView on page 74. This action call is assigned to the `LIST_MA` menu action of the `SMARTBOOK` application menu and the input argument is by default set to `MACalendar`.

If you want to use another field group, set this input argument accordingly.

- The following information is included:
  - The appointment's label defined in the `MA.Calendar` table caption is applied.
  - The appointment's start and end time.
  - Furthermore, the first four fields of the `MACalendar`'s field group are displayed in fixed format (upper part of the following screenshot), all other fields are listed below:



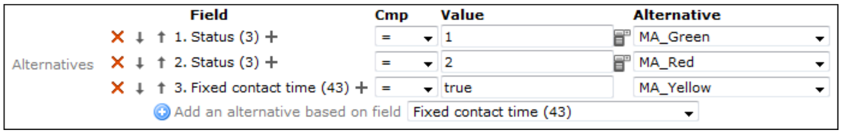
Defining Color Indicators for Appointment

You can define that color indicators are applied to your appointments depending on a specific value of a catalog field.

For example - the appointment type or its status.

To define a color indicator:

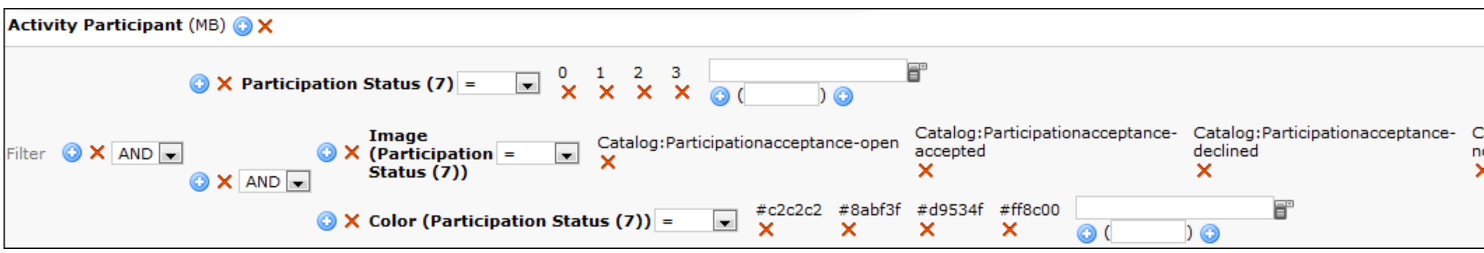
1. Define an Expand configuration for each color that you want to assign and specify its **Color** parameter. Enter values for RGBA in hexadecimal (e.g. #FF0000FF) or floating point (e.g. 1;0;0;1) format. You can also use constants for some colors (e.g. red, blue, green, gray, yellow).
2. Switch to the Expand configuration in the MA info area where you want to assign the color indicators.
3. Add a row for each color indicator in the **Alternatives** area and define the criteria for each color. For example:



4. This example (taken from the LSI industry solution) defines that depending on the appointment data the following alternative Expand configurations are applied:
   - Row 1: If the Status is set to "Completed", `MA_Green` is applied.
   - Row 2: If the Status is set to "Scheduled", `MA_Red` is applied.
   - Row 3: If Fixed contact time is set to true, `MA_Yellow` is applied.

Defining an Image and Color Indicator for a Participation Status

By default, the `MB.CATALOGAcceptaceImage` filter is available in the `update.tablet` configuration to apply a color and image to a participation status.



The given filters are applied to the specified field in the calendar as well as in Edit and New actions.

The criteria must be defined as follows:

- Criteria names defining the image displayed must start with the `Catalog:` prefix, e.g. `Catalog :Participationacceptance -open`.
- Criteria defining the color must be values for RGBA in hexadecimal (e.g. #FF0000FF) or floating point (e.g. 1;0;0;1) format. You can also use constants for some colors (e.g. red, blue, green, gray, yellow).
- Define an image and color for each criterion in the same order than the catalog's field values.

Defining Working Hours for the Week View

You can define which hours are shown when a user switches to the week view.

If the users want to access other hours, they have to scroll up or down. This range is set to 8:00 h to 18:00 h by default.

You can change this default setting using the following Web Configuration parameters:

- CalendarFirstWorkingHour.
- CalendarNumberOfWorkingHours.

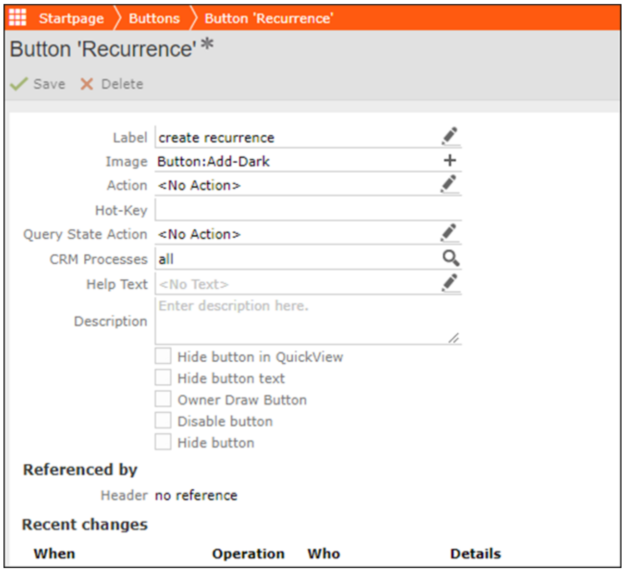
Configuring the Creation of New Recurrence Activity Series from an Existing Activity

The Activity Repitition (WH) infoarea is used to configure the recurring activity.

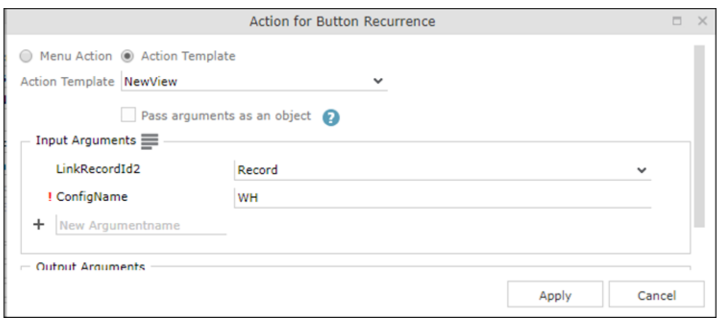
You can allow the user to create a new recurrence series using an existing recurrence Activity configured on the calendar, such that the new recurrence series starts from the day the activity was scheduled to start.

To configure the recurrence series, perform the following steps:

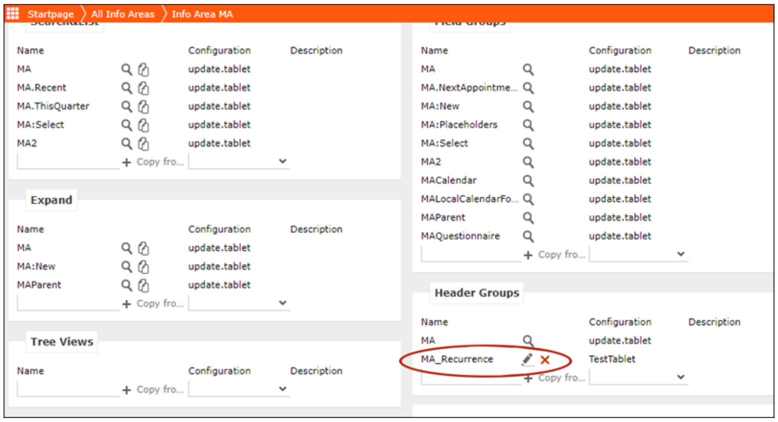
1. Add a button that creates a new Recurrence series. (In this sample procedure we create a button called **Recurrence**)
    a. Select the required CRM.Pad sub-configuration for the user in the CRM.Designer.
    b. Click **Buttons** under **Navigation** panel.
    c. Add a new button named **Recurrence**.
    d. Edit the button as shown in the screenshot below:



e. Configure the Action template (**NewView**) as shown in the screenshot below:

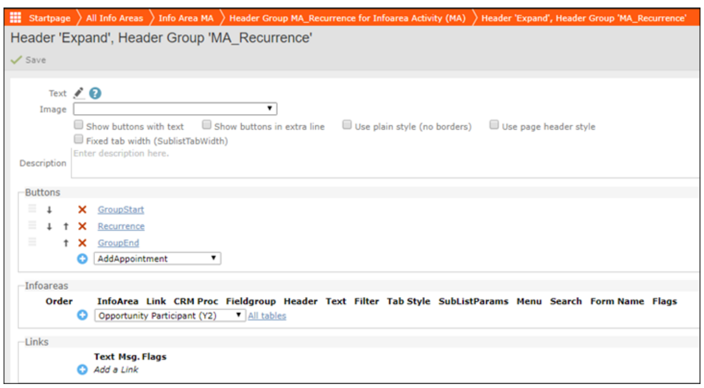


2. Add the button to a new header in the MA Info area.
   a. Create a new header group to display the button. The following screenshot shows the **MA_Recurrences** header group added to the MA info area.



b. Open the header group and add the **Expand** header to the **MA_Recurrence** header group.

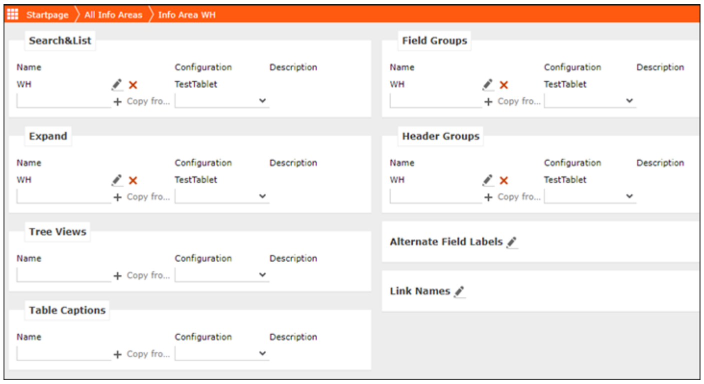
c. Open the **Expand** header and add the **Recurrence** button. See the screenshot below:



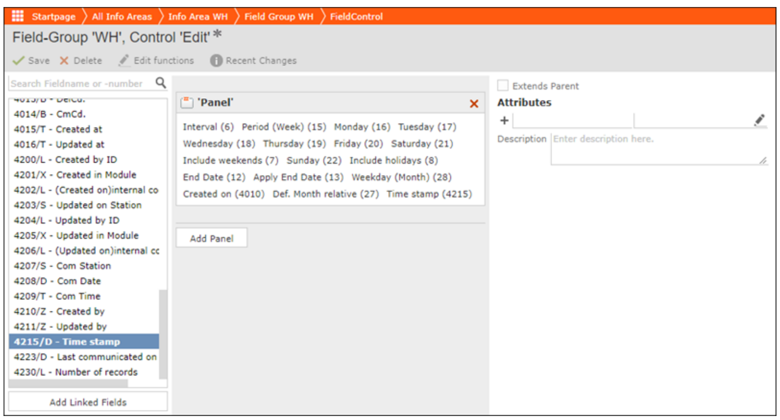
d. Click **Save**.

3. Create a new field group (**WH**) in the Activity Repetition (**WH**) info area with expands for **Details** and **Edit** controls to display the Recurrence button.

   a. Create a new Field Group in the WH info area.



b. Edit the WH field group and add **Details** and **Edit** controls. Add panels to the controls and populate them as shown in the screenshot below, for Edit control:



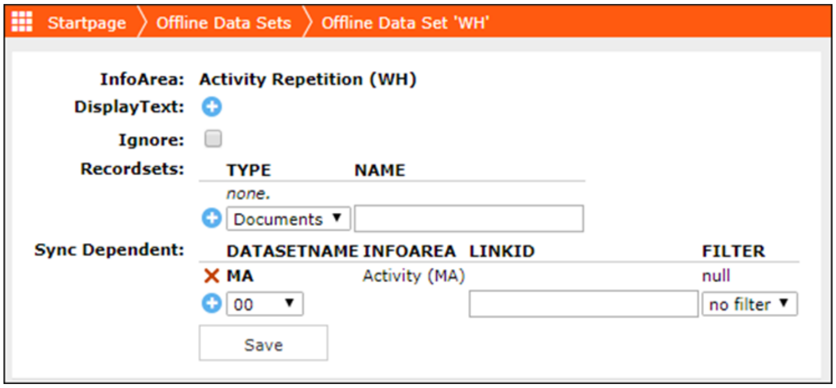
c. Save the controls.

4. Include the Activity Repetition (WH) info area in the Offline Data Sets and Client Data Model for the Tablet configuration for synchronization of CRM.Pad application.

   a. On the Start Page, click **Offline Data Sets** under **Tablet** panel.

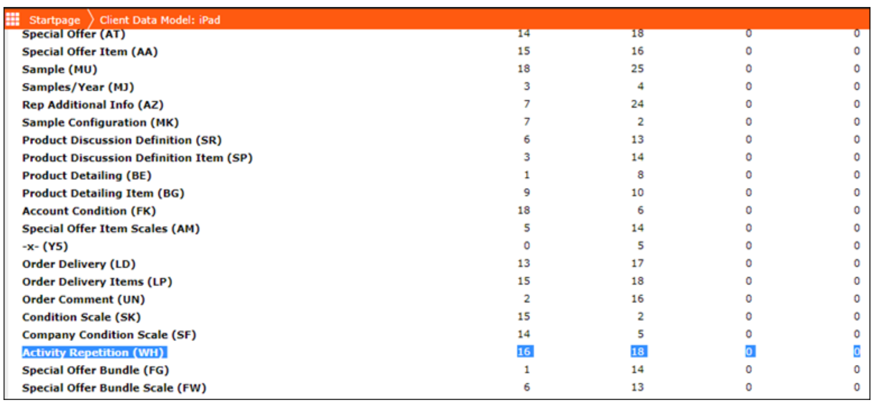
   b. Add the Activity Repetition (WH) info area to the offline data set.

   c. Edit the WH info area data set and add the MA info area to the **Sync Dependent**. Save the configuration.



d. Open the **Client Data Model** under the **Tablet** Panel.

e. Click Update Data Model. The **Activity Repetition (WH)** should display in the Info Area column. See the screenshot below.



Participants

Learn how to define and edit participants.

You can define the participants that are displayed, e. g. for a MA record.

Displaying Participants

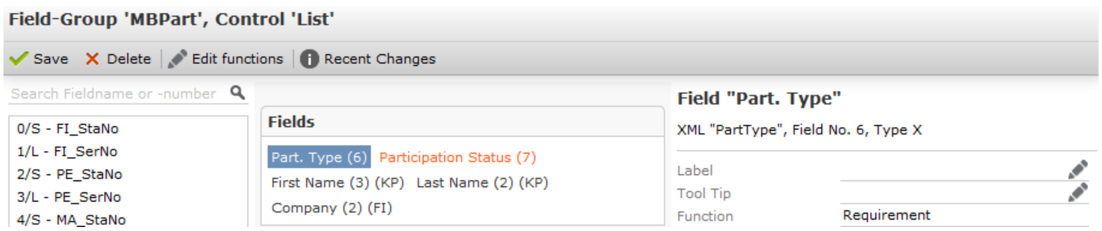
You can define that the participants of an activity (MA) record are displayed. Participants can be reps (ID records) and other participants (MB records).

To add participants to a MA record:

1. Switch to the desired field group's control for the MA info area, and assign the `PARTICIPANTS_<S&L>` type (e.g. `PARTICIPANTS_MBPart`), see REPPARTICIPANTS / LINKPARTICIPANTS.

The update.tablet configuration contains the MBPart Search&List configuration. You can also specify a filter in the Search&List configuration to limit the number of records.

2. Define the List control for the corresponding field group (e.g. MBPart). This List control defines the links that are queried (usually KP and FI) and which fields are displayed. The following image shows the default configuration:



Ensure that the following **Functions** are assigned:

Field	Function
Part. Type	Requirement
Participation Status	Acceptance

3. Ensure that the Search&List configuration (e.g. MBPart) referencing the field group exists.

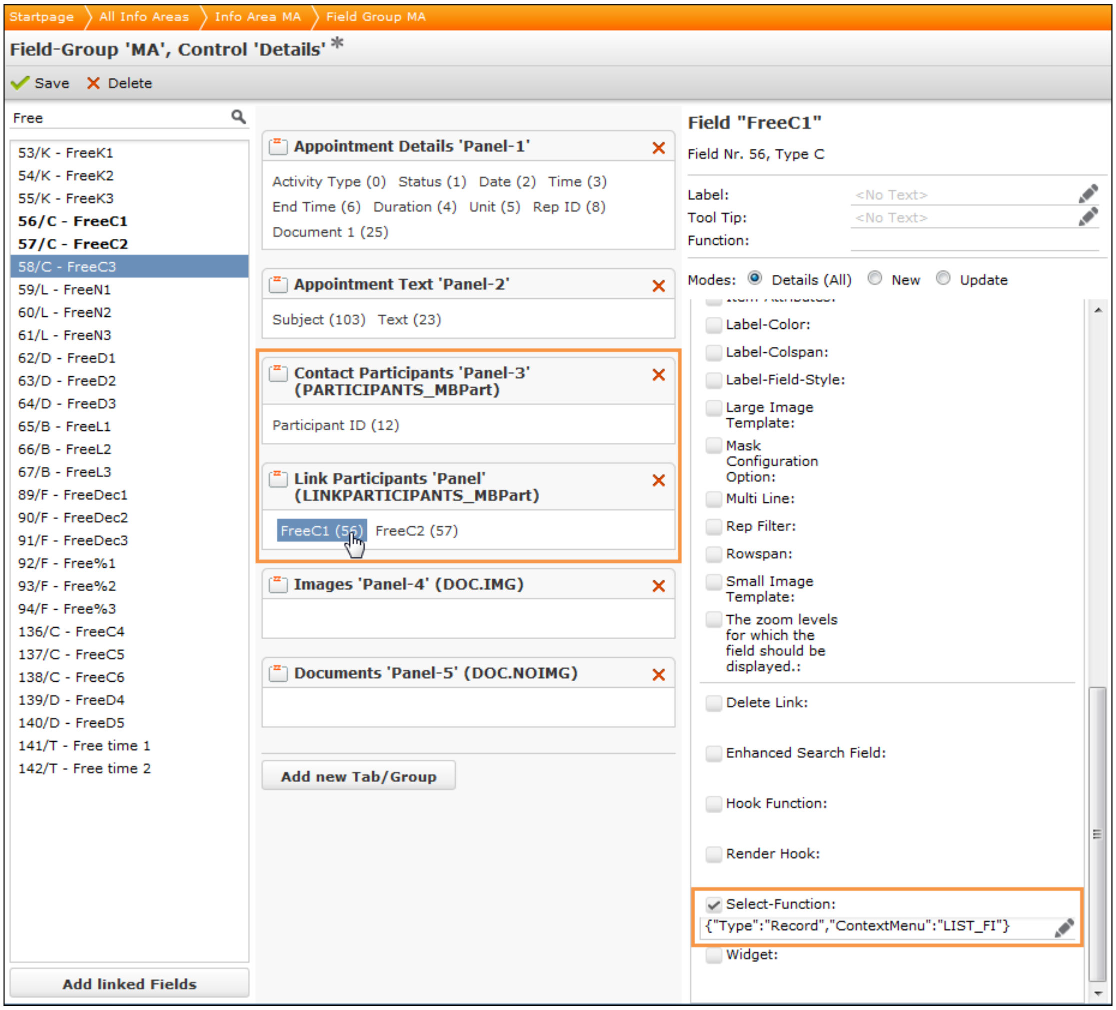
   If you assign SHOWRECORD as default action in the Search&List configuration, the user can open the corresponding company or person by tapping on the participant.

Adding/Editing Participants

You can also define that users can add participants or edit existing ones. This chapter describes the default configuration included in the MA configuration.

To add/edit participants:

1. Switch to the Edit control of the activity's field group (e.g. MA) and ensure that the following two tabs exist:
   - **Internal Participants** (reps) with the REPPARTICIPANTS_<S&L>
   - **External Participants** (MB records) with LINKPARTICIPANTS_<S&L> type assigned, see REPPARTICIPANTS / LINKPARTICIPANTS.
2. Add the field from which participants are read from or written to (in the default configuration the Participant ID field) to both tabs.
3. The Participant ID field is added to both tabs. Ensure that two more fields (FreeC1 and FreeC2) are added to the **Link Participants** tab. You could use any other text field as well, as they are only needed as placeholders for the following two **Select-Functions**:
   - {"Type":"Record","ContextMenu":"LIST_KP"}
   - {"Type":"Record","ContextMenu":"LIST_FI"}



4. These **Select Functions** define the menu actions used for searching companies or persons. In this example the `LIST_KP` and `LIST_FI` menu actions are applied. They are included in the `update.tablet` configuration and are used to call the company and person lists. If you want other lists (e.g. using different filters), add a new context menu action with the `RecordListView` action call assigned.

Accepting or Declining Activities

You can define that activity participants can accept or decline activities.

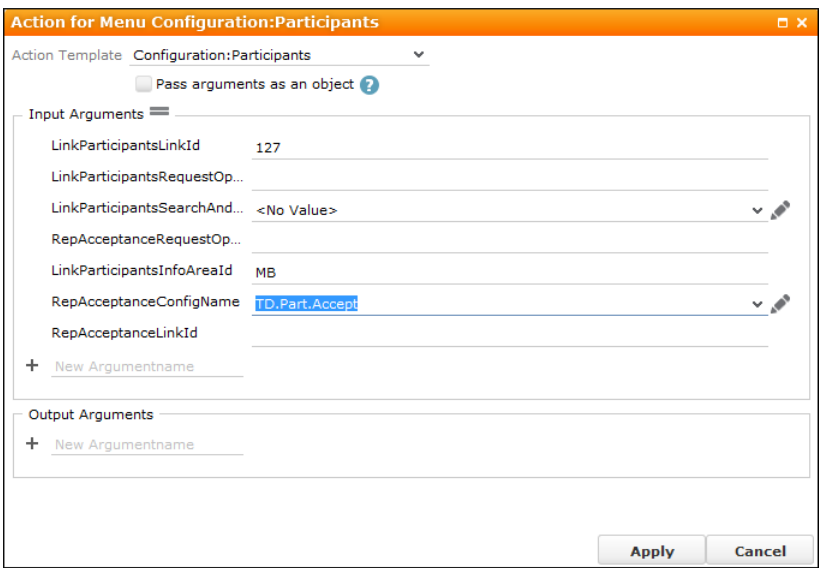
The `update.tablet` configuration contains the following configuration units for this purpose:

- The Edit control of the `TD.Part.Accept` field group (TD info area). in the To-Do (TD) info area:

Ensure that the following **Functions** are assigned:

Field	Function
Rep/Group ID	RepId
Participation Status	Acceptance

- The `TD.Part.Accept` Search&List configuration
- The `Configuration:MAParticipants` context menu with the `Configuration:Participants` action call assigned, >> Configuration:Participants on page 84.
- Assign `Configuration:Participants` as action call, see Configuration:Participants on page 84:



The following business logic is also applied:

- **Server:**

  The acceptance is defined by TD records (that are linked by the `127` LinkId). These records must be created by the server's business logic. Set the **Create To-Do** field (field number 133 in the BTB vertical) in the MA record. These records are only created if the contact lies in the future and was not set to "For Information".

The acceptance can be set in the TD record's Participation Status field of the TD record that is linked to the MA record.

- **CRM.pad:**

  TD records cannot be created in CRM.pad. A different status than **Open** can only be displayed if the TD record already exists. The status icon is only displayed for participants, for whom the TD record already exists.

Displaying Absences or other Info Areas in the Calendar

By default, only appointments are shown in the calendar. You can define that additional company- and person-related data is also displayed, e. g. absences.

To display data from other info areas:

1. Create a context menu action for each info area you want to include in the calendar.
2. Assign the `CalendarView` action to the previously created context menu actions. Only the following input arguments are supported/needed, see CalendarView on page 74:
   - `InfoArea`
   - `ConfigName`
   - `FilterName`
   - `RepFilter`: The filters of all defined calendar configurations (i.e. all info areas displayed in the calendar) are applied.
   - `Filter1` - `Filter5` and `Additional Filter`: You can assign the same filters that were given in the "main" calendar's action call (default: `SMARTBOOK` application menu, `LIST_MA` menu action). These filters can contain parameters.

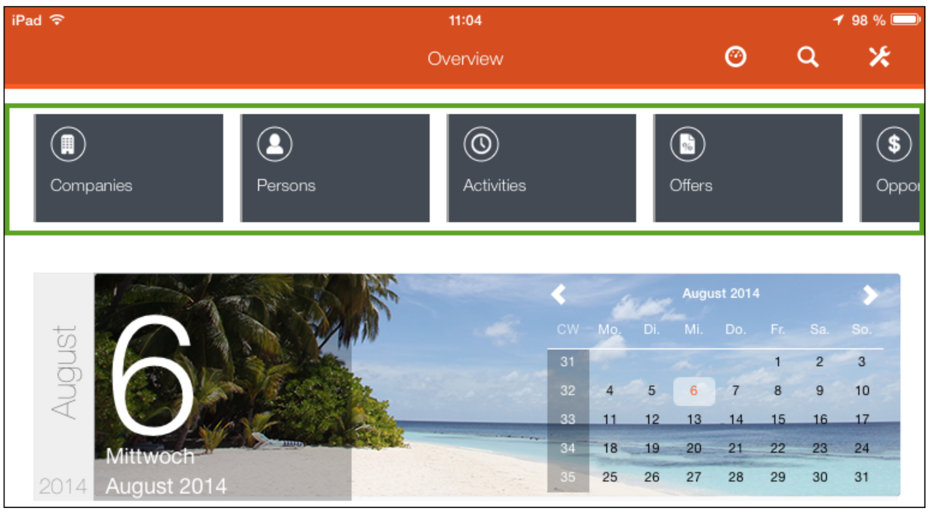
     If no filter is specified, all items of an info area are displayed regardless of the reps selected by the user.
3. Switch to the "main" calendar's context menu action (default: `LIST_MA` menu action) and enter the previously created context menu action names into `AdditionalCalendarConfigs` input argument field of the `CalendarView` action call.

Insight Board

Learn how to configure insight boards and add it to start page and details control.

Three types of Insight Board are available:

- On the login screen: You can add an Insight Board to the login screen with which users can access information like a tutorial, the Aurea blog, twitter or facebook page, see Adding an Insight Board to the Login Pages.
- SMARTBOOK menu: By default, all items that have the `RecordListView` action assigned in the SMARTBOOK menu (defined for previous 1.x version) are displayed at the top of the start page as Insight Board, see SMARTBOOK Menu from Version 1.x.



- Your Insight Boards: You can define your own Insight Boards and add them to the start page or to separate Details pages, see Adding the Insight Board to a Details Control.

Insight Boards can contain any number of tiles. If a user taps on a tile, the assigned action (usually displaying a list) is executed.

Currently, two types of tiles can be added to the insight board, see Configuring an Insight Board:

- Counting the number of existing records that switch to the record list when the user taps on the tile.
- Calling any action template **except** `RecordListView`.

Configuring an Insight Board

To configure an Insight Board:

1. Switch to the **Context Menu** area and add a new context menu, e. g. `InsightBoard`.
2. Add a context menu action for each tile and assign the `InsightBoardItem`, see InsightBoardItem on page 100.
   - You must at least specify the `ConfigName`, i.e. the name of a Search&List configuration. This results in a tile displaying the number of records and switching to the record list when the user taps the tile.
   - You can provide a background color for an Insight Board item by specifying the hexcode of the required color in the `BackgroundColor` input argument.
   - If you want to call any action template **except** `RecordListView`, you need to also specify the `ContextMenu` input argument. The action assigned to the given context menu action is executed, when the user taps on the tile.
3. Specify the `InsightBoard.TitleEnabled` Web Configuration parameter if you want to display a header, see InsightBoard.TitleEnabled.

Adding the Insight Board to the Start Page

To add an Insight Board to the start page:

1. Switch to the **Legacy Forms** area and open the **StartPage** form.
2. Add a row of type `InsightBoard` and enter the following parameters (separated by semicolon) in the **Func** field.
   - `MenuName`: Name of the Insight Board's context menu.
   - `RecordId`: e. g. `ID .$ curRep` for the current rep.
   - `Param1`: Name (with the `$` prefix) of a dependent start page component. For example, the default `StartPage` Legacy Form contains a `DatePicker` row with `Calendar` as **Valuename** assigned. You can therefore use `$Calendar` to display a tile with the selected day.

   Example:

   ```
   InsightBoard;ID.$curRep;$Calendar
   ```
3. You can define further options in JSON syntax in the **Options** field:
   - `MaxVisibleItems`: Limits the number of tiles displayed. If more tiles exist, the user can slide within the Insight Board.

     Example:

```
{"MaxVisibleItems":"2","Vertical":"1"}
```

Adding the Insight Board to a Details Control

To add the Insight Board to a Details control:

1. Switch to the desired Details control and add a new panel.
2. Assign the desired **Type** to this panel, see INSIGHTBOARD / INSIGHTBOARDH / INSIGHTBOARDV.

Contact Times

Learn how user can enter the contact times (like opening hours, call center times, etc).

You need to define the following:

1. A field group for the CT info area containing a List and an Edit control with the following fields and assigned functions:

Field	Function
Weekday	DAYOFWEEK
Type	TYPE
Morning From	FROM
Morning To	TO
Afternoon from	AFTERNOON FROM
Afternoon to	AFTERNOONTO

Note: The Type field is a catalog field. Locked values from a catalog are not displayed in the application.

2. A Search&List for the CT info area
3. An Expand configuration for the CT info area.
4. An Edit button for the FI Expand header with the `ContactTimesEditView` action assigned, see ContactTimesEditView on page 92. Enter the Search&List name from step 1 (that references the CT Edit control) as **SearchList** input argument.
5. An additional panel of type `CONTACTTIMES_<Search&List>` (e. g. `CONTACTTIMES_CT`) in the FI Details control. The Search&List is the one defined in step 1 referencing the CT Edit control.
6. An Offline Dataset for CT.

Reports

You can define the reports generated in Aurea CRM.pad.

These reports are named using the following syntax:

`[$$$YMMDD]_[FI.ID-KP.ID]_[ReportName].pdf`

Configuring Core Reports

To configure a core report:

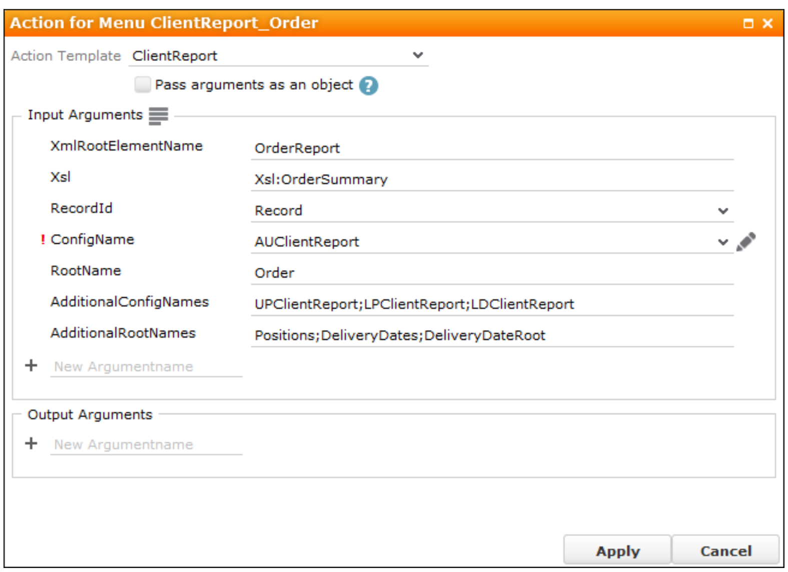
1. Assign the `Report` action to the desired context menu action, see Report on page 124.
2. Enter the name of an existing report format.

Configuring Client Reports

Example for a client report based on an order record:

1. Assign the `ClientReport` action to the desired context menu action, see ClientReport on page 79.

   Example:



You can influence the element names in the XML file:

1. Switch to the List control of the client report's field group (in the following example the `AUClientReport` field group that is available in the `update.tablet` configuration of the OTC vertical).
2. Ensure that the following **Functions** (i.e. the fields' XML element names) are assigned:

Field	Function
Order No.	OrderNumber
Status	Status
Date	Date
Company	CompanyName
Street	Address0
Postal Code	Address0ZIP
City	Address0City
Country	Address0Country
Street	Address1
Postal Code	Address1ZIP
City	Address1City
Country	Address1Country
Order Value	Value
Delivery Date	DeliveryDate
Discount	Discount
Status	Status
Rep ID	Rep

Sending a Client Report by E-Mail

You can also define that a client report can be sent by e-mail, e. g. the order preview that the user can send to the client for information purposes.

To enable the "Send by e-mail" function:

1. Assign the `ClientReportWithAction` action to the desired menu action or button, see ClientReportWithAction on page 81.
2. Define the following input arguments:
   - `SendByEmail`
   - `SendByEmailAction`
   - `SendByEmailAttachReport`
3. You can define with the `SigningConfig` input argument that a client report can only be sent by e-mail after it has been signed by the customer, e. g. an order, see SigningConfig Input Argument.

Pictures

Learn how a user can add photos taken with the device to records (e. g. to persons or products).

Note: Ensure, that the Document (D1) and Document Link (D3) tables are available on the device (i.e. included in the `update.tablet` configuration). An uploaded image is always linked to the current record using the Document Link (D3) table.

Online Upload

To configure that the user can upload pictures in online mode:

1. Add the `UploadImage` button (included in `update.tablet`) to the Expand header of the desired info area, e. g. Company (FI) or Person (KP). This button has the `PhotoUploadAction` action call assigned by default, see PhotoUploadAction on page 111.

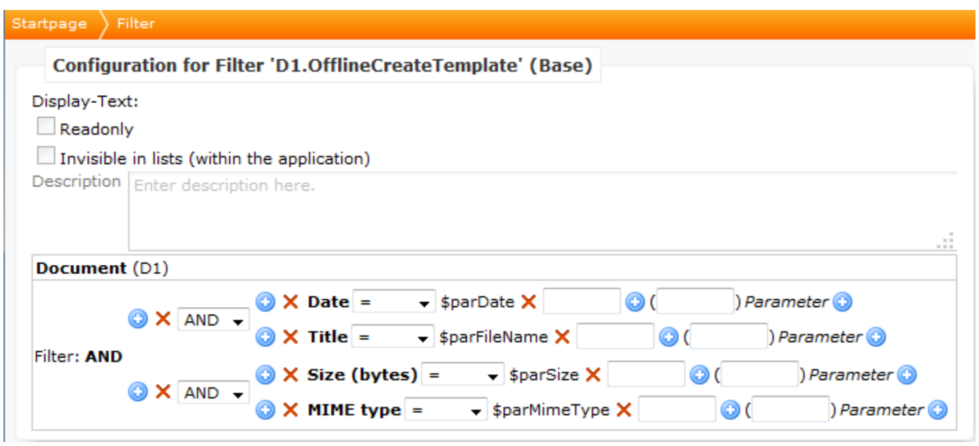
   You can also add a button and assign this action call.

2. The photo upload is configured in the action template's `Config` input argument, see Config Input Argument on page 112.

Offline Upload

To configure that the user can also upload pictures in offline mode you need:

1. The `D1.OfflineCreateTemplate` filter (included in `update.tablet`):



2. The `Configuration :OfflineDocumentCreation` action template must be assigned to the `Configuration:OfflineDocumentCreation` context menu action (both configuration units exist in `update.tablet`), see Configuration:OfflineDocumentCreation on page 84.

3. Define the following input arguments for the `Configuration:OfflineDocumentCreation` action call, see Configuration:OfflineDocumentCreation on page 84:

Input Argument	Value
`DocumentLinkInfoAreaId`	`D3`
`DocumentTemplateFilter`	`D1.OfflineCreateTemplate`(filter included in the update.tablet configuration).

Uploading Pictures Using a Non-WLAN Connection

If pictures are uploaded using a WAN (e. g. 3G) connection, the settings of the following Web Configuration parameters are applied:

- Sync.DocumentUploadMaxSizeForWan.
- Sync.DocumentUploadMimeTypesForWan.

Uploading Pictures in the Quick Add

You can add photos in the quick add with the `DocumentsDefinition` input argument of the `SerialEntry` action, see SerialEntry on page 127.

Note: This option is only available in the OTC vertical.

To add this option:

1. Switch to the desired action call that has the `SerialEntry` action assigned.
2. Define the `DocumentsDefinition`.

Uploading Pictures to Empty Records in the Quick Add

To be able to upload photos to an empty item record:

1. Switch to the Edit control of the field group given in the `DestinationConfigName` input argument, see SerialEntry on page 127.
2. Add a dummy field and set the following **Field Attributes**:
   - **Hide Field** - optional
   - **Empty Line**
3. Enter a **Function** name for the field. This function name and the value to be set can be given in the `hasDocumentsColumnFunctionName` and the `hasDocumentsColumnValue` input arguments, see DocumentsDefinition Input Argument (OTC/LSI/CG only). The `DocumentsColumnValue` is usually `1` or `true`, however, it depends on the given field's type.

Uploading Pictures to the Details/Edit Page

You can add a picture to your Details and Edit view, e. g. for person records.

To add a picture to the Details/Edit view:

1. Switch to the desired control (Details/Edit) of the desired info area, e. g. Person.
2. Add a FreeC field, e.g. FreeC1, to the desired panel. This field holds the image key.
3. Enable the **Image** field attribute and - if desired - limit the image's size. The following example shows the default value applied if no JSON is given:

```
{"previewWidth":"90","previewHeight":"120"}
```

QR Codes / vCard Scanner

You can define how users can scan QR codes with their CRM.pad.

You can configure CRM.Pad application to scan any QR Code generated by a third party to add and edit data fields and add any data to the database. You can also configure a plain text QR Code that can be associated with any text field using the `qrcode.text` function. The following table shows the list of functions that can be associated with field names in a broad sense:

Field	QR Code Function
QR Code text	qrcode.text
Last Name	vcard.n.lastname
First Name	vcard.n.firstname
Tel1	vcard.tel+work+voice
Mobile Tel	vcard.tel+cell
E-Mail	vcard.email+work
Birth Date	vcard.bday
Internal Name	vcard.org.0
Postal Code	vcard.adr.postalcode
Ort	vcard.adr.locality
Street	vcard.adr.street
Country	vcard.adr.country
Website	vcard.url
Department	vcard.org
Position	vcard.org.0

The following is a QR code sample (vCard 4.0) containing the data listed below:

```
    BEGIN:VCARD
    VERSION:4.0
    N:Mustermann;Erika;;;
    FN:Erika Mustermann
    ORG:Wikipedia
    TITLE:Oberleutnant

PHOTO;MEDIATYPE=image/jpeg:http://commons.wikimedia.org/wiki/File:Erika_Mustermann_2010.jp

    TEL;TYPE=work,voice;VALUE=uri:tel:+49-221-9999123
    TEL;TYPE=home,voice;VALUE=uri:tel:+49-221-1234567
    ADR;TYPE=home;LABEL="Heidestraße 17\n51147 Köln\nDeutschland"
    :;;Heidestraße 17;Köln;;51147;Germany
    EMAIL:erika@mustermann.de
    REV:20140301T221110Z
    END:VCARD
```

Configuring QR Code Scanning in EditView and NewView for any Type of QR Code

Learn how to configure QR code scanning in editview and newview.

The following example procedures show the following:

1. Adding a button to scan and edit (**EditView**) the First Name and Last Name fields in the field control mask of the KP infoarea. See, QR Code Scanning to Add a Person in Company in EditView.
2. Adding a button to scan and add a new Person (EditView) to the Person Info area using plain text in the QR Code. See, QR Code scanning plain text to add a person in EditView.
3. Adding a button to scan and add a new contact person (NewView) to the Person in Company (KP) info area. See, Configuring QR Code Scanning for Business Cards.

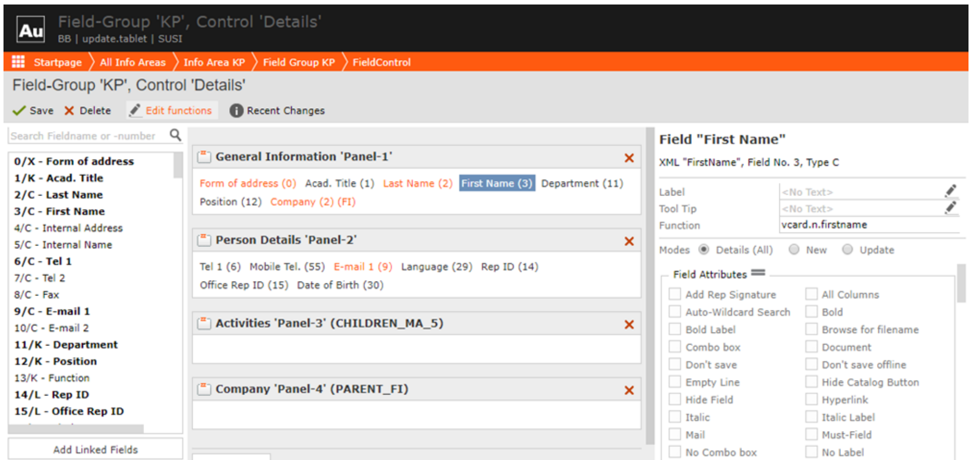
QR Code Scanning to Add a Person in Company in EditView

Perform the following steps in the CRM.Designer to add a person in EditView by scanning a QR Code:

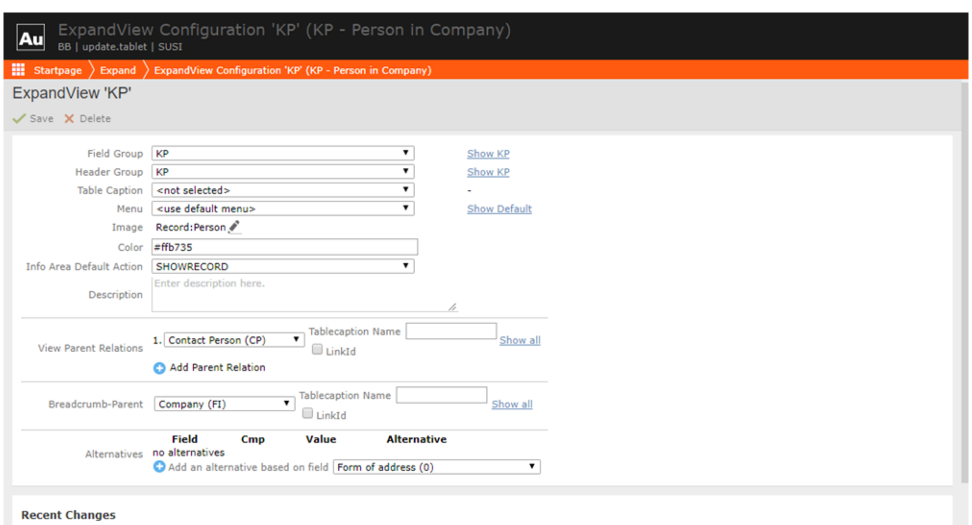
1. Open the All Info Areas page and select KP (Person in Company) info area. The Info Area KP page displays.
2. Select the KP field group for the update.tablet configuration. The Field Control For ‘KP’ page displays.

Note: In this sample configuration we use the Person in Company (KP) info area. But this is applicable to any infoarea.

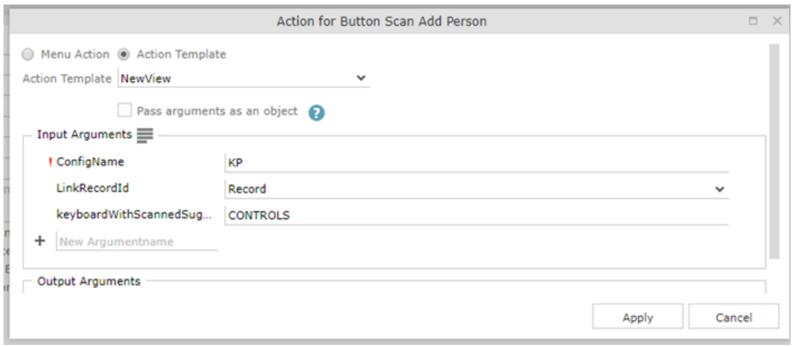
3. Select the Details Control. Set up the panels as shown in the following screen shot. Select the field First Name (3).



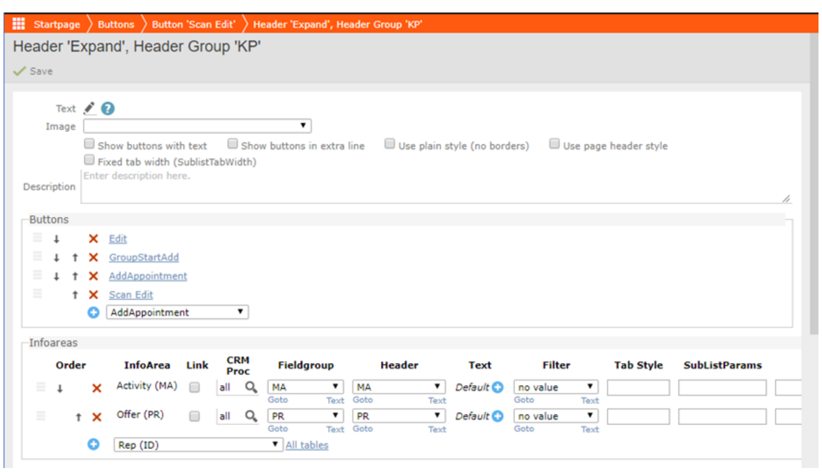
4. Enter the value vcard.n.firstname for the Function attribute of the Field “First Name”.
5. Select Last Name (2) and enter the value vcard.n.lastname for the Function attribute of the Field “Last Name”.
6. Click Save.
7. Go to the Start Page and click Expands under Info Area. The All Expands Page displays.
8. Select KP info area and copy the default configuration. The ExpandView ‘KP’ page displays.
9. Set up the ExpandView as shown in the screenshot below:



10. Click Save. Go to the Start Page.
11. Add some buttons to Scan the Person details using the QR Code.
12. Add a new button, Scan Edit and copy from the Add Person button. The Button ‘Scan Edit’ page displays.
13. Select the EditView action template and open the action dialog. The Action for Button Scan Edit dialog displays.
14. Configure the keyboardWithScannedSuggestion argument with CONTROLS as the input value. Click Apply. See the screenshot below:



15. Click Save and go to the Start Page.
16. Add the button to the expand view header. Click Headers. The Headers page displays.
17. Open the KP header group. The Header Group KP for Info Area Person in Company page opens.
18. Select Search expand option. The Header Search, Header Group 'KP' page opens.
19. Select Scan Edit button from the drop-down list in the Buttons pane. Click **+** to add the button to the Header. See the screenshot below:



20. Click Save. The configured person is now able to scan and edit the Person in Company details using the QR Code.

QR Code scanning plain text to add a person in EditView

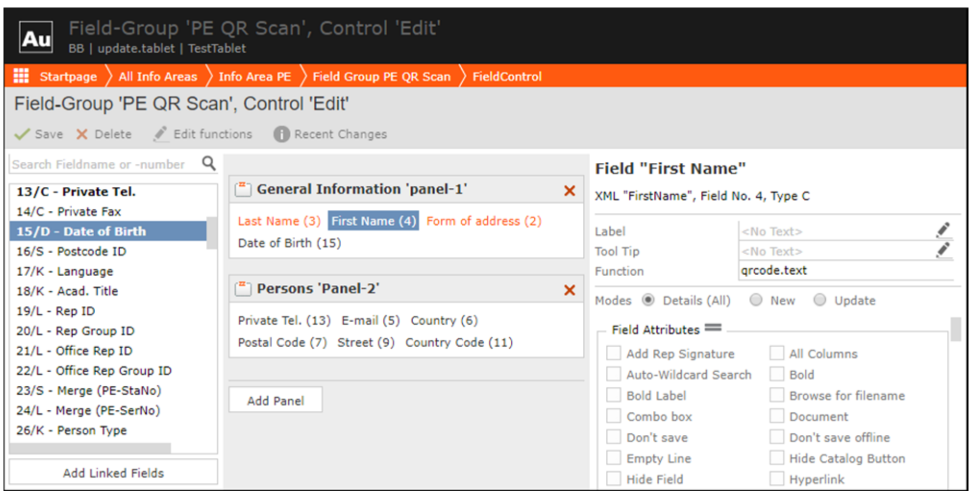
Perform the following steps in the CRM.Designer to add a person in EditView by scanning a QR Code:

1. Open the All Info Areas page and select PE (Person) info area. The Info Area PE page displays.
2. Add a new field group PE QR Scan for the update.tablet configuration. The Field Controls for 'PE QR Scan' page displays.

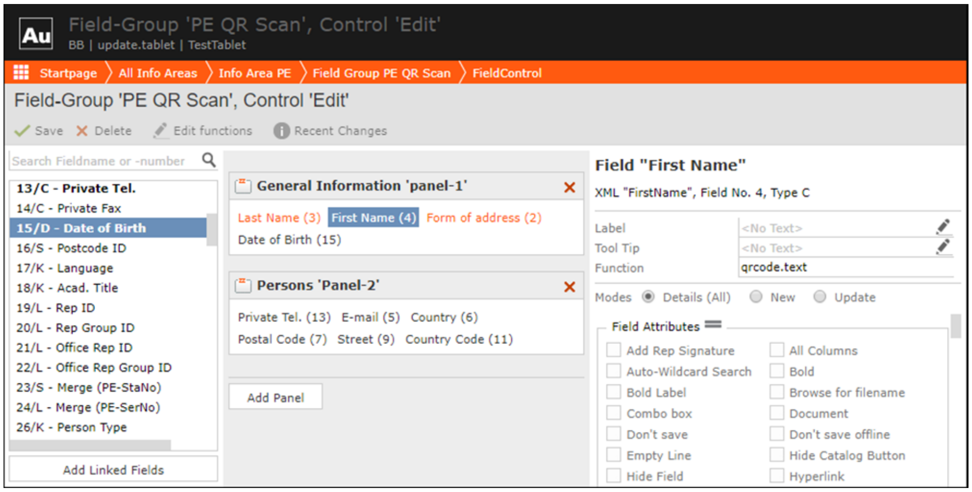
Note: Note: In this sample configuration we use the Person (PE) info area. But this is applicable to any infoarea.

---

3. Select the Edit Control. Set up the panels as shown in the following screen shot. Select the field First Name (4).



4. Enter the value qrcode.text for the Function attribute of the Field “First Name”.
5. Click Save.
6. Go to the Start Page and click Expands under Info Area. The All Expands Page displays.
7. Create new expand configuration PE_Scan for the Person (PE) info area and copy the default configuration.
8. Open the PE_Scan expand view. The ExpandView ‘PE_Scan’ page displays.
9. Set up the ExpandView as shown in the screenshot below:



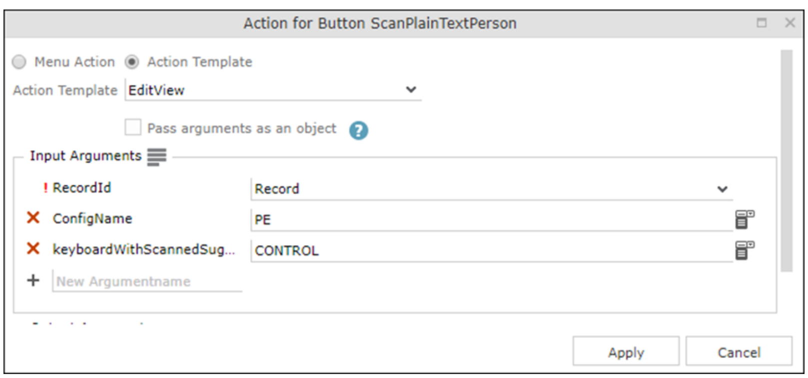
10.Click Save. Go to the Start Page.

11.Add a button ScanPlainTextPerson to and copy from the Edit Person button.

12.Open the ScanPlainTextPerson button. The Button ‘Scan Edit’ page displays.

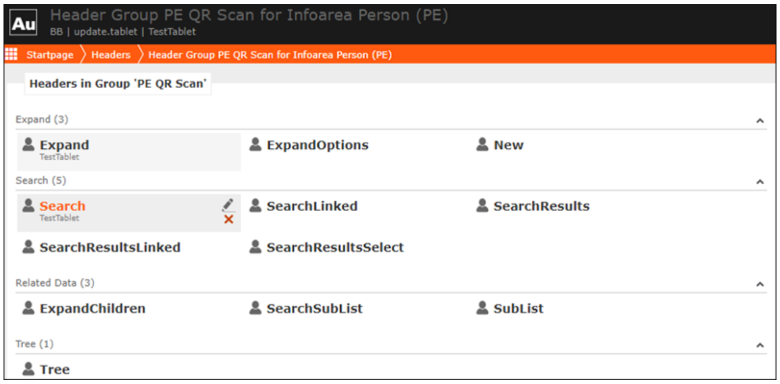
13.Open the Action template. The Action for Button ScanPlainTextPerson dialog displays. Select the EditView action template.

14.Configure the keyboardWithScannedSuggestion argument with CONTROLS as the input value. Click Apply. See the screenshot below:

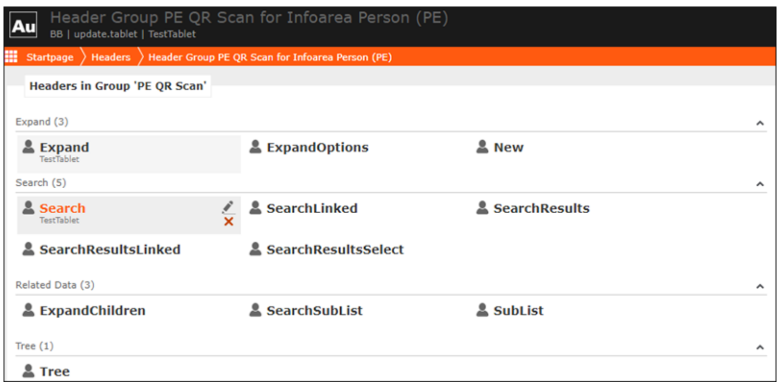


15.Click Save and go to the Start Page.

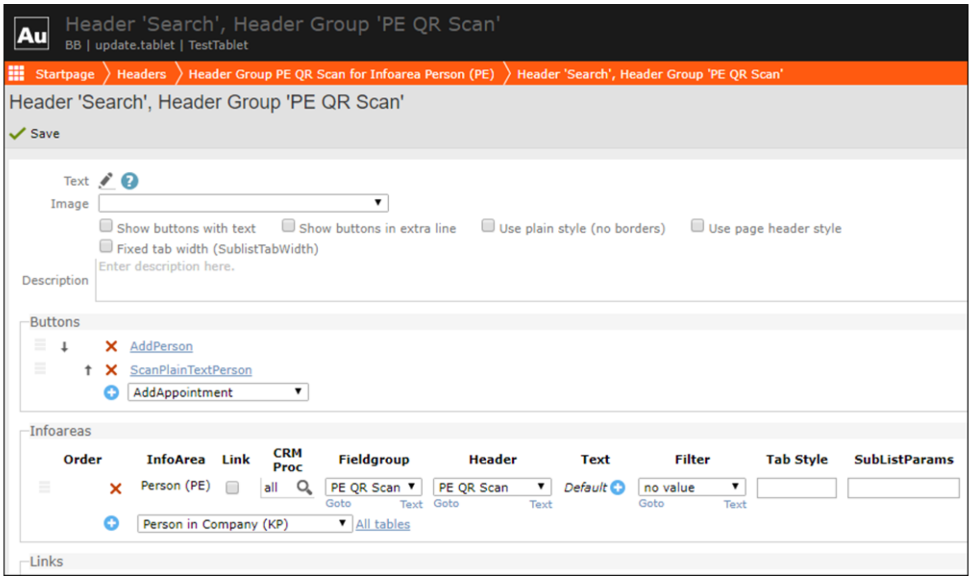
16.Add the button to the expand view header. Click Headers. The Headers page displays.



17. Open the PE QR Scan header group. The Header Group PE QR Scan for Infoarea Person (PE) page opens.

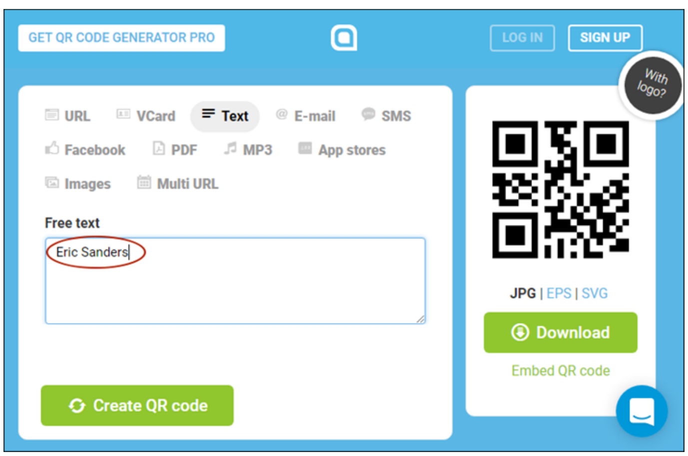


18. Select Search expand option. The Header Search, Header Group 'PE QR Scan' page opens.
19. Select ScanPlainTextPerson button from the drop-down list in the Buttons pane. Click **+** to add the button to the Header. See the screenshot below:



20.Click Save. The Configured user is now able to scan and edit Person details from a plain text QR Code.

21.The following screenshot shows the QR Code for a name in plain text.

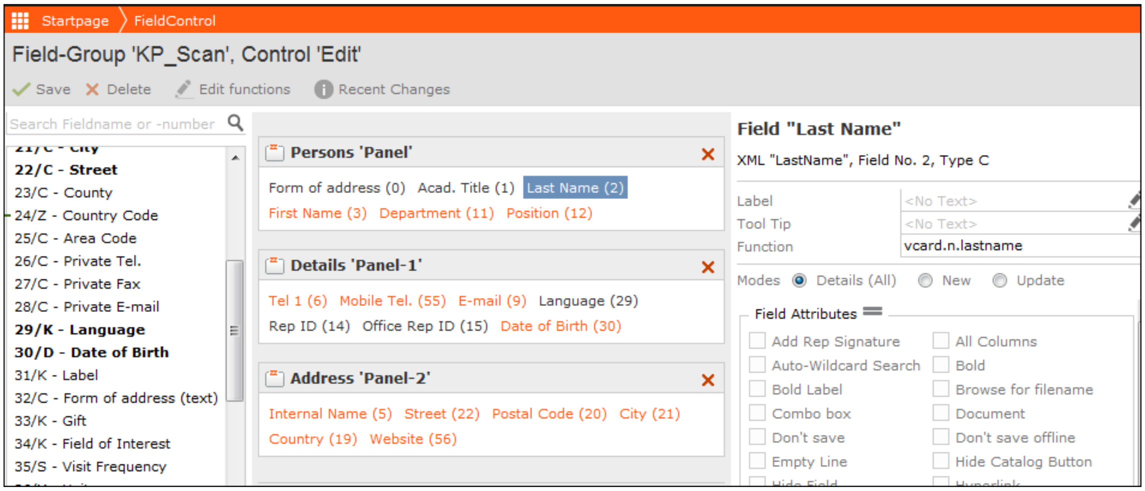


Configuring QR Code Scanning for Business Cards

The following example shows how to configure scanning business card to add a new a contact person (KP).

To configure the QR scan function to scan business cards to add a new a contact person (KP):

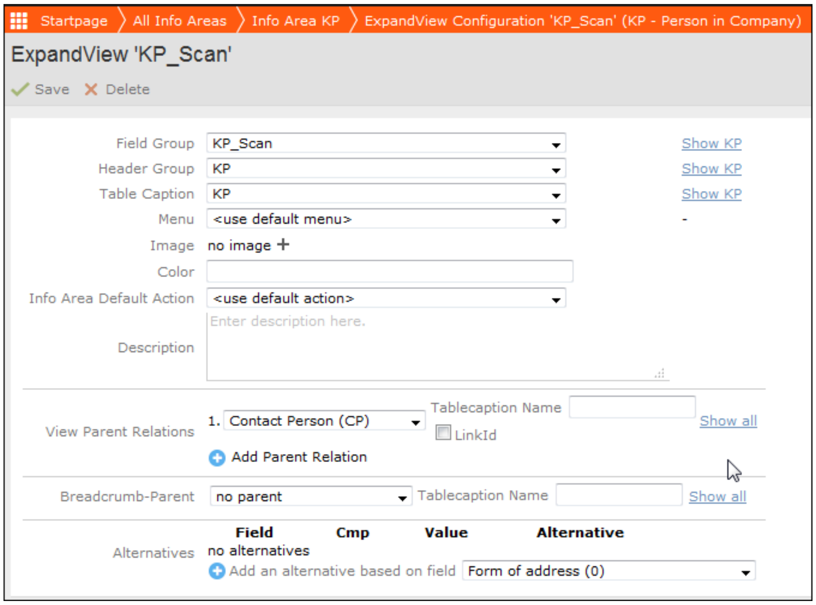
1. Click on **Infoareas** on the start page and then switch to the Person in Company (KP) info area.
2. Add a field group, e.g. KP scan:



3. Assign the following **Function** names to the following fields:

Field	Function
Last Name	vcard.n.lastname
First Name	vcard.n.firstname
Tel1	vcard.tel+work+voice
Mobile Tel	vcard.tel+cell
E-Mail	vcard.email+work
Birth Date	vcard.bday
Internal Name	vcard.org.0
Postal Code	vcard.adr.postalcode
Ort	vcard.adr.locality
Street	vcard.adr.street
Country	vcard.adr.country
Website	vcard.url
Department	vcard.org
Positition	vcard.org.0

4. Add an **Expand** configuration, e.g. `KP_Scan`, to the KP info area and and assign the previously added field group, e.g `KP_Scan`:



5. Switch to the **Buttons** area (from the start page) and add a button (e.g. `AddPerson_Scan`), assign the `NewView` action and specify the following input arguments:

Input Argument	Value
keyboardWithScannedSuggestions	CONTROLS
ConfigName	KP_Scan
LinkRecord	Record

6. Add the button to the desired Expand header.

Surveys/Questionnaires/Portfolios

Surveys can be carried out from an activity (MA) record. Questionnaires are assigned to e. g. an activity in Aurea CRM web or can be generated in the course of a marketing campaign.

The `update.tablet` configuration already contains all the required definitions to carry out a survey. The following chapters describe this default configuration.

If the assigned Survey (U1) record is only available on the server, it is automatically synchronized to the device.

Default Configuration

The following default configuration is included in the out-of-the-box version of CRM.pad.

- **Field Groups - List control**
  - **Questionnaire (F1)** (root record): Ensure that the following functions are assigned to the following fields of the default `F1Quest` field group :

Field	Function	Field Attribute
Questionnaire	QuestionnaireID	
Text	Label	
Type		Enable Hide Field

  - **Questions (F2):** The F2 info area stores the questions, a record must exist for each question. Ensure that the following functions are assigned in of the default F2Quest field group:

Field	Function	Field Attribute
Questionnaire (F1)	QuestionnaireID	
Question No.	QuestionNumber	
Question	Label	
Multiple Answers	Multiple	
Info Area Code	InfoAreaId	
Field Number	FieldId	
Question No. (F2#2)	FollowUpNumber	
New section	NewSection	
Mandatory Answer	Mandatory	
Answer Number (F3#2)	Default	
Read answer	Read	
Save answer	Save	
Do not display		Enable Hide Field

  - **Answers (F3)**: The F3 info area stores the possible answers. If a user can only choose from predefined answers, a F3 record must exist for each answer option. Ensure that the following functions are assigned in the default F3Quest field group:

Field	Function
Questionnaire (F1)	QuestionnaireID
Question No. (F2)	QuestionNumber
Answer No.	AnswerNumber

Field	Function
Answer	Label
Question No. (F2#1)	FollowUpNumber

- **Field Groups - Edit control**
  - **Activity (MA):** Ensure that the following fields (and function) exist in the Edit control of the MA field group:

Field	Function
Questionnaire	Related:QuestionnaireID
Survey on	

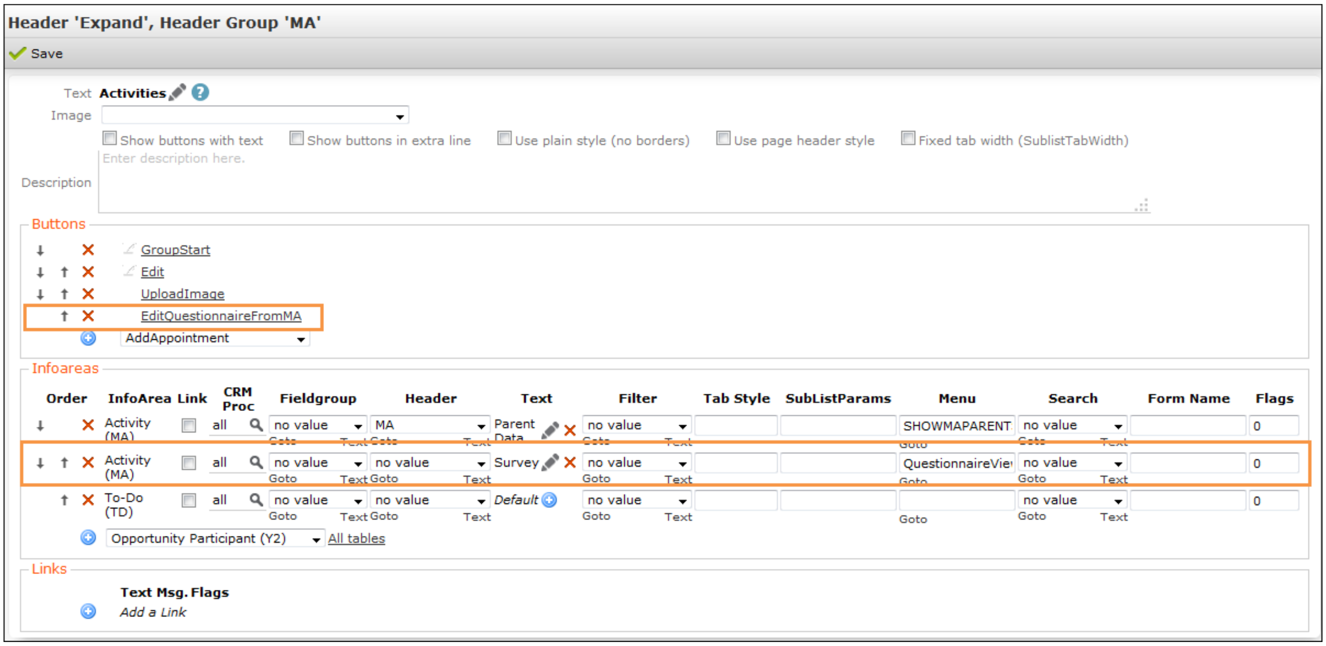
- **Search&List configurations**
  - **F1Quest:** References the field group with the same name in the F1 info area.
  - **F2Quest:** References the field group with the same name in the F2 info area.
  - **F3Quest:** References the field group with the same name in the F3 info area.
- **Context Menu Action**

  The `QuestionnaireConfiguration` context menu action has the `Configuration :Questionnaire` action template assigned. The following input arguments are assigned by default and reference the above mentioned Search&List configurations:

Field	Function
QuestionnaireSL	F1Quest
QuestionsSL	F2Quest
QuestionsAnswersSL	F3Quest

- **Header Group**

  Ensure that the following items exist in the MA Expand header:



- The `EditQuestionnaireFromMA` button allows the user to answer (i.e. edit) the questionnaire.
- The item in the **Infoareas** area displays the questionnaire in an additional tab in the activity's Details view in read-only mode, if the name of the context menu action that calls the questionnaire (default: `QuestionnaireView`) is given in the **Menu** column.

Adding a Finalize Survey Button

You can add a **Finalize Survey** button to the questionnaire. When the user taps on this button, the questionnaire cannot be edited anymore. You can define filters that define the "finalized" state of the survey and assign them to the button.

To add the **Finalize Survey** button:

1. In the default configuration, the `ConfirmSurvey` button has the `SaveAndConfirm` action template assigned, >> SaveAndConfirm on page 126.
2. Ensure that the following input arguments are defined:
   - `RecordId`: Must be set to `Record`.
   - `ConfirmFilter`: If desired, you can change the default value (`FINALIZE`) to another filter, e. g. `U1.Confirmed`.
   - `BaseRecordConfirmFilter`: Defines the conditions which questionnaires are displayed read-only.

- Ensure that the following input arguments are given in the `QuestionnaireEditView` action call (assigned to the `EditQuestionnaireFromMA` button):
  - `ConfirmButtonName`: The button calling the `SaveAndConfirm` action, in the default configuration `ConfirmSurvey`.
  - `ConfirmedFilterName`: The given filter defines the finalized state of a survey. Only records that do **NOT** meet the conditions can be edited by the user. Records meeting the conditions are read-only (the finalized button is not displayed anymore). This argument is set to `U1.Confirmed` in the default configuration.
  - `ReadOnly`: If you set this argument to `true`, the questionnaire is always read-only. The user cannot edit it.

Portfolios

If you want to display a portfolio on the start page (Overview), set the `Portfolio` parameter in the `QuestionnaireEditView` action template (assigned to the `EditQuestionnaireFromMA` button) to `true`.

Note: Portfolios can only be defined in Aurea CRM web and can only be displayed if one exists.

Quick Add

The quick add allows the user to enter several item records belonging to one header record in List view.

For example:

- Order items (UP) are added to an order (AU) record.
- Delivery dates (LP) are added to order item (UP) records.
- Point Of Sale (root records are stored in the PS info area, the child records in the PP info area).

This chapter describes how to configure the quick add in CRM.pad. You can find further information about the quick add in the SerialEntry topic in the *Aurea CRM web Administrator Guide* and in the Quick Add chapter in the Aurea CRM win Administrator Guide.

The Quick Add Online and Offline

Learn about the quick add stores.

The quick add stores each item when the use switches to another item. If a server connection is available, the records are stored online on the server, otherwise they are stored locally on the device and must be synchronized later on.

In online mode when the quick add is closed, all remaining changes are transferred to the server, the server's business logic is applied (including custom triggers) and all the modified records (including the root record) are reloaded from the server after being saved. For example, pricing calculations carried out by triggers for an order record are immediately displayed in CRM.pad.

Some functions are, however, implemented in CRM.pad to ensure that they are also available in offline mode, e.g.:

- Adding up totals in destination child records (e.g. the **Quantity** column)
- Creating destination parent records for the product detailing.
- Pricing for orders
- Quotas for samples

When configuring the quick add, ensure that the business logic applied on the server does not interfere with the client's business logic but still enables the client to carry out subsequent processes. Example: Prices applied offline by the pricing must not be transferred to the server (and, therefore, cannot override the business logic applied on the server). The order summery is generated on the client.

Basic Quick Add Information

You can use the quick add to add records to or edit records from a destination info area.

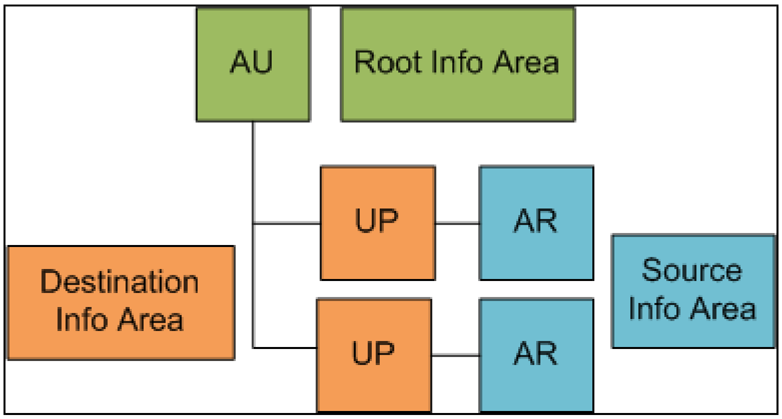
These records are linked to a root record and to a set of source records from which you selected the desired one.

- Root record/info area: The root record's info area. The quick add is started from this record (e. g. AU)
- Source record/info area: The records displayed in the rows of the quick add are read from the source info area (e.g. AR records for an order)
- Destination record/info area: The records that are added as items to the order, i.e. the rows where you added data (e.g. UP records for an order). Each destination record is linked to the root record and to the corresponding source record.

If a List control containing sort fields also exists in this field group, its sort fields are used to sort existing records (e.g. the records in the Overview).

Example: Orders in the OTC/LSI versions

You can add several order items (UP - destination records) to an order (AU - root record) that are linked to the articles (AR - source records) that were selected from the available set of Master Item (AR) records:



Defining a Basic Quick Add

Learn how to define a basic quick add.

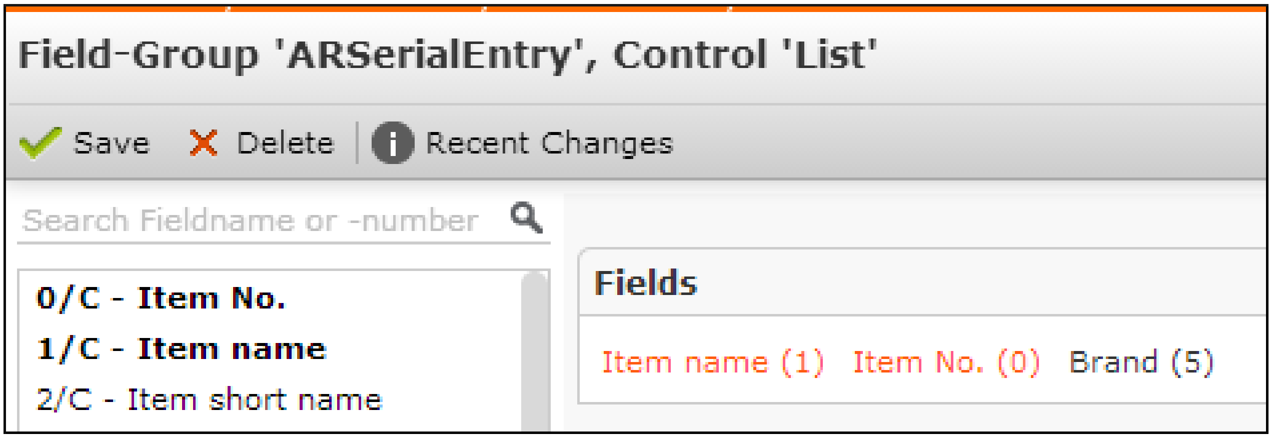
To define a basic quick add:

1. You need to assign the `SerialEntry` action to either a button or context menu action, see SerialEntry on page 127.
2. You need to define at least the following input arguments for the `SerialEntry` action call:
   - `EditType`
   - `RecordId`
   - `SourceConfigName`
   - `DestinationConfigName`

Source Field Group

Define the source field group that is referenced in the `SourceConfigName` input argument of the `SerialEntry` action call.

Example: The List control of the default `ARSerialEntry` field group:



The fields are displayed in the quick add on the left of each row of the list.

You can define that these values are handed over to the destination record by assigning the following **Function** names:

Field	Function
Item Name	`CopyItemName`
Item No.	`CopyItemNumber`

All functions starting with `Copy` are copied to a field in the destination field group that needs the same function assigned.

Destination Field Group

The default configuration contains the **UPSerialEntry** field group that defines the destination records. Its Edit control defines the fields that users can enter when they switch to the quick add's Edit mode and the given data is stored in this info area.

Ensure that the following **Functions** are assigned to the following fields:

Field	Function
Quantity	`Quantity`
Value	`EndPrice`
Item No.	`CopyItemNumber`

Field	Function
Item Description	`CopyItemName`
End Price	`UnitPrice`
Currency	`Currency`
Discount	`Discount`
Discount vol. in kind	`FreeGoods`

Available Functions in the Standard Quick Add

The following **Functions** can be assigned to fields in field groups:

Function	Description
`CopyItemNumber` / `CopyItemName`	Data from the source field group is copied to the destination field group from and to the fields that have these functions assigned. **Note:** If you do not use the copy functions, the Item Number field needs to be entered manually for the UP records, as the AR link does not work correctly in the order entry quick add.
`FixedValue`	Allows to specify a fixed value, e.g. `FixedValue :3` or `FixedValue:"<string>"`.
`StepSize`	Defines that the specified step size for the items quantity is applied.
`ListEditField`	Defines that a Boolean value can be edited in lists. If not set (for each field), they can only be edited in the Details view.
`Quantity`	Defines the quantity field needed for pricing, step size and more.
`FreeGoods`	Defines the field for the free goods (needed for pricing).
`EndPrice`	`UnitPrice` multiplied by `Quantity`.
`UnitPrice`	Base price of an item.
`NetPrice`	`EndPrice` minus discount.

Function	Description
Currency	Defines the currency field.
ServerApproved	If this field is **not** set to ServerApprove :true, the fields with the UnitPrice + NetPrice **Functions** assigned are displayed in italic.

The fields that have the Quantity, Discount, and FreeGoods functions assigned are the fields where the user can enter data.

The EndPrice, UnitPrice, NetPrice, Currency, ServerApproved,and StepSize functions define functions that are specific to the order's quick add.

Functions required for entering data in a PDF file:

Function	Description
SourceImage	Defines that an image from the source's root info area is displayed.
CopyItemNumber	This field is used to match-up the records from the PDF with the database.

Functions required for the quota calculation:

Function	Description
Quota	Optional. Defines the quota for a specific time period for an item and overrides the default settings specified for the Configuration :Quota action call (if the field exists and its value is < 0).
Items	Defines the number of items of the current time period that are still available
EndDate	Defines the time period's end.
Year	Defines the time period as calendar year (if the field with EndDate assigned does not contain a valid date).
StartDate	Optional. If the CalendarYearPeriods was defined in the Configuration :Quota action call, this field is used to calculate the end of the previous time period.
Limited	If a field has this function assigned and its value is either empty or 0, samples can be issued without limit. If its value is < 0, the maximum number of issued samples is limited to the field's value.

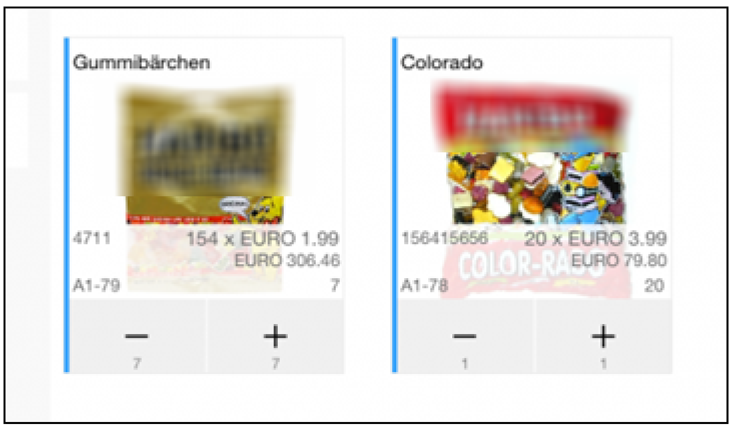
Field Attributes for the Quick Add

The following field attributes apply to the quick add:

- `Hide Field`: If set, the field is not displayed in the application.
- `Don't save`: The field is only stored locally (i.e. it is not transferred to the server). For example: If no server connection is available, you may still want to display the result of a pricing calculation. If transferred to the server, this result can influence the business logic applied on the server.
- `Don't save offline`: The field is not stored at all, neither on- nor offline.
- `Empty line`: Defines whether a destination record can be deleted. If at least one field has the `Empty line` attribute assigned **and** all fields that have the `Empty line` attribute assigned are empty (or 0 - if it is a numeric field), the record is deleted.
- `Image`: If set (in the source's root info area), the field's content is displayed as an image in the quick add.

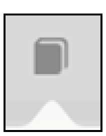
Adding Images to the Quick Add

If a field in the source info area's List control has the `Image` field attribute set or the `SourceImage` **Function** assigned, the field's content is displayed as an image.



Adding PDF Files to the Quick Add

You can also use PDF files to select items from the quick add. If defined accordingly, the items can be selected with the following icon:



To define that PDF files are "clickable":

1. Open the desired PDF file in Acrobat Reader.
2. Switch to the `Comments` tab.
3. Select the "clickable" area.

Note: Always select a rectangle as "clickable" area. You can use different shapes, but only rectangles within these shapes can be defined for input.

4. Enter one of the following commands in the comment:

Command	Description

`<ItemNumber> or`
`{"ItemNumber":<ItemNumber>} or`
`{"ItemNumber":<ItemNumber>,"StepMode":"imc"}`
	Opens the Details view and increases the quantity by the specified step size.

`{"ItemNumber":<ItemNumber>,"StepMode":"noop"}`	Opens the Details view without changing the quantity.
`{"ItemNumber":<ItemNumber>,"StepMode":"inc", "StepSize":"10"}`	Opens the Details view and increases the quantity by 10.
`{"ItemNumber":<ItemNumber>,"StepMode":"set", "StepSize":"10"}`	Opens the Details view and sets the quantity to 10.
`{"ItemNumber":<ItemNumber>,"StepMode":"delete"}`	Removes the item from the shopping cart.
`{"ItemNumber":<ItemNumber>, "pdfHIColor":"#00FF0045"}`	The barcode that is "clickable" is highlighted with the color given in hex format.

Replace `<ItemNumber>` by the value that you want to hand over to the field that has the CopyItemNumber function assigned (usually the item number).

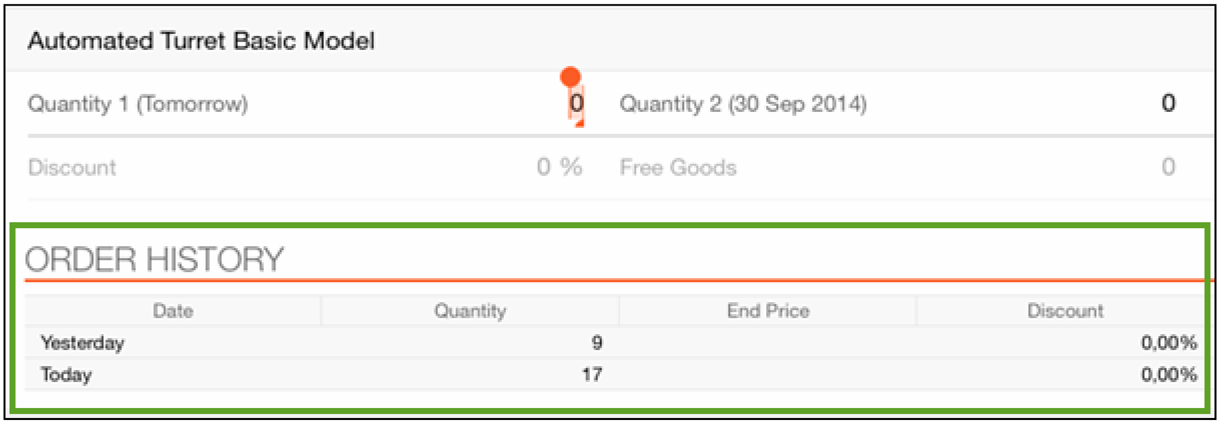
Example: `{"ItemNumber": "11203","StepMode":"inc"}`

Note: Always enter a string value as item number. Therefore, the above example does not work if you enter `{"ItemNumber":11203,"StepMode":"inc"}`, even if an integer field is referenced.

5. Enter the name of a Search&List configuration (defined in the D1 info area) in the `ProductCatalogSource` input argument of the `SerialEntry` action call. This Search&List configuration defines the icon of the button that is displayed in the quick add and the filter applied to limit the number of available documents.

Info Panel

You can define that additional information is displayed for the current record in the quick add:



Define the `InfoPanelDefinition` input argument in the `SerialEntry` action call if you want to add an info panel, , see SerialEntry on page 127.

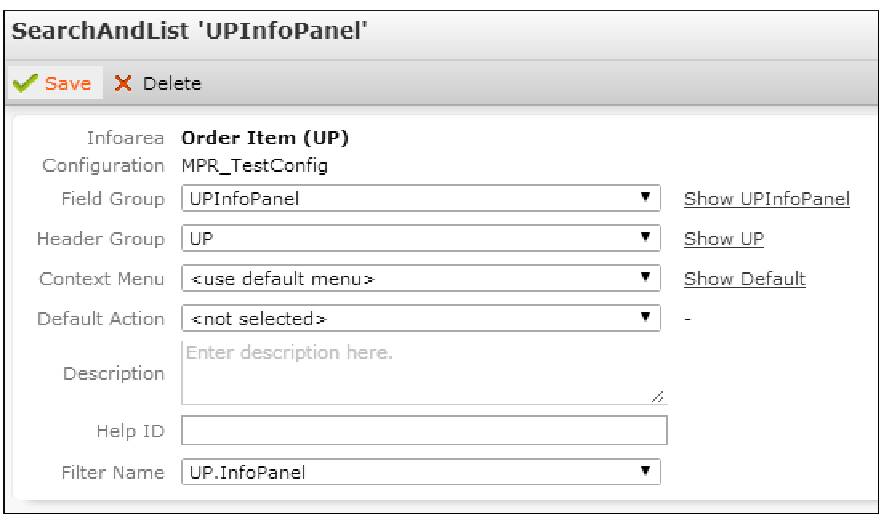
Info Panel Search&List Configuration

You can define a Search&List configuration that determines

- the filter that limits the number of displayed records.
- the info panel's display text.
- the info panel's header.

Enter this Search&List configuration's name in the `SourceCopyFieldGroup` input argument of the `SerialEntry` action call.

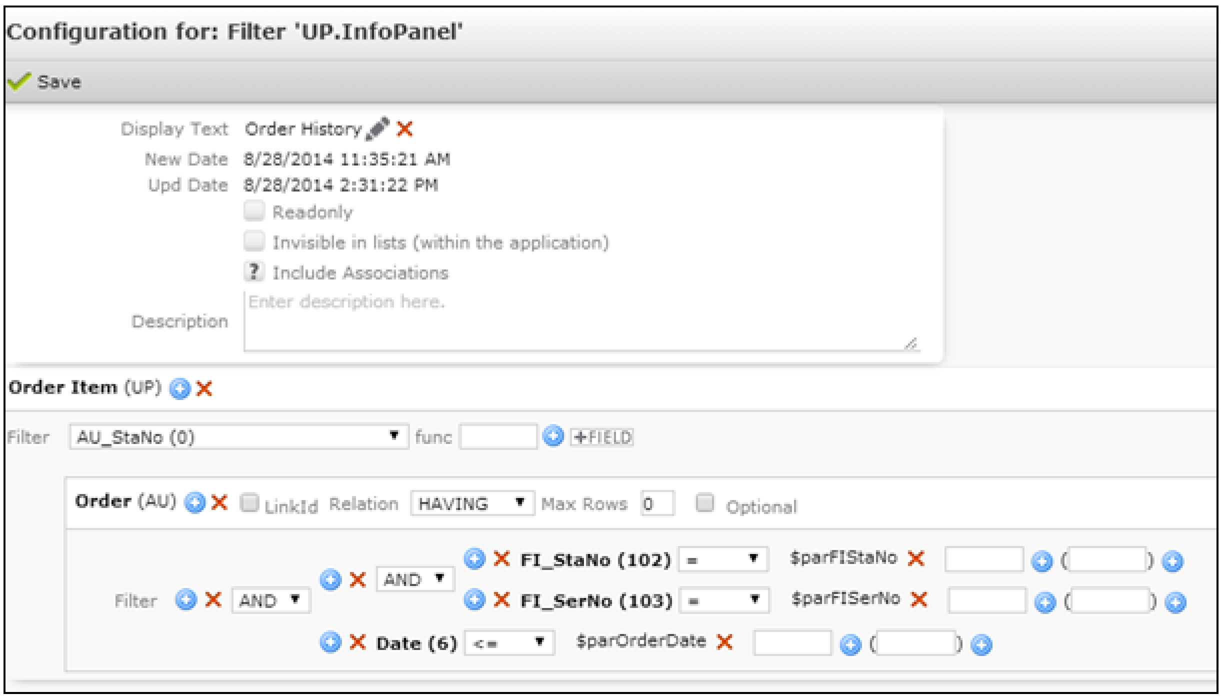
Example:



This Search&List configuration references the following configuration units:

- List Control of the `UPInfoPanel` field group:

- `UP.InfoPanel` filter: The following example defines that order items are only listed for the current company if the order has the current or an earlier date assigned.



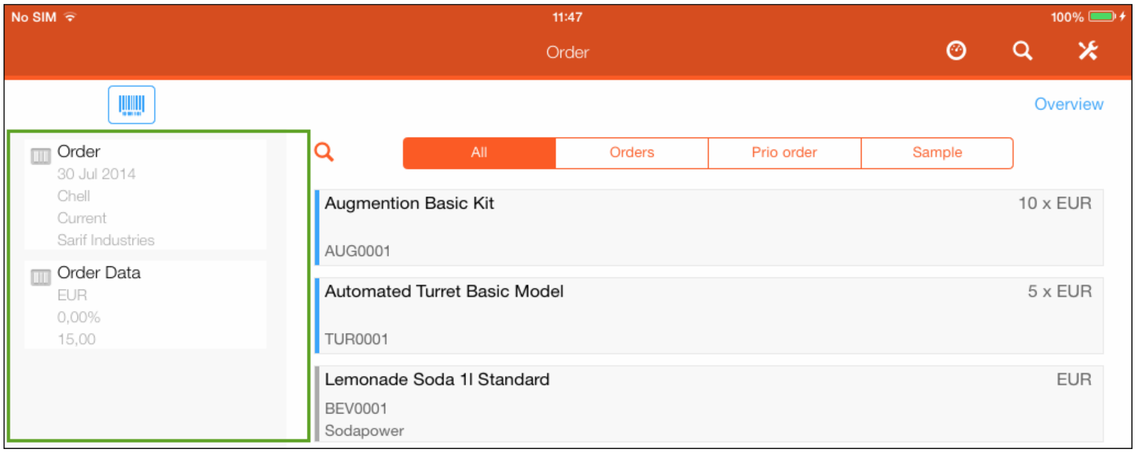
The filter's **Display Text** (in this example "Order History") is displayed in CRM.pad as the info panel's header.

The filter's parameters must exist in the List control of the referenced field group (specified in the `SourceCopyFieldGroup` input argument of the `SerialEntry` action call) and must have the required **Functions** assigned, e. g. in the `AUCopyFields`' List control:

Field	Function
FI-StatNo (0)	FIStaNo
FI-SerNo (1)	FISerNo
Date (6)	OrderDate

Quick Add Overview

In landscape mode, the quick add overview is also be displayed.



You can define whether it is displayed on the left or right side, see **Show Quick Add Overview on the left** option in the iPad Settings, see Overruling Default Log Settings in the iPad Settings.

The fields displayed are defined in Details control of the field group that is referenced in the Search&List configuration specified in the `SerialEntry` action call's `DestinationRootConfig` input argument. If this field group does not exist, the Details control of the info area with the same name than the info area is applied.

Only the fields (without field labels) are displayed, empty fields are ignored. The displayed icon is the defined info area icon (of the first field). You can override the defined image with the `ParenInfoImage` control attribute, see ParentInfoImage.

Filtering Items

You can define filters that limit the number of displayed items by specifying the available filters in the following input arguments in the `SerialEntry` action call, see SerialEntry on page 127:

- `Filter1` to `Filter6` (only for fixed filters without parameters).
- `PositionFilter1` to `PositionFilter4`: Filters requiring parameter input can be given.
- `PositionAdditionalFilters`: Max. four filters can be entered, separated by semicolon. Filters requiring parameter input can be given.
- `HierarchicalPositionFilter`: The provided filters are hierarchically combined from the left to the right. If a filter is set by the user, all filters to the right are reset.

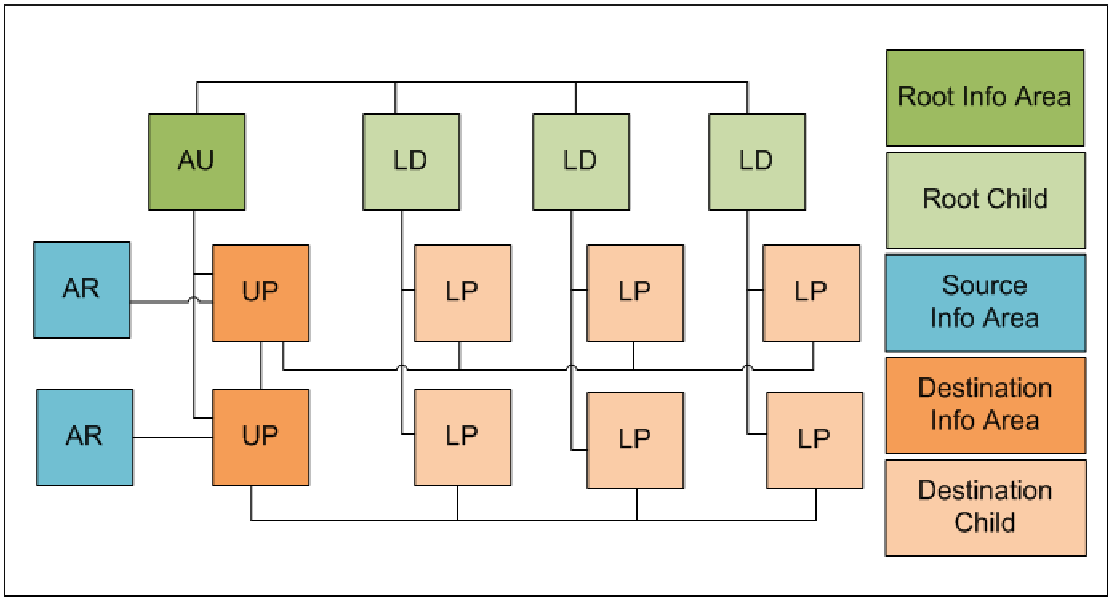
If one of the `PositionFilter/PositionAdditionalFilters` arguments is specified, filters entered in `Filter1` to `Filter6` are available in a pop-up.

Ensure that fields used as parameter criteria in filters exist in the List control of the specified `SourceConfigName` input argument. They must also have the required functions assigned. You can define the `Hidden` field attribute to prevent them from being displayed in the application.

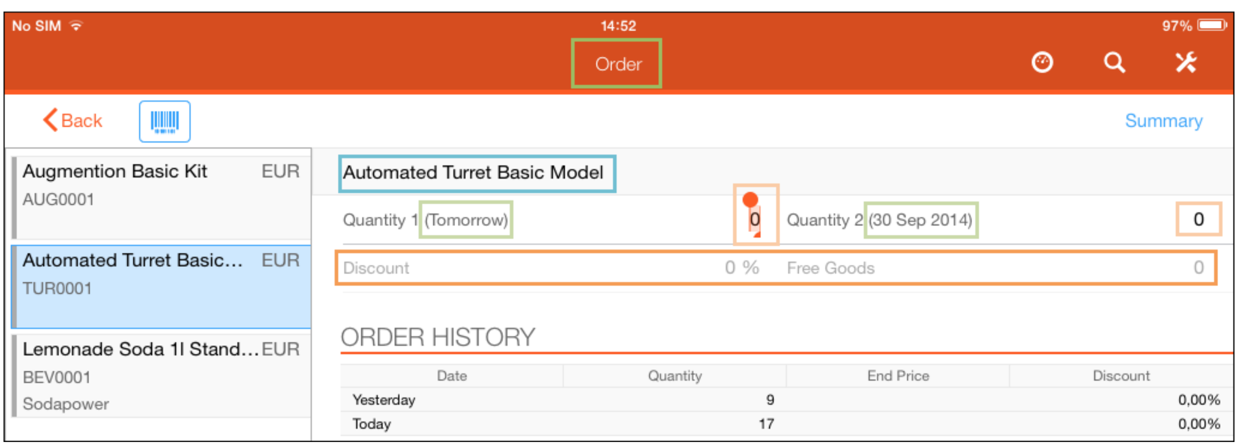
Destination Child Records

In addition to create a record in the defined destination info area, you can also create several child records for this destination record (i.e. 1:n links).

Example: An order in CRM.pad LSI can have several delivery dates (LD, 1:n links to AU). When the user enters data in the quick add, a delivery date item (LP, 1:n links to LD and UP) is created.



The following example shows how the data is displayed:

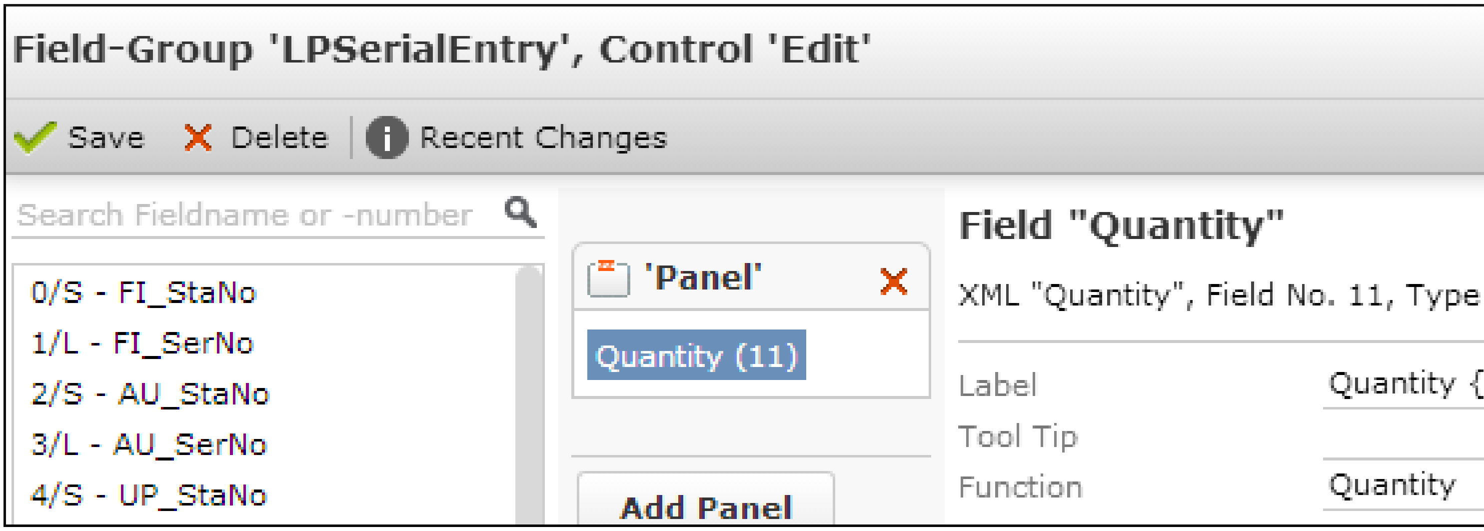


Define the following input arguments in the `SerialEntry` action call to enable this feature, see SerialEntry on page 127:

- `SourceChildConfigName`: Name of a Search&List configuration referencing the root's record's child info area.
- `DestinationChildConfigName`: Name of the field group that contains the Edit control used by the quick add.

Example: Destination Child Record Field Group

he following example shows a destination child field group:

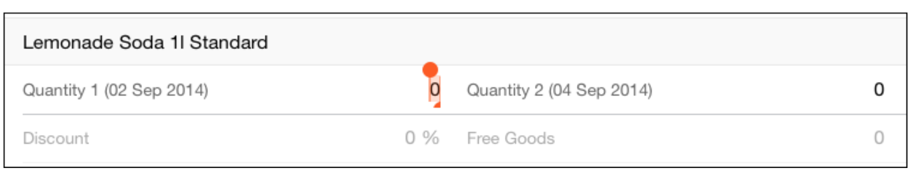


Add the Quantity field and assign the `Quantity` **Function**. This function hands the total of all destination child values over to the Quantity field in the destination info area.

Define the **Label** field by entering `Quantity {1} {2}`:

- `{1}` numbers all the fields of the child's field group controls.
- `{2}` is replaced with the value of the (first) field from the root's child record.

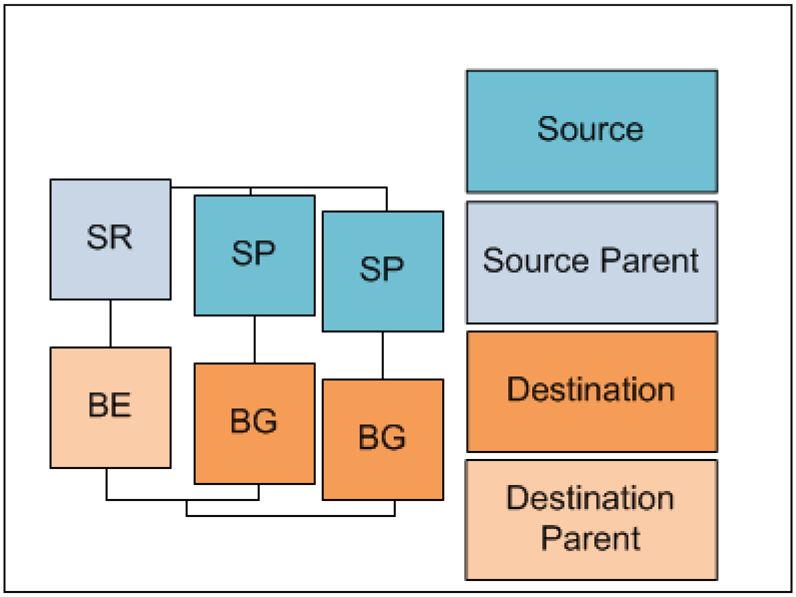
The following image shows two (numbered) Quantity fields. The delivery date displayed in brackets (as defined in the field group's label in the image above) shows the delivery dates. This data is read from the List control of the field group specified in the `SourceChildConfigName` input argument of the `SerialEntry` action call (in this example, the delivery dates):



Destination Parent Records

Source parent records are used for product detailing definition (SR). A product detailing record consists of several product detailing items (SP). Each SP record references an item master (AR) record.

Example: Several product detailing records were assigned to a rep. When the rep opens the quick add, not only a product detailing item (BG) record is created per item, but also a product detailing (BE) record.



Define the following input arguments in the `SerialEntry` action call, see SerialEntry on page 127:

- `SourceParentInfoAreaId`
- `DestinationParentConfigName`

Adding POS Items Automatically

You can define that POS items are automatically added specifying the `Options` input argument for the `SerialEntry` action template by entering the following JSON string:

`{"autoCreatePositions" :true }`

The following additional options are available:

- `"autoCreateDoNotAsk":true`: No message box is displayed; the records are created without delay
- `" autoCreateMessage :"mytexts:0"`: Message box text (you can use the `%d` parameter to display the number of created records).
- `"autoCreateMessageHeadline":"mytexts:1"`: Message box headline

The format for referencing texts is `<clientTextgroupsuffix >: <Text Index>`. The text group must be named in CRM.designer as follows: `smartbook:<clientTextgroup-suffix>`

Example:

```
{"autoCreatePositions" :true , "autoCreateMessage":"mytexts:0", "autoCre-
ateMessageHeadline":"mytexts:1"}
```

Issuing Samples

Learn how to configure sample issuance in CRM.pad.

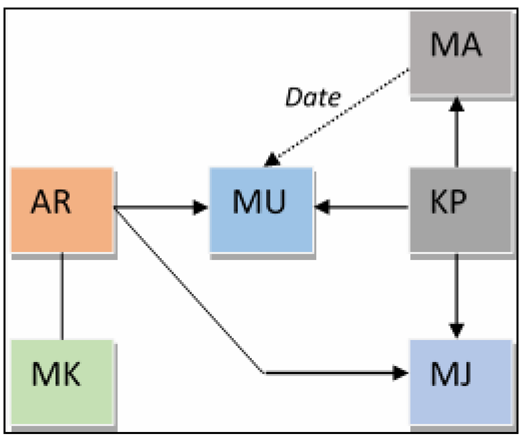
When samples are issued, quota calculations are required.

The rep is only allowed to hand out a limited number of a certain drug to clients (e. g. a physician). This maximum number is displayed in the sample quick add.

Note: The sample quick add is **not** thoroughly described in this chapter. Detailed information is available in the >> Samples chapter in the OTC Business Logic Manual.

Involved Info Areas

The following info areas are taken into account for calculating the quota:

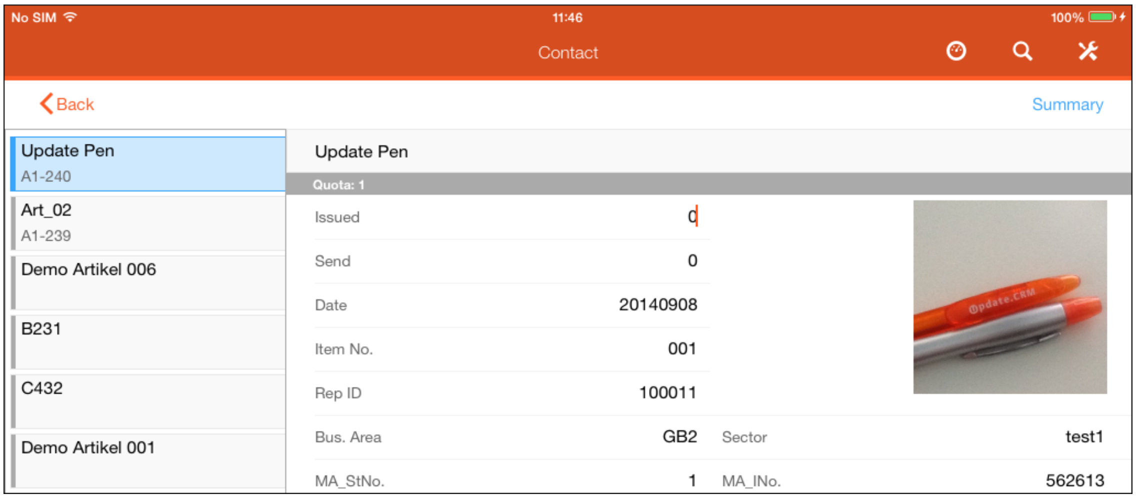


- Item Master (AR): Source record configuration for Samples.
- Sample (MU): Destination record configuration. The items where the user has entered data.
- Person in Company (KP) or Company (FI): Root record configuration. The record for which the MU records are created.
- Activity (MA): Samples are usually issued when a rep visits a client, therefore, this process is started from a MA record (that provides the issue date).
- Sample Code (MK): Defines for each item master record how many samples can be issued for the defined time span in total. These records define whether samples can be issued at all.
- Samples (MJ): For each item/person/sample code combination an MJ record is created (i.e. each MJ records has tree parent records). This record defines the rest of the time span and how many samples have been issued so far.

These records are only available offline after synchronization as they are only processed by the server. If a rep works offline, the created MJ records are not synchronized. CRM.pad calculates the quota locally by subtracting the issued samples from the server quote. The difference (i.e. the samples issued) is then synchronized to the server allowing to calculate the new quota.

Display in CRM.pad

The quota calculated by the client is displayed as follows:

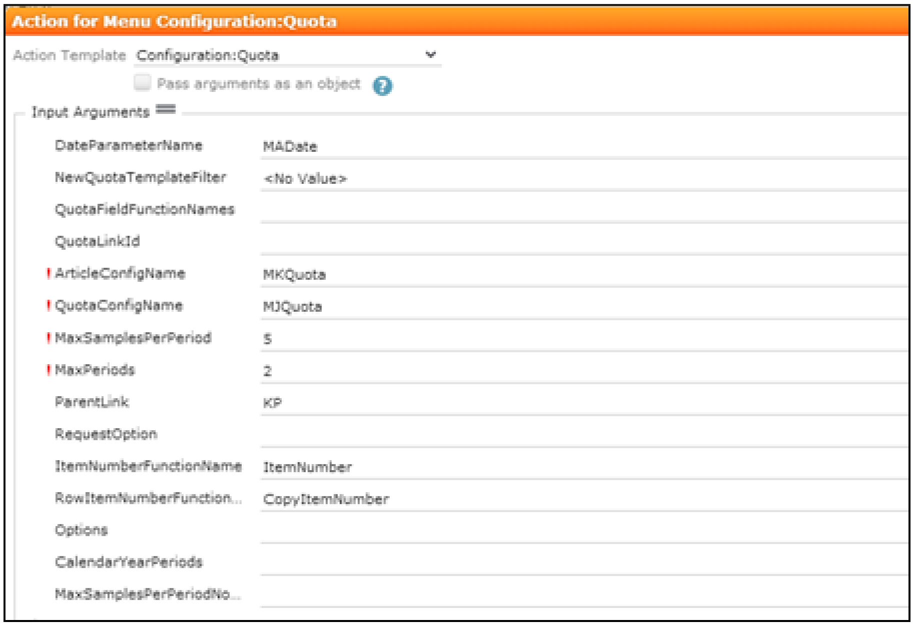


If the user wants to issue more samples as he may issue (quota), this line is marked in red. You can define whether a rep can hand out more samples than the available quota or whether the allowed number is automatically set to the number of the remaining quota, see `Options` input argument of the `Configuration :Quota` action call, see Configuration:Quota on page 88.

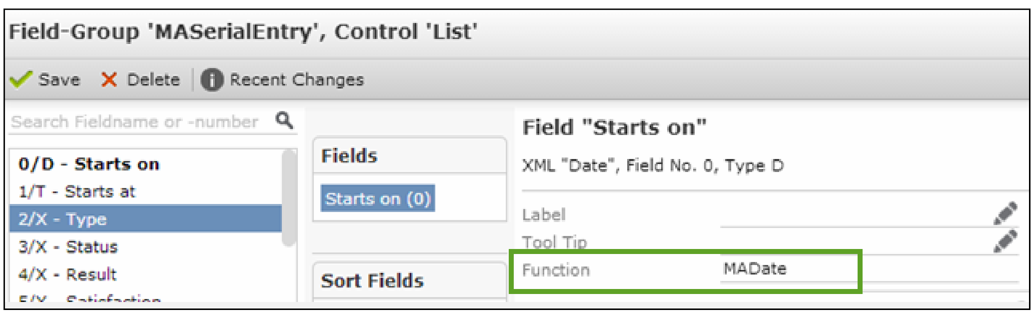
Configuring the Sample Quick Add

To enable the quota configuration, enter `Configuration :Quota` in the `QuotaConfiguration` input argument of the `SerialEntry` action call, see SerialEntry on page 127. This input argument references the `Configuration :Quota` context menu action that has the `Configuration:Quota` action call assigned, see Configuration:Quota on page 88.

The following screenshot shows the default configuration of this action call:



The field group (`MASerialEntry` in the above example) given in the `SourceCopyField-Group` input argument of the `SerialEntry` action call must contain a date field that has a **Function** assigned, e.g. `MADate`.



Enter this **Function** name in the `DateParameterName` input argument of the `Config-uration :Quota` action call. The date is then handed over to the sample record.

Configuring the Item Master

You must define the Item Master fields that are required for the Sample Code (MK) records.

Assign the following **Function** names in the List control of the field group specified in the `ArticleConfigName` input argument of the `Configuration :Quota` action call:

Field	Function	Description
Item No.(0)	`ItemNumber`	Mandatory, specifies the field holding the item number.
Units After Launch (7)	`Quota`	Specifies the field that defines the quota per time span for an item. Enter this function if you want to override the default quota setting. Only applies when the field content is > 0.

Configuring the Quota

You must define the fields in the Sample info area that are required for the Sample Code (MK) records.

The `QuotaConfigName` input argument references a Search&List configuration. This Search&List references a field group (e.g. `MKQuota`) that needs the following **Functions** assigned its List control:

Field	Function	Description
Item No.(0)	`ItemNumber`	Mandatory, specifies the field holding the item number.
Quantity issued	`Items`	Mandatory, specifies the field holding the number of samples issued.
End of Period (17)	`EndDate`	Mandatory, specifies the end of the time period.
Year	`Year`	Optional, defines the calendar year as the time period. In this case, the the End of Period field must not contain a valid date.
Start of Period	`StartDate`	Optional. If the `CalendarYearPeriods` input argument was set to `true`, this function is needed to calculate the past period's end date.
Allocated (3)	`Limited`	Defines that no quota is applied.

Pricing

You can define that Aurea CRM's pricing business logic is also available for the order quick add, not only in online but also in offline mode.

The pricing business logic needs:

1. A price list containing the list prices for all articles defined in Aurea CRM win.
2. An optional pricing condition information (defined in conditions)

3. An optional pricing condition information for special offers (defined in the Special Offers or Special Offer Scales)
4. An optional pricing condition information for the linked account (defined in the Account Conditions)
5. An optional bundle pricing condition information for special offers (defined in the Special Offer Bundles or Special Offer Bundle Scales)

Offline pricing works as follows:

1. Does a special offer pricing condition exist?
2. If (1) is yes, does a bundle pricing exist?
3. If (1) is no, does an account pricing condition exist?
4. If (1) and (3) are no, does a standard pricing condition exist?
5. If (1) or (3) or (4) is yes, apply the conditions.
6. If the condition does not return a fixed price, use the price from the price list.

A condition record (KD, AA, FK or FG) is valid, if:

- It matches with one of the fields listed in the `FunctionNameApplyOrder` parameter (and the condition is not empty).
- All fields with function names listed before that name are blank in the condition record.
- No "more specific" condition records exist.

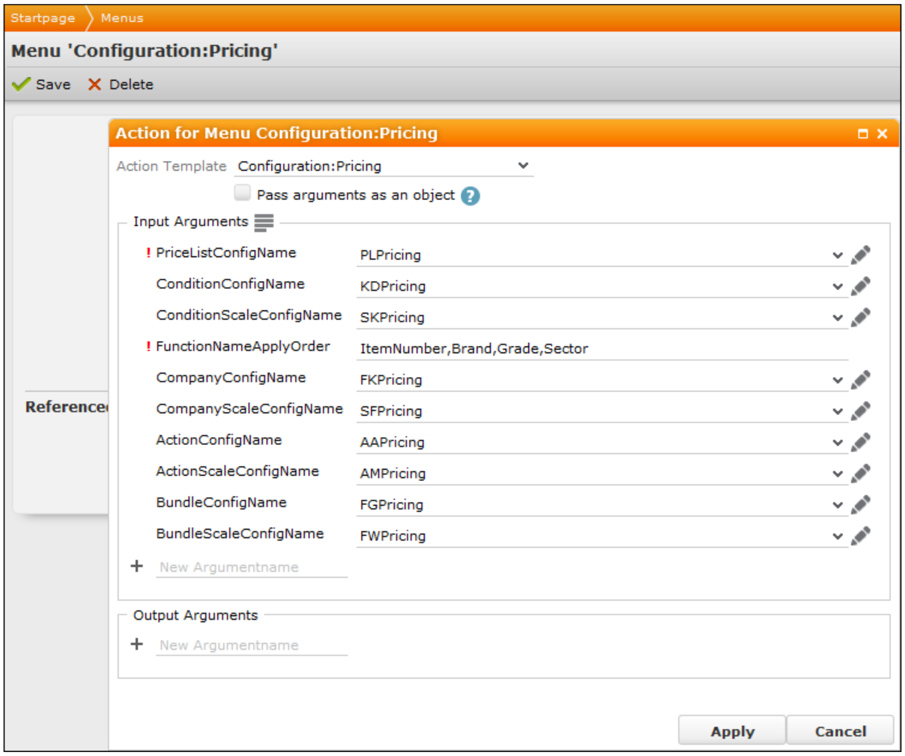
Configuring the Pricing

The pricing is enabled when you enter `Configuration :Pricing` in the `PricingConfiguration` input argument of the `SerialEntry` action call

For further details, see SerialEntry on page 127.

This input arguments references the `Configuration :Pricing` context menu action that has the `Configuration:Pricing` action call assigned, see Configuration:Quota on page 88.

The following picture shows the default configuration of this action call:



Note: If you use Serial Entry, it is necessary to add a PriceList (PR) to article (AR) for the pricing to work correctly.

- **Before adding a PriceList (PR) to article (AR)**: When using the Function UnitPrice on AR and UP, the determined prices from AR are rounded in UP. (Example - Price in AR = 12,95 €, price in UP = 12,00 €).
- **After adding a PriceList (PR) to article (AR)**: When using the Function UnitPrice on AR and UP, the determined prices from AR are not rounded in UP. (Example - Price in AR = 12,95 €, price in UP = 12,95 €).

Note: If you use Serial Entry, it is necessary to add a PriceList (PR) to article (AR) for the pricing to work correctly.

- Before adding a PriceList (PR) to article (AR): When using the Function UnitPrice on AR and UP, the determined prices from AR are rounded in UP. (Example - Price in AR = 12,95 €, price in UP = 12,00 €).
- After adding a PriceList (PR) to article (AR): When using the Function UnitPrice on AR and UP, the determined prices from AR are not rounded in UP. (Example - Price in AR = 12,95 €, price in UP = 12,95 €).

Configuring the Price List

You can define that the price lists defined in Aurea CRM web are also available in CRM.pad.

You can allocate items to the list prices by assigning the `ItemNumber` function.

Note: Allocating price lists by the item record itself is also available but only for compatibility purposes. Allocate price lists only by `ItemNumber` function.

To configure the price list:

1. Specify the `PriceListConfigName` input argument in the `SerialEntry` action call. This enables
   - the currently valid price list records (based on the `PL.OrderFilter`).
   - the price and article properties for each article.
2. The default `PLPricing` Search&List configuration references the `PLPricing` field group and applies the `PL.OrderFilter`.
3. If you want to add brand or grade as filter criteria (in addition to `ItemNumber`), switch to the `PLPricing` field group and enter the following functions for the listed fields:

Field	Function
End Price	UnitPrice
Item No.	ItemNumber
Brand	Brand
Grade	Grade

4. If you define PriceList in `Serial Entry`, the price should be picked up from that price list while adding a Item in `Serial Entry`. For this to work, make the following configuration changes:
   - set `MatchWithParentPricing` = true, parameter for Pricing Configuration
   - Set the Function name as "PriceList" for Pricing field in SerialEntry FieldGroup control
   - Use the same function name for the field in Pricing Info Area

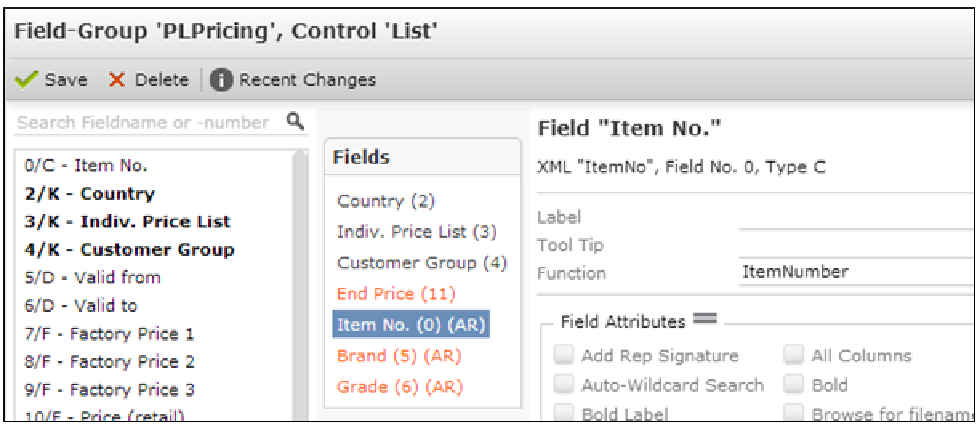
Assigning List Prices to Items

Learn how to assign items to the list prices.

You can assign items to the list prices either in by the item record itself or by assigning the `ItemNumber` function:

- By default, the item record is used.
- If the `ItemNumberFunctionName` was specified in the `SerialEntry` action call, the `ItemNumber` function is applied.
- If the `Pricing.AutoItemNumber` was set and the `ItemNumber` or `CopyItemNumber` functions were assigned in the source field group, the `ItemNumber` function is applied.

Note: Use the `ItemNumber` function, as the allocation by the item record itself is only available for compatibility purposes.



Allocation by ItemNumber function:

Ensure that a field in the `PLPricing`'s List control has the `ItemNumber` **Function** assigned (as shown in the screenshot above, this can also be a link field from the AR info area).

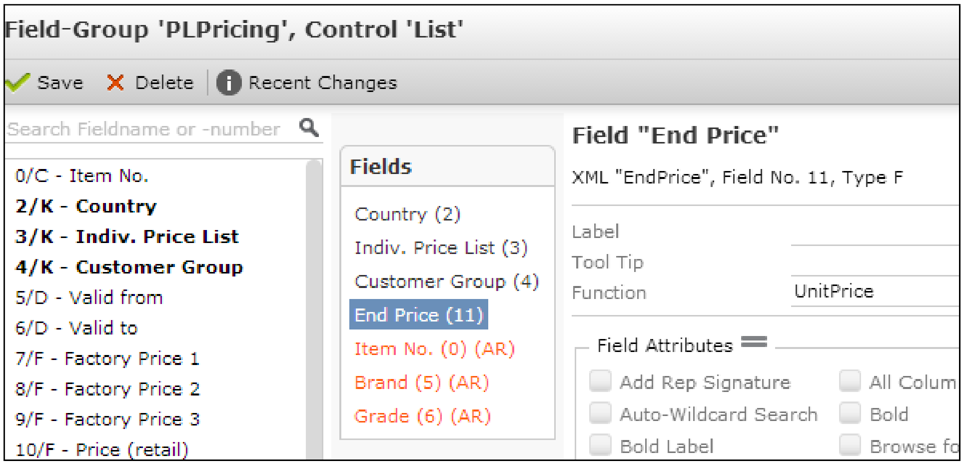
Allocation by AR record:

Ensure that one link field from the AR info area was added to the `PLPricing`'s List control (as shown in the example above).

Assigning Unit Prices

You can also assign unit pricing using functions.

The following example applies the `UnitPrice` function to the End Price (11) field:



Assigning Bulk Prices / Bulk Discounts

You can assign bulk prices or bulk discounts in price lists.

This is available by default in the ISI industry solution, if you assign price lists with the `ItemNumber` function.

You need to define the following **Function** names in the List control of the `PLOrder` field group.

Field	Function
Item No. (0)	`ItemNumber`
Cust.Group.Price List (1)	`Price List`
Selling Price (4)	`UnitPrice`
1st Volume Price (12)	`UnitPrice0`
2nd Volume Price (14)	`UnitPrice1`
3rd Volume Price (15)	`UnitPrice2`
4th - 10th Volume Price (15 - 21)	`UnitPrice3 - UnitPrice9`
1st Volume Dscnt.% (22)	`Discount0`
2nd - 10th Volume Dscnt.% (23 - 31)	`Discount1 - Discount9`

You can assign up to 10 bulk prices.

Which of the existing bulk prices are applied in the SM info area is defined by the `ItemNumber` function.

You need to define the following **Function** names in the List control of the `PLOrder` field group:

Field	Function
Product Group	`ProductGroup`
Product	`Product`
Item No.	`ItemNumber`
1st Bulk Vol. - 10th Bulk Vol.	`Qantity0 - Quantity9`

Specify the Search&List configuration referencing the `PLOrder` field group in the `BulkVolumeConfigName` input argument of the `Configuration :Pricing` action call (of the `Configuration:Pricing` context menu action).

You need not to define a bulk volume record for each item. You can also compare SM records using different item fields, e.g. Product Group or Product. In this case, they must be included in the source and destination field groups. In addition, specify the `BulkVolumeMatchFunctionNames` input argument of the `Configuration :Pricing` action calls.

Example:

`ProductGroup ,Product ;ProductGroup` defines that if the combination product group plus product is not found, only the product group is taken into account.

The different combinations (variants) are delimited by semicolon (;), the fields of each variant are separated by comma (,). Only SM records that have no `ItemNumber` function assigned can be searched.

Converting Currencies

Learn about defining currency conversion pricelist in CRM.pad.

You can define that price lists in other currencies than the order's currency can be applied. Prices are automatically converted based on the Base Currency info area. If no currency conversion was defined, ensure that only pricing data in the order's currency is given (e.g. using `Copy` fields or applying a Template filter).

To convert currencies:

1. You need to define a Search&List configuration that references a field group (e.g. `WKConversion`) whose List control has the required `Functions` assigned:

Field	Function	Comments
Base Currency 1	BaseCurrency	
Base Currency 2	BaseCurrency2	
Conv. Factor1	ExchangeRate	The exchange rate for Base Currency 1.
Conv. Factor 2	ExchangeRate2	The (optional) exchange rate for Base Currency 2.
Currency	Currency	The current currencies.

If you want to convert one currency to another, and both are not the base currency, the first currency is converted to the base currency and then converted from the base currency to the second currency.

2. Enter the Search&List name in the `CurrencyConversionConfigName` input argument of the `Configuration :Pricing` action call (from the `Configuration:Pricing` context menu action).
3. You also need to specify the `SourceCopyFieldGroup` input argument of the `SerialEntry` action call by entering the name of the field group that has the `Currency` **Function** assigned. For example, for an order the default field group's name is `PRCopyFields`.

Configuring Standard Conditions

Learn how to define the standard conditions that are applied.

To configure standard conditions:

1. Specify the `ConditionConfigName` and `ConditionScaleConfigName` input arguments of the `Configuration :Pricing` action call, see Configuration:Pricing on page 85.

Note: Usually you do not need to enter the `ConditionScaleConfigName` as the filter for the condition is automatically applied.

2. You need to define which fields are used as filter criteria in the List controls of the `KDPricing` or `SKPricing` field groups by assigning the required functions. For example:

KDPricing List Control.

Field	Function
Brand	Brand
Grade	Grade
Sector	Sector
Item No.	ItemNumber

Field	Function
Fixed price	UnitPrice
Discount	Discount
Discount vol. in kind	FreeGoods

SKPricing List Control

Field	Function	Description
Volume From	MinQuantity	Minimum quantity of a scale/bundle
Volume Up To	MaxQuantity	Maximum quantity of a scale/bundle
Discount	Discount	The discount applied to condition/scale/bundle
Fixed price	UnitPrice	A fixed unit price applied to condition/scale/bundle
Discount vol. in kind	FreeGoods	Free goods awarded to scale/bundle
Item No.		

3. In addition, the following functions are also available for the SK fields, but are not used in the standard version's default configuration:
   - `MinPrice`: Minimum sum price of a bundle
   - `MaxPrice`: Maximum sum price of a bundle

Configuring Company Conditions

Learn how to define the company conditions that are applied.

The same functions as for standard conditions are available.

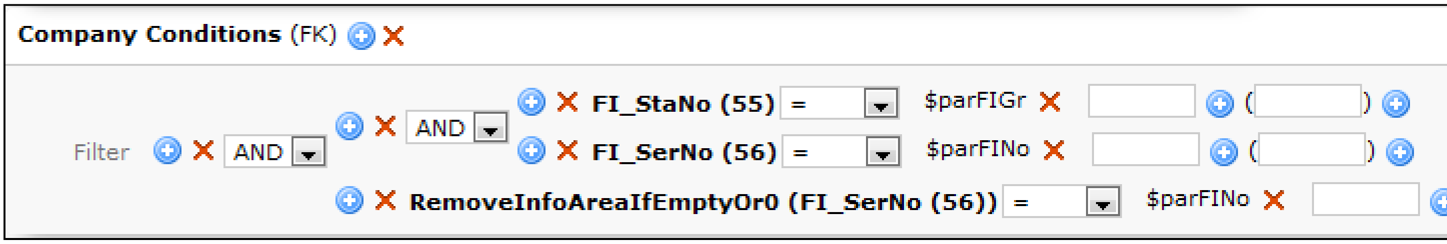
To configure the company conditions:

1. Specify the `CompanyConfigName` and `CompanyScaleConfigName` input arguments of the `Configuration :Pricing` action call, see Configuration:Pricing on page 85.
2. Define the company link fields manually, as they are not set by default.
3. Define which fields are used as filter criteria in the List controls of the `AUCopy-Fields` field group by assigning the required functions. For example:

Field	Function
Currency (4005)	Currency
FI-StaNo (0) (FI)	FiGr

Field	Function
FI-SerNo (1) (FI)	FiNo
Special Offer No. (59)	SpecialOfferNumber

4. The following `FK.CompanyFilter` is included in the `update.tablet` (BC) configuration:

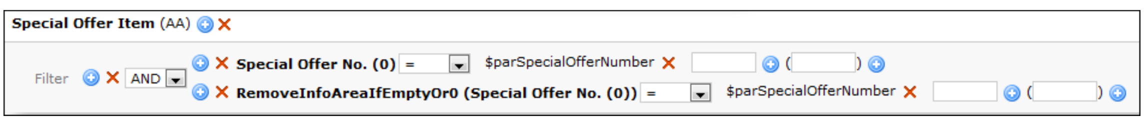


Configuring Special Offer Conditions

You can define that special offer conditions are applied.

To configure the company conditions:

1. Specify the `ActionConfigName` and `ActionScaleConfigName` input arguments of the `Configuration :Pricing` action call, see Configuration:Pricing on page 85.
2. The following `AA.SpecialOffer` filter (that can be given as `ActionConfigName` input argument) is included in the `update.tablet` (BC) configuration:



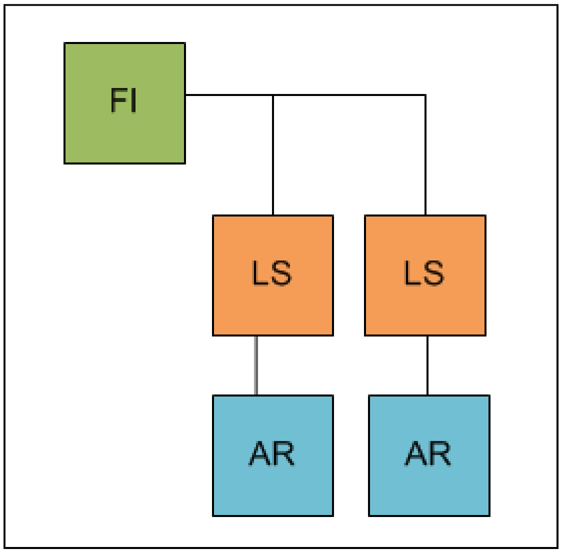
3. Ensure that the `RemoveInfoAreaIfEmptyOr0` condition is set to disable the pricing for special offers if `SpecialOfferNumber` is empty or zero.

Listing

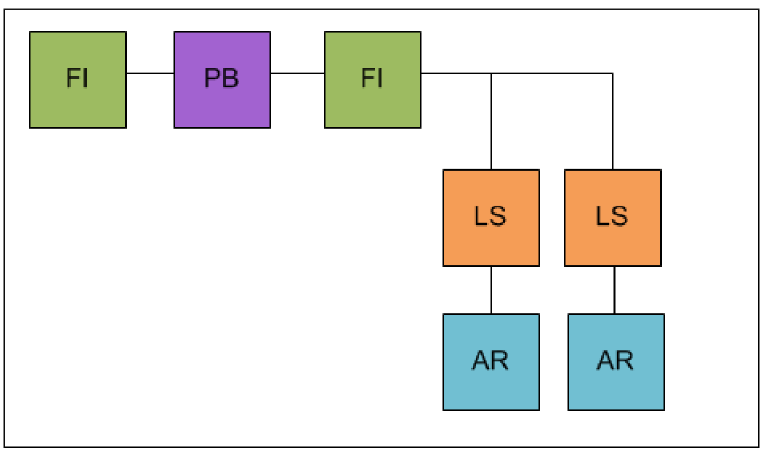
Information on which articles/products are listed in which companies are stored in the Listing (LS) info area.

The articles/products can be linked in the following two ways:

- Direct link to the company record:



- Link by a relationship hierarchy that is linked to the FI records:



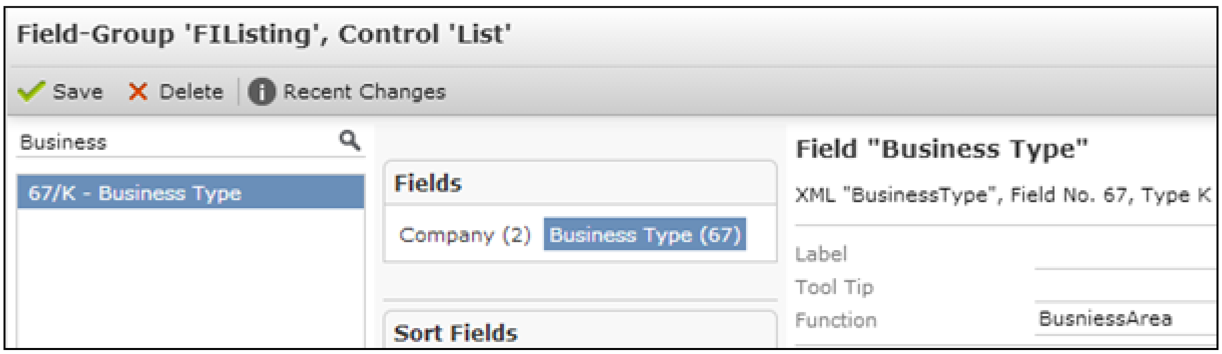
A Listing filter is displayed in CRM.pad containing all articles/products that are listed for the current company. Values from the available listings can be transferred to the quick add items where child records do not yet exist.

Configuring the Listing

To define that the listing is applied in the quick add:

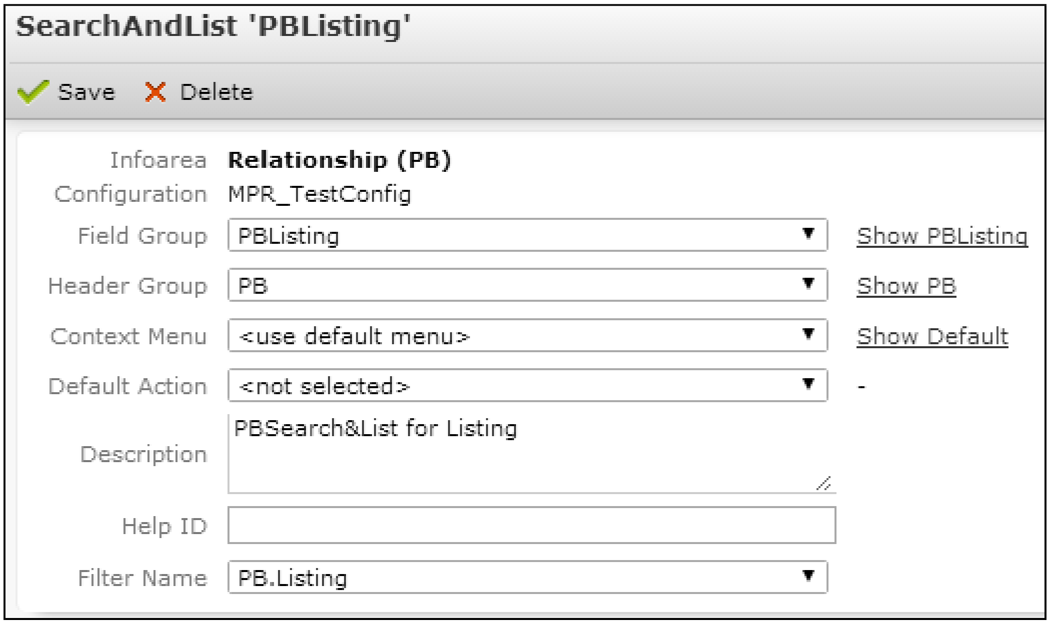
1. Specify the `ListingOwnerFieldGroupName` in the `Configuration :Quota` action call assigned, see Configuration:Pricing on page 85.

2. Enter the name of a Search&List configuration that references the field group that has the required **Functions** assigned in its List control. These functions define which fields from the FI record are loaded to filter the LS records.

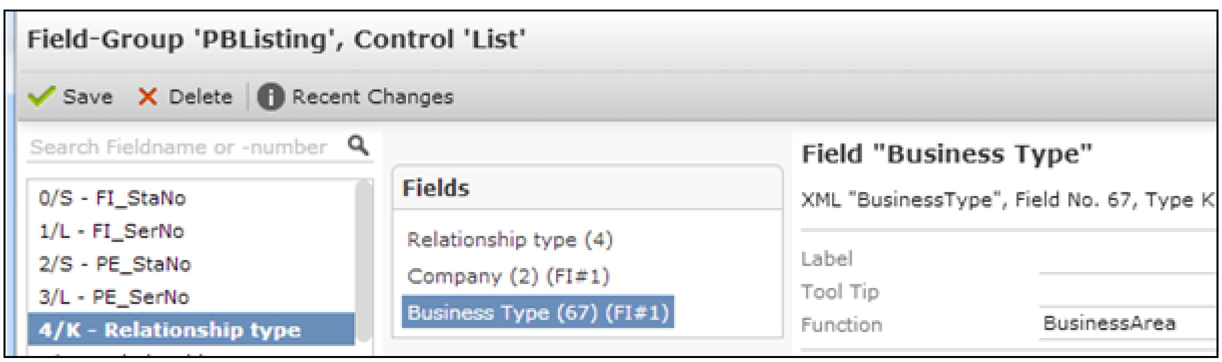


3. You can also specify the `RelatedListingOwnersConfigName` input argument that references a Search&List configuration that is used to search for all related FI records from a record based on the defined relationships (PB info area). The Listings for all FI records are also valid for the "root" FI record. For example, if no listings exist in the current company, listings from the related companies can be applied.

   In the default configuration, the `PBListing` Search&List configuration is used:



This Search&List configuration references the List control of the `PBListing` field group. This field group must contain the same fields than the field group referenced in the `ListingOwnerFieldGroupName` input argument. Ensure, that the `LinkId1` is set, otherwise the fields of the related companies cannot be identified:



4. Specify the `ListingControlName` input argument that references a Search&List configuration that is used to read the listing records linked with the FI record.

   In the default configuration, the `LSListing` Search&List configuration that references the `LSListing` field group is used. Assign the following **Functions** to the List control of this field group:

Field	Function
Item No.	CopyItemNumber
Sector	Sector
Brand	CopyBrand
Grade	Grade

Field	Function
Placing 1	Placing1
Placing 2	Placing2

5. Specify the Search&List configuration given in `ListingControlName`. Define this filter to check (at least) the validity of the LS records by comparing them to order or activity dates.

Applying a Listing Filter

For example, if you specify the `ApplyHierarchy` input argument by entering `Copy-ItemNumber ;CopyBrand ;Sector,Grade`, the listing record is mapped to an item if one field that has the `CopyItemNumber`, `CopyBrand; Sector` or `Grade` functions assigned matches. If more than one record exists, the first matching record is applied.

If a listing filter is applied for the order entry, only the records with a valid listing record are displayed.

For Point-Of-Sale you can also pass data from the listing record to the destination record, e.g. the placing. The listing record not only defines that an item is listed, but also where. In this case, an item can be listed more than once.

Each item is displayed according to the found listing records once or several times. If listing filter items are added, they are linked to the parent LS record.

Barcode Scanner

You can also use external barcode scanning devices or iPad's camera to add items (instead of using the keyboard).

To enable the barcode scanner:

1. Set the `ScanMode` input argument of the `SerialEntry` action call to `true`:

   An additional button ( ) is displayed. The user can turn the scan mode on or off with this button.

2. If you want to use iPad's camera, enable the `SoftScannerEnabled` Web Configuration parameter, see SoftScannerEnabled.

Note: The `SerialEntry` action template's `DestinationConfigName` and `Destination-ChildConfigName` input arguments determine the field group that is searched for a field with the `Value` function assigned. If a field of the same field group has the `StepSize` function assigned, the value of this field is used as step size (i.e. number of items) each time an article is scanned.

The user can tap on the button to enable the scan mode. The quick add switches to the shopping cart. As soon as a code is scanned the corresponding article is automatically referenced. The keyboard is not available as long as the scan mode is active.

Most barcode scanner devices can be set to stand-by mode (temporarily). In this case, the scan mode is still active, but the keyboard is displayed and the user can enter values manually. If the device is enabled again, the keyboard is closed and locked again.

You can define barcode commands in order to be able to navigate without keyboard:

- By creating custom commands to enter values. These command codes start with % followed by the value to be entered, e.g. %3 enters the value 3 in the current field. You can also enter multiple digits in one field by e.g. scanning %3 and %1 to enter the value 31.
- By setting Boolean fields with the barcode scanner:
  - `%Boolfield1%1` sets the logical (Boolean) field that has the `Boolfield1` function assigned is set to `true`.
  - `%Boolfield1%0` sets the field is set to `false`.
- To navigate by barcodes you can use the following commands:
  - `clear`: Clears the current field.
  - `delete`: Deletes the current record.
  - `next`: Moves to the next field.
  - `previous`: Moves to the previous field.
  - `ok`: Closes the active record.

Barcode Examples

The following barcodes are examples and were created with the "Code-128" barcode format. Since CRM.pad only interprets a barcode string that has already been decoded and that is sent by the scanning hardware, it actually does not matter which format you use. The scanner only and is only depends on the hardware you are working with.

Synchronizing Items

After you edit data using the quick add the item records are synchronized automatically.

However, linked records (e. g. containing pricing information) might not be synchronized. You can ensure that this data is synchronized as well.

Note: Some linked records may not be synchronized if there are custom filters defined on the record.

To synchronize linked data:

1. Switch to the action call that calls the quick add page, e. g. `SerialEntry` or `SerialEntryInput`.
2. Enter the following JSON into the `Options` parameter:

   `{"syncStrategy":"[info area"]}`, e.g. `{"syncStrategy":"UP"}`

Circle of Influence (CoI)

The circle of influence displays relationships between records in a graphical overview.

The nodes represent e.g. company and person records, the lines the relationship records.

If a node has more than 5 sub-nodes, two buttons with < and > are displayed. If the user clicks on them, the previous or next records are displayed.

Users can expand or collapse further levels by clicking on a node. The levels are color-coded depending on how remotely they are related to the center node.

A list of all available relationships can be displayed at the bottom of the graph.

You can define which records are displayed by specifying a fixed filter for the action,

In CRM.pad, two action templates are available to configure the Circle of Influence:

- CircleOfInfluenceView on page 77: With this action template you can only display relationships that are linked with the Relationship (PB) info area.
- CircleOfInfluenceTreeView on page 77: Use this action template to display links between any info areas, e. g. company, persons and activities of properties, property participants and property keys.

The circle of influence consists of nodes (representing records) and edges (illustrating the relationships), MiniDetails, and a toolbar containing functions applicable to the current node.

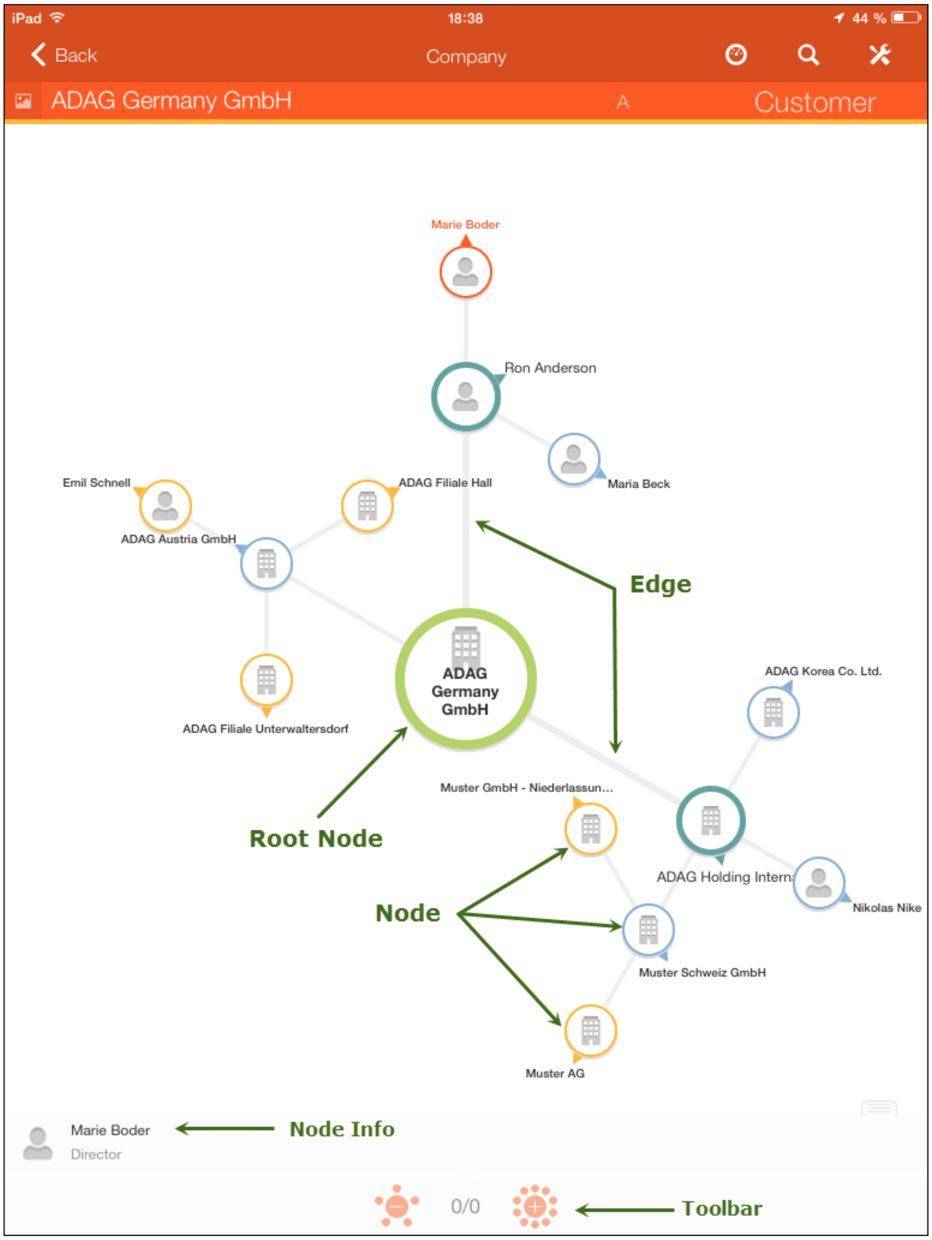
Terminology

Learn about the terminologies used in circleofinfluence.

The following terms are used:

- `Root Node`: The main node, i. e. the current record
- `Node`: Other related records
- `Edge`: The relationship connecting two nodes
- `Node Info`: Additional Information for the selected node
- `Edge List`: All relationships of the selected nodes. If the root node is selected, the relationships can be edited.
- `Toolbar`: Contains the actions that can be performed for the selected record (Show more/less).

Example:



Configuring a CoI Using the CircleOfInflucenceView Action Template

You can define a "basic" Circle of influence with the `CircleOfInfluenceView` action template.

For further details see, CircleOfInfluenceView on page 77. However, this action template offers fewer functionality than the `CircleOfInfluenceTreeView` action template, as you can only display records that are linked to the PB info area.

You need to define the following configuration units:

Field Groups

Add the following field groups to the following info areas:

Note: If you need to add linked fields, always add them with the **Add Linked Fields** button, even if "Z" fields (computed fields) are available in the info area.

Company (FI) - FI_CoI field group:

- **Details control:**

  Company (C/2)

- **Details control:**

  Company (C/2), Country (K/5), Street (C/10), Postal Code (C/6), City (C/8)

Relationship (PB) - PB_COI field group:

- **List control:**

Field	Function
Relationship type (4)	`relationName`
Company (2) (FI#1)	`trgtNodeFieldFI`
Last Name (3) (PE#1)	`trgtNodeFieldKP`
First Name (4) (PE#1)	`rgtNodeFieldKP`
Country (5) (FI#1)	`trgtNodeFieldFI`
City (8) (FI#1)	`trgtNodeFieldFI`

- **Search control:**

  Relationship type (4)

- **Edit control:**

  Relationship type (4), Relationship (5), Text (15), Last Name (2) (PE#1), First Name (3) (PE#1), Company (2) (FI#1)

- **MiniDetails control:**

Field	Function
Company (2) (FI#1)	`mainFieldKP,listFieldKP,mainFieldFI,listFieldFI`
Last Name (3) (PE#1)	`mainFieldKP,listFieldKP`
First Name (4) (PE#1)	`mainFieldKP,listFieldKP`
Country (5) (FI#1)	`mainFieldFI,listFieldFI`
Postal Code (7) (FI#1)	`mainFieldFI`
City (8) (FI#1)	`mainFieldFI`
Relationship type (4)	`listFieldFI`
Relationship (5)	`listFieldFI`

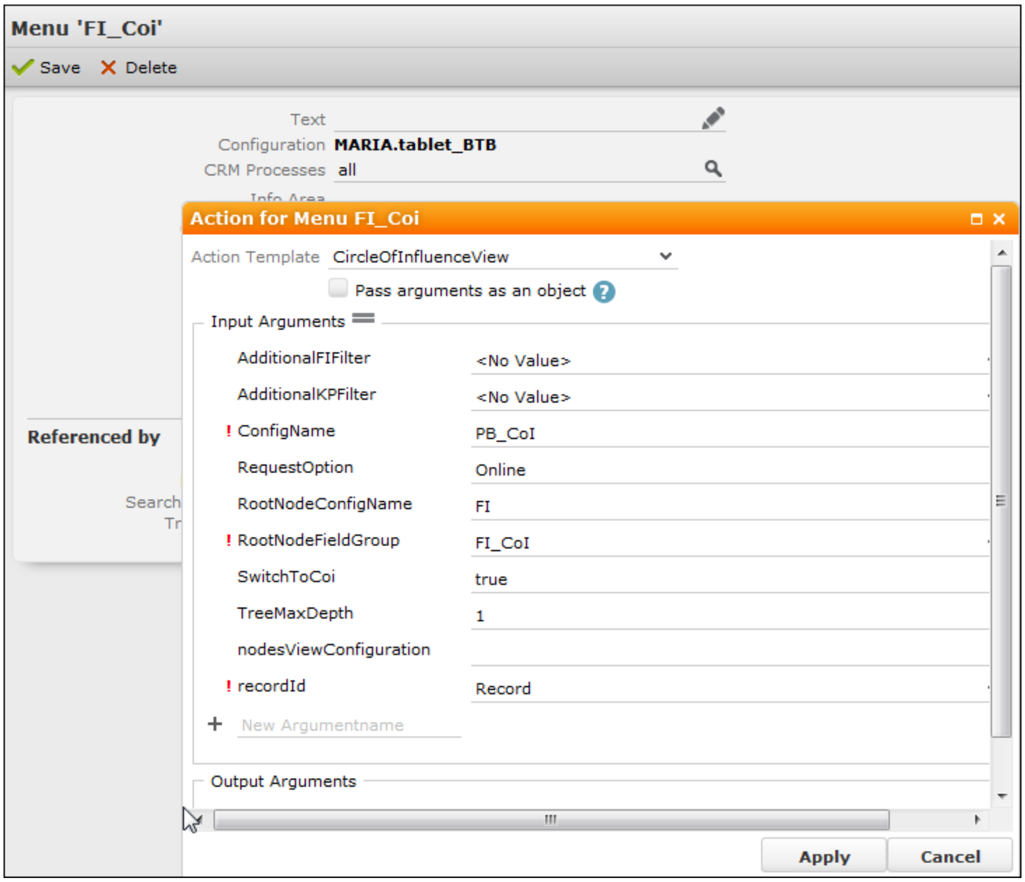
Search&List

Add a Search&List Configuration to the PB info area called `PB_COI` and select the **Field Group** you have created before (`PB_COI`).

Context Menu Action

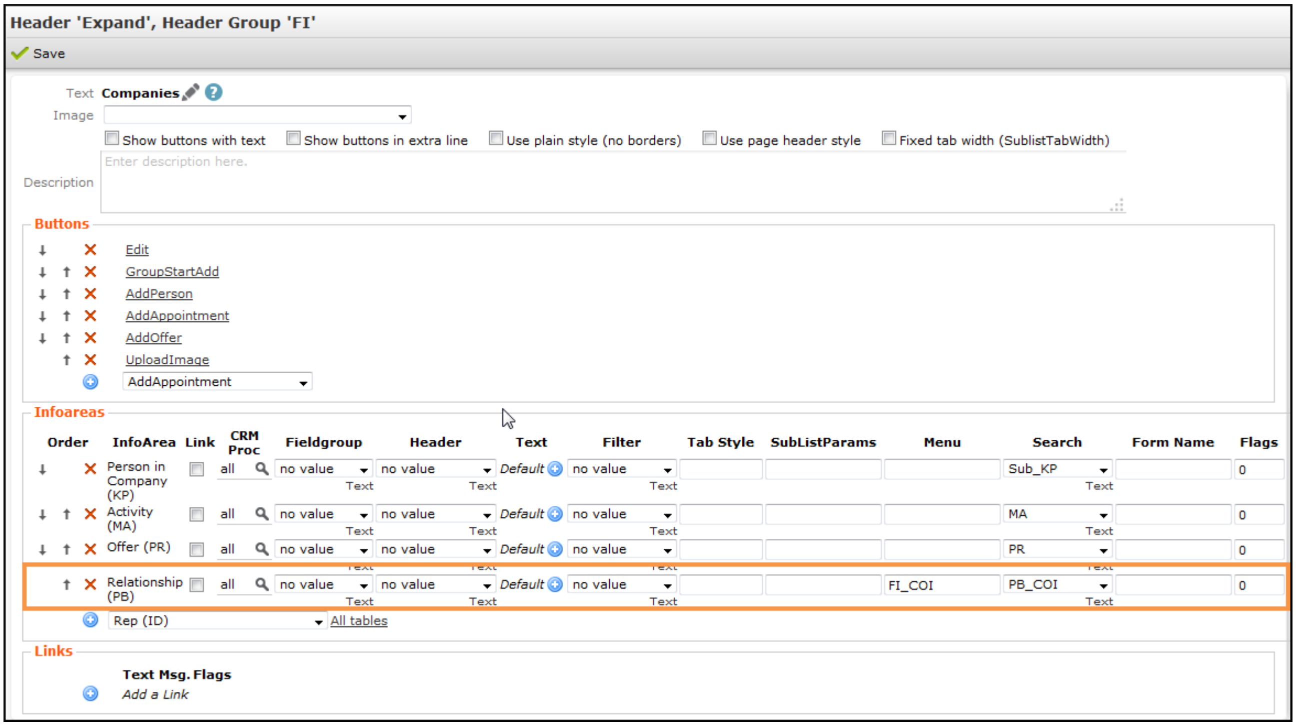
Define a context menu action and assign the `CircleOfInfluenceView` action, see CircleOfInfluenceView on page 77.

Example:



Header Group

Switch to the **Header Group** of the Company (FI) info area and add a line to the **Expand** header's **Infoareas** area:



Configuring a CoI Using the CircleOfInfluenceTreeView Action Template

You can also assign the `CircleOfInfluenceTreeView` action call to display more complex data in the Circle of Influence view.

For further details, see CircleOfInfluenceTreeView on page 77.

You need to define the following configuration units:

- Field Groups on page 321.
- Expand Configurations on page 324.
- Tree View on page 327.
- A context menu action calling the `CircleOfInfluenceTreeView` action.
- A line in the desired header group's `Infoareas` area e. g. FI of KP.

Field Groups

Learn how to add field group to info areas for configuring a CoI using the CircleOfInfluenceTreeView Action Template.

Add the following field groups to the following info areas:

Company (FI) - FI_CoI field group:

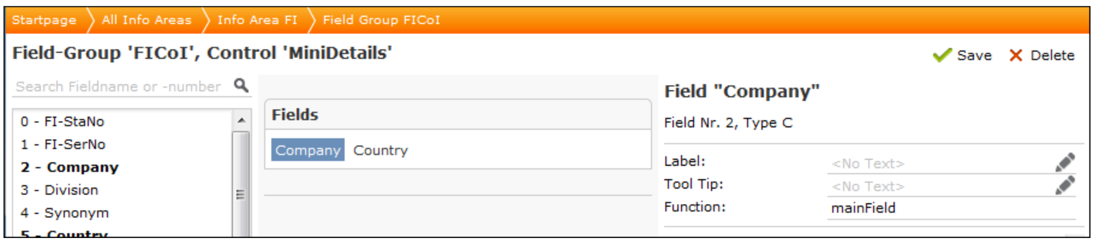
- **Details control:**

Add the fields you want to be displayed in the root node.

For example, if you only add the Company field, the company name is displayed:

- **MiniDetails control:**

  Add the fields you want to display as additional information, e.g.:



  You can assign the following functions:

  - `mainField` results in these fields being displayed as additional information at the bottom of the screen:

  - `listField` results in these fields being displayed in the edgle list.

Relationship (PB) - PBCoI field group :

- **List control:**

Add the desired fields and assign the corresponding functions. Add the Company, First Name and Last Name fields as linked fields with the link ID set to `#1`.

Field	Function
Company (2) (FI#1)	`trgtNodeFieldFI`
Last Name (3) (PE#1)	`trgtNodeFieldPE,initial2PE`
First Name (4) (PE#1)	`trgtNodeFieldPE,initial1PE`
Country (K/5)	`mainField`
Street (C/10)	`listField`
Postal Code (C/6)	`listField`
City (C/8)	`listField`

- **MiniDetails control:**

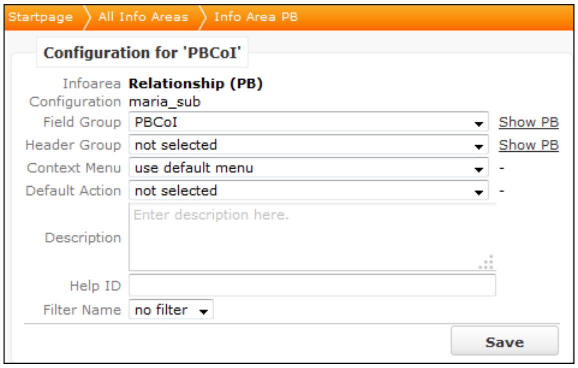
  Add the fields you want to display as additional information, e.g.:

Field	Function
Company (2) (FI#1)	`mainFieldFI,listFieldFI`
Last Name (3) (PE#1)	`mainFieldPE,listFieldPE`
First Name (4) (PE#1)	`subFieldPE,listFieldPE`
City (8) (FI#1)	`subFieldFI,listFieldFI`
Relationship type (4)	`listField`
Relationship (5)	`listField`

Search&List

Learn how to Search&List to filter the child records for configuring a CoI using the CircleOfInfluenceTreeView Action Template.

Add a Search&List to filter the child records, e. g. the relationships:



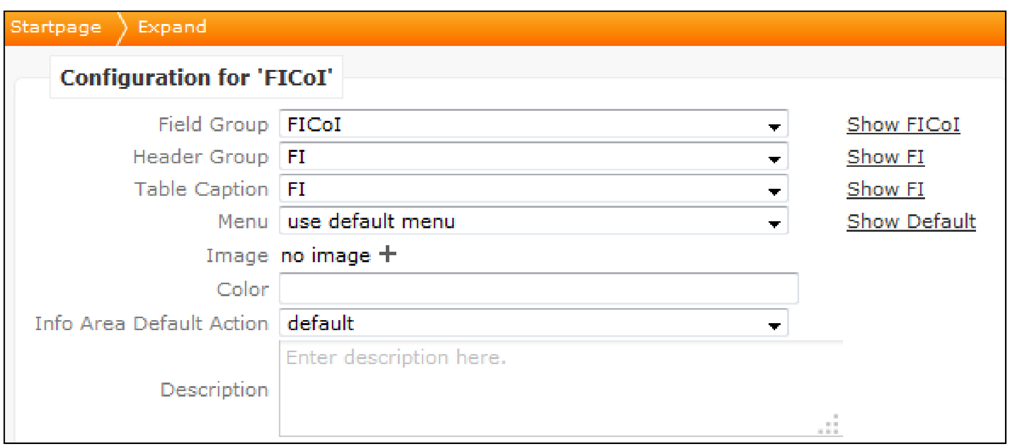
Note: If this Search&List has too many hits (e.g. if company "A" is queried, both companies and persons with relationships to company "A" are loaded). You can define a filter in this Search&List to limit the number of hits.

Expand Configurations

Learn how to add expand configurations for configuring a CoI using the CircleOfInfluenceTreeView Action Template.

Add the following Expand configurations:

FI info area:



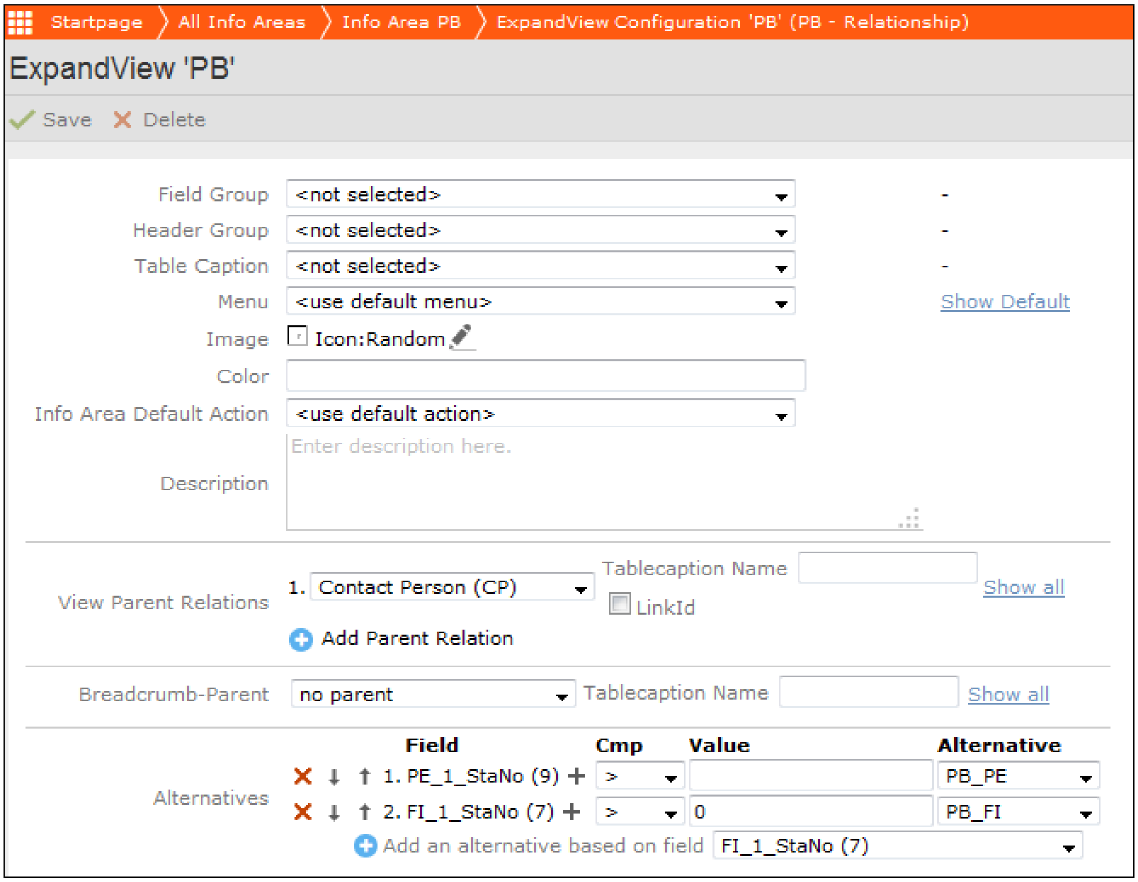
The image defined in this configuration is also displayed in the node info.

You can also define alternatives to adjust the icon displayed.

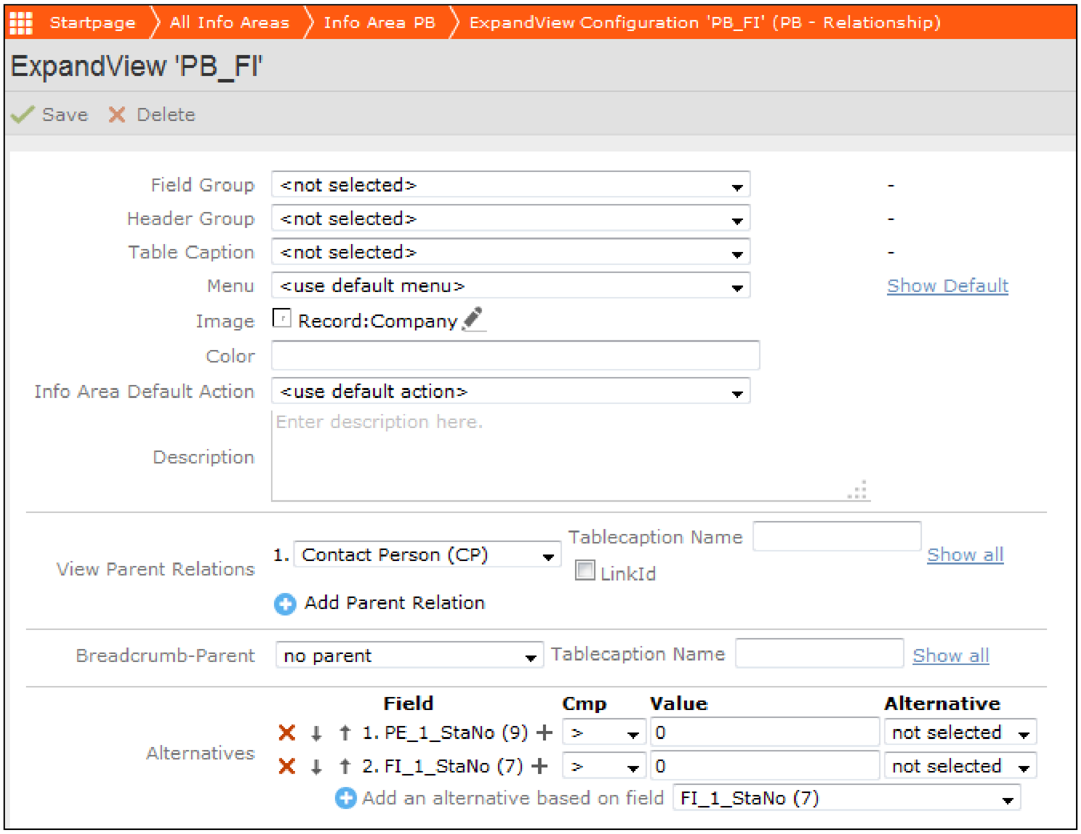
PB info area:

You need to define the following Expand configurations:

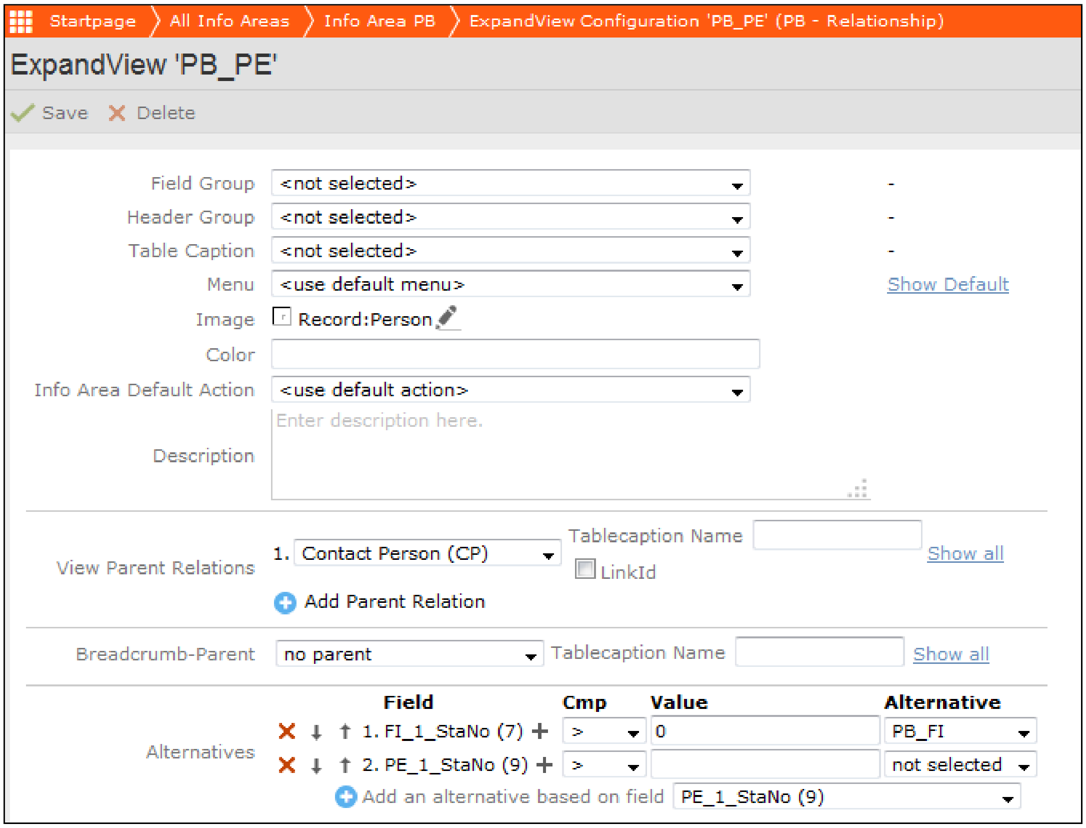
- **PB** (applied to the root node)



- **PB_FI** (applied to a linked FI node)



- **PB_PE** (applied to a linked PE node)

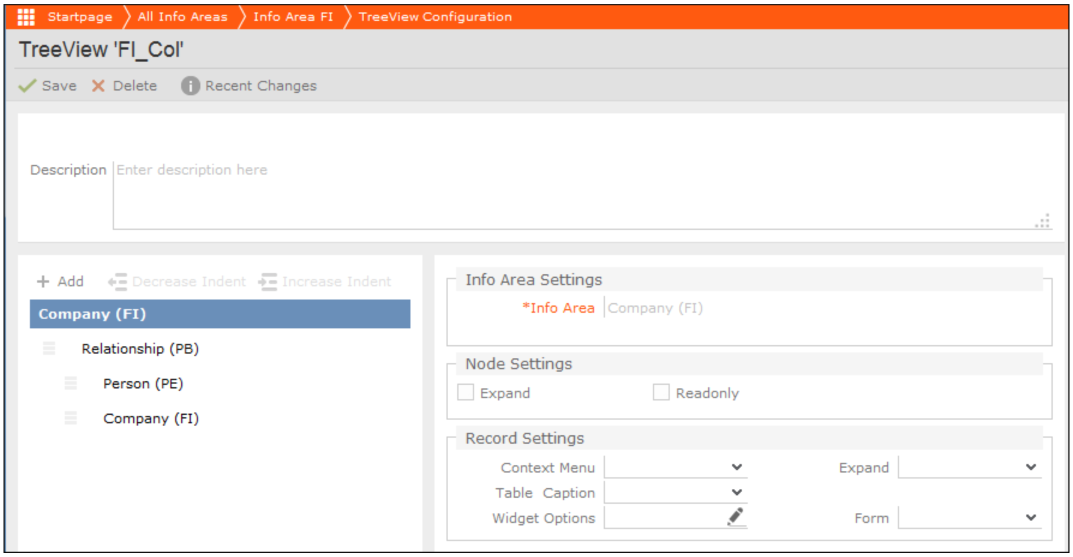


Tree View

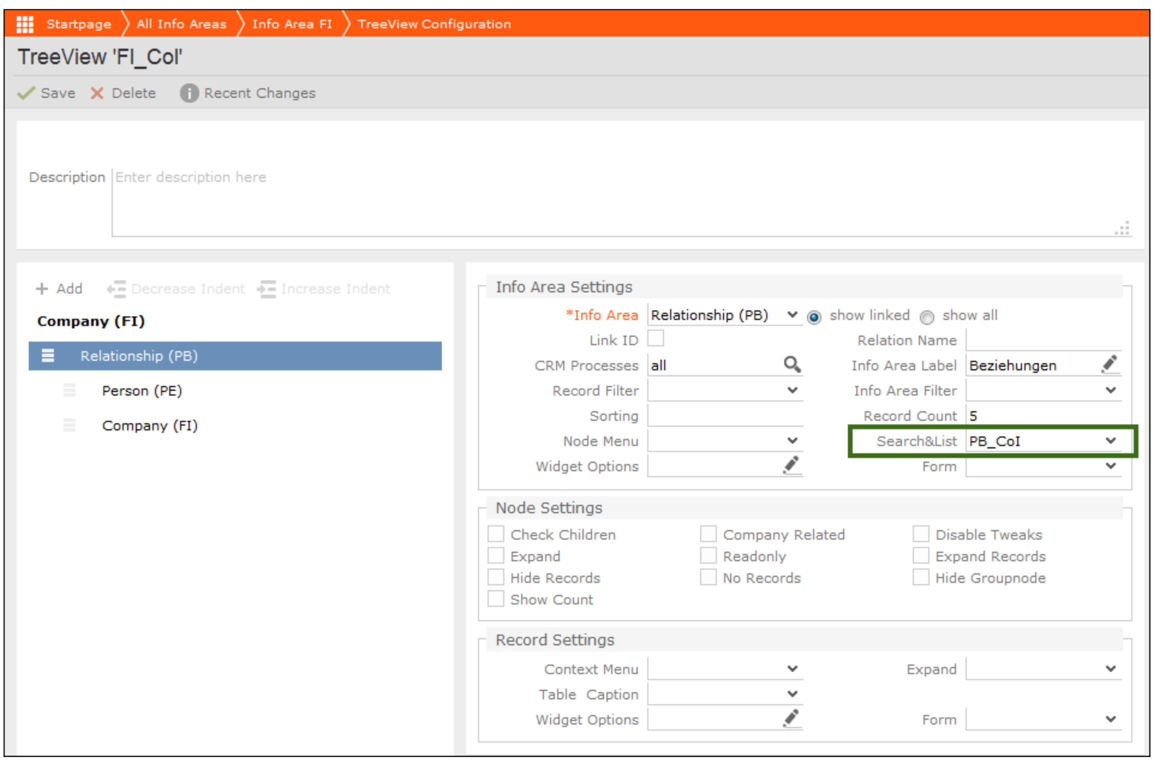
Learn how to define a treeview for configuring a CoI using the CircleOfInfluenceTreeView Action Template.

To define the Tree view:

1. Switch to the desired info area's field group, e. g. `FI_CoI`, add a tree view and define the root node, e.g. Company:



2. Add the Relationship (PB) info area and enter the name of the Search&List configuration you defined previously for the CoI in the FI info area, e.g. `FI_CoI`, in the **Search&List** field of the **Info Area Settings** area:



A query is executed for the info area given in the **Search&List** field using the root node's `RecordId`. In the example above, all relationships are loaded for he root node company.

3. You can also set the following options:
   - **Record Count**: Defines the max. number of hits.

     Note: The number of hits might not be the same than the existing number of nodes as several relationships can exist between two nodes. By default, the record count is set to 5 for performance reasons. For offline queries, a higher value might be applicable.

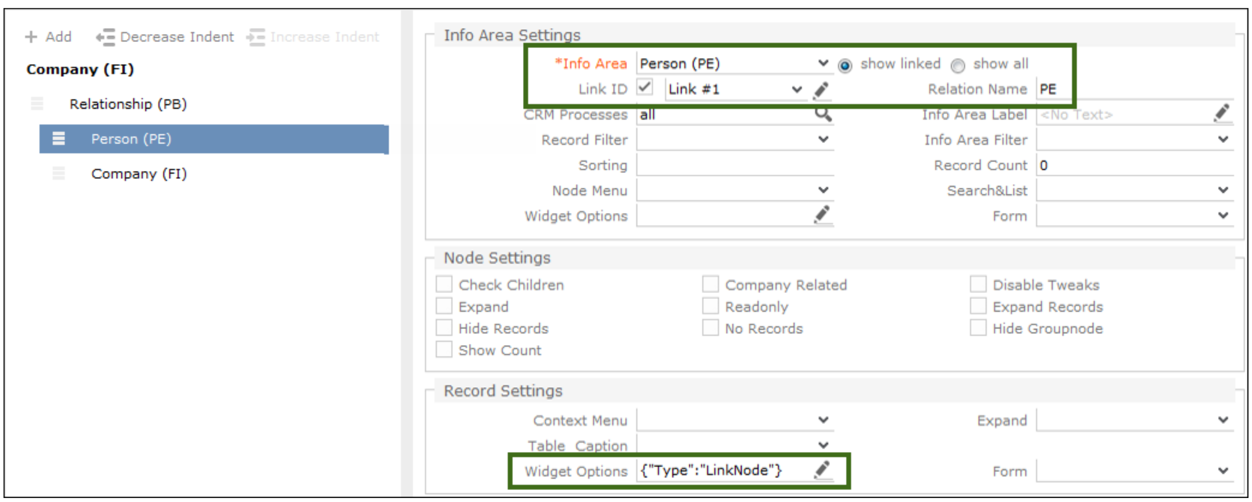
   - **Hide Groupnode**: If set, a group node is displayed for each given Search&List in separate tabs.

Specifying the Nodes Details

You can define that different fields are displayed in the CoI depending on the relationship type. For example, a relationship defined for a company is displayed differently from a relationship defined for a person.

To specify the nodes details:

1. Switch to the desired Tree view, e. g. `FI_CoI` and add another tree node, e.g. Person (PE):



2. Enter `{"Type":"LinkNode"}` in the **Widget Options** field.

   The records that have been loaded by the parent Search&List configuration are then checked whether the parent records can be linked with the current info area records. In the example above, the system checks whether the PB records can be linked with the PE record by the **Link ID** `#1`. If yes, the fields that have the `trgNodeField`, `listField` and `mainField` functions assigned apply the `trgNodeFieldPE`, `listFieldPE` and `mainFieldPE` instead.

Note: By default, the info area ID of the selected info area (e. g. PE for the Person info area) is added to the default function names as suffix. However, you can also enter another function name in the **Relation Name** field.

This node (`LinkNode`) only serves for further individualization of the results, no further queries are executed.

3. Add another node for the Company (FI) and set the same parameters than for the Person (PE) info area. This allows to display different fields for these two info areas.

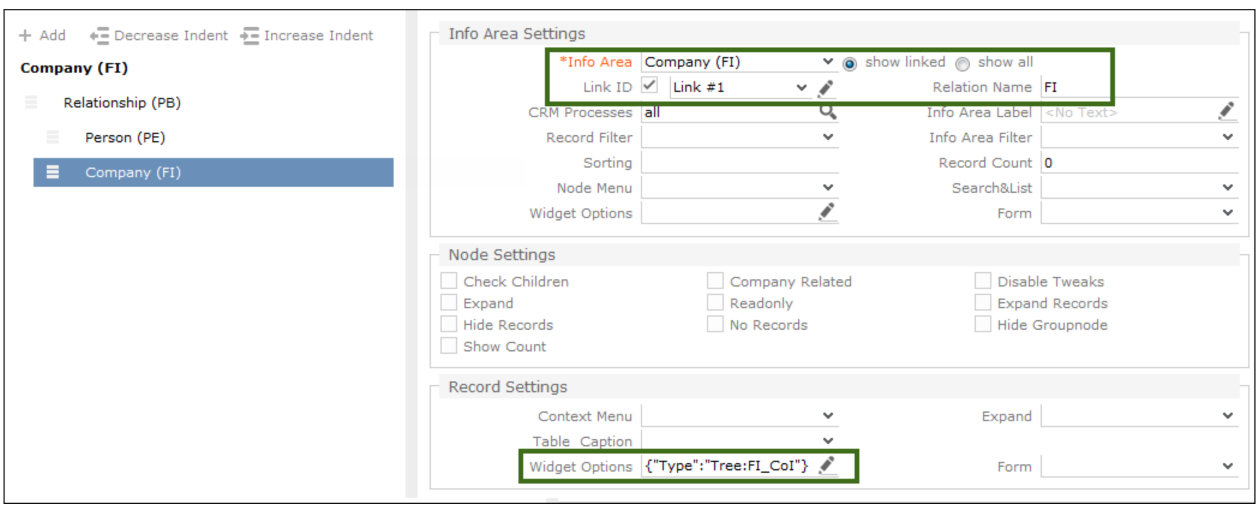
Note: The order of the given info areas matters in this example. Usually, a PB/PE relationship also checks for a FI link. You can avoid this by adding the tree view node for PE before the FI node.

Adding Further Info Area Nodes

You can either add further info area nodes or you can refer to other ones or even the same tree view node.

To add further nodes, e. g. companies:

1. Switch to the desired tree view, e. g. FI_CoI and add another tree node, e.g. Company:



2. Define the **Info Area**, the **Link ID** and the **Relation Name** in the **Info Area Settings** area.

Enter `{"Type":"Tree :FI _CoI"}` (where `FI_CoI` is the name of the desired tree view configuration) in the **Widget Options** field. If a record that is linked by the **Link ID** `#1` is found, the `FI_CoI` tree view is applied and this node is handled as if it was the root node of the `FI_CoI` tree view.

3. The following **Widget Option** functions are also available:
   - `EdgeEditAction`: To specify a context menu action that is executed if the Edit button of the edge list is tapped.
   - `NoteDetailAction`: To specify a context menu action that is executed when the user taps on a row in the edge list or double-taps on a node.

   Example:

   `{"Type":"Tree:PE_CoI","EdgeEditAction":"EDIT_PB_PERecord";"NodeDetailAction":"SHOW_PB_PEDetails"}`

Example for Displaying a Property Structure

You can also use the Circle of Influence to depict the structure of a property (OJ), its sub-properties (OJ) and the participants (OB) involved.

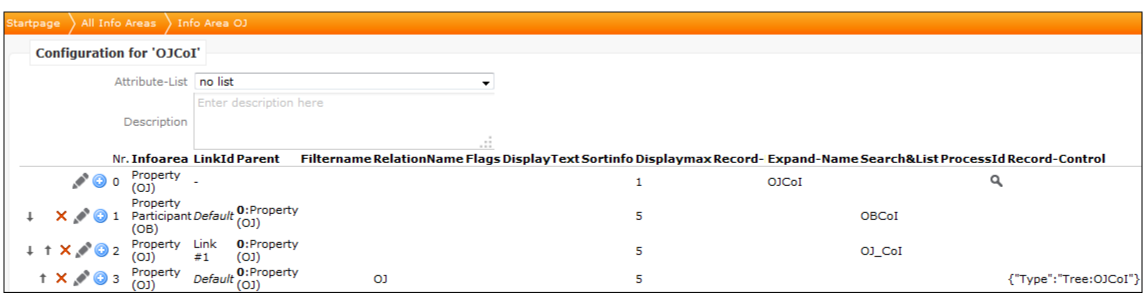
Example:

- `Root node`: OJ
  - `Node`: Search&List OB
  - `Node`: Search&List OJ (for sub-properties)
    - `Link Node:` OJ referring to `Tree:OJ`

An OJ root node has OB child records and is linked to OJ sub-properties. The sub-properties are recurring to themselves, as these sub-properties may have participants and sub-properties as well. They can only be displayed, when the sub-property becomes a root node.

To configure this:

1. Define the required field groups, Search&List, Expand configuration, etc.
2. Define a Tree view, e.g.:


Startpage 〉 All Info Areas 〉 Info Area OJ

Configuration for 'OJCoI'

Attribute-List: no list

Description: Enter description here

Nr.	Infoarea	LinkId	Parent	Filtername	RelationName	Flags	DisplayText	Sortinfo	Displaymax	Record-	Expand-Name	Search&List	ProcessId	Record-Control
0	Property (OJ)	-							1		OJCoI			
1	Property Participant (OB)	*Default*	**0**:Property (OJ)						5			OBCoI		
2	Property (OJ)	Link #1	**0**:Property (OJ)						5			OJ_CoI		
3	Property (OJ)	*Default*	**0**:Property (OJ)		OJ				5					{"Type":"Tree:OJCoI"}

Example for Displaying a Company Structure

You can also use the Circle of Influence to depict the structure of a company, its contact persons and the activities involved.

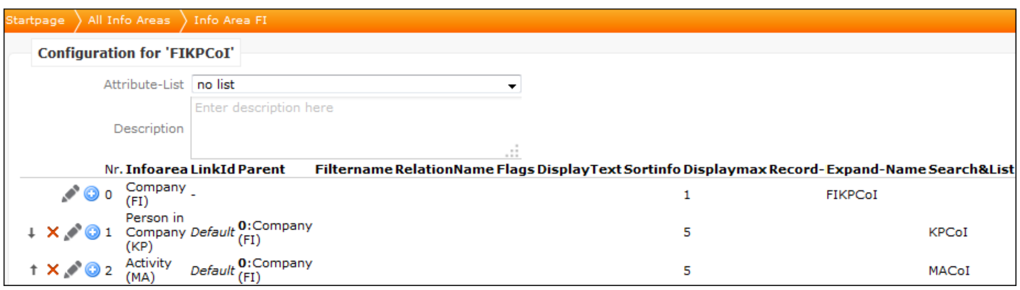
Example:

- `Root node`: FI
  - `Node`: Search&List PE
  - `Node`: Search&List MA

An FI root node has PE and MA child records. The related contact persons and activities are displayed in the Circle of Influence.

To configure this:

1. Define the necessary field groups, Search&List, Expand configuration, etc.
2. Define a Tree view, e.g.:



Interests/Characteristics

You can define that a user can store a set of interests/characteristics for a company or a person in the Interest (IT) info area.

The user can select values from the following catalogs in the IT info area:

- Interest Group (45) defining the characteristics group
- Interest (46) defining each characteristic

You can configure interests/characteristics as follows:

- Displaying Interests/Characteristics in a Company's the Details View on page 333.
- Configuring Interests/Characteristics by Adding Info Areas to the Data Model on page 335.

Displaying Interests/Characteristics in a Company's the Details View

A quick way to display characteristics exists for the company info area.

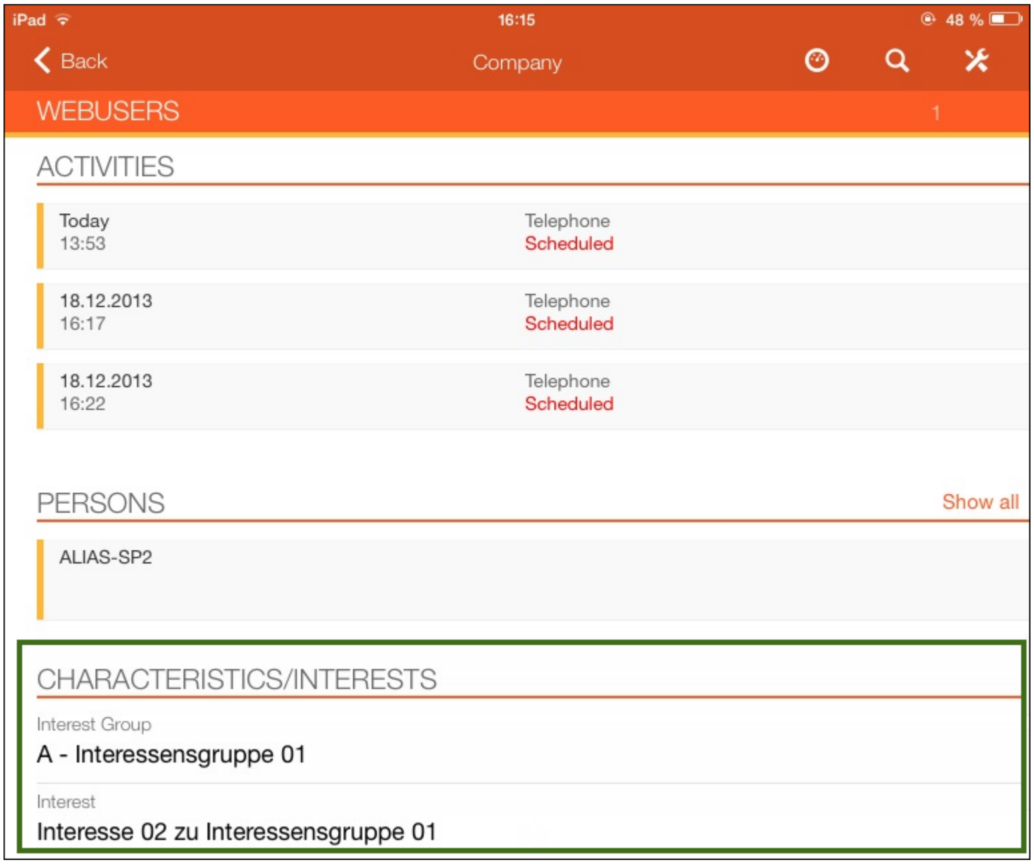
Note: At present, you cannot add a characteristics panel to the person (KP, PE) info areas.

To display the characteristics in the company's Details view:

1. Add a panel to the Details control of the desired field group (e.g. FI) and assign the `CHARACTERISTICS` type, see CHARACTERISTICS:
2. Add the following fields and assign the following functions:

Field	Function
Interest Group (0) (IT)	Group
Interest (1) (IT)	Item
Date and FreeC1	No function required
Any other field	ShowAdditionalFields

3. Ensure that the IT field group is also configured. At least the List control must contain the above fields (functions need not be assigned).
4. The characteristics are displayed in CRM.pad as follows:



Editing Characteristics

To enable editing characteristics in a Company's Details view:

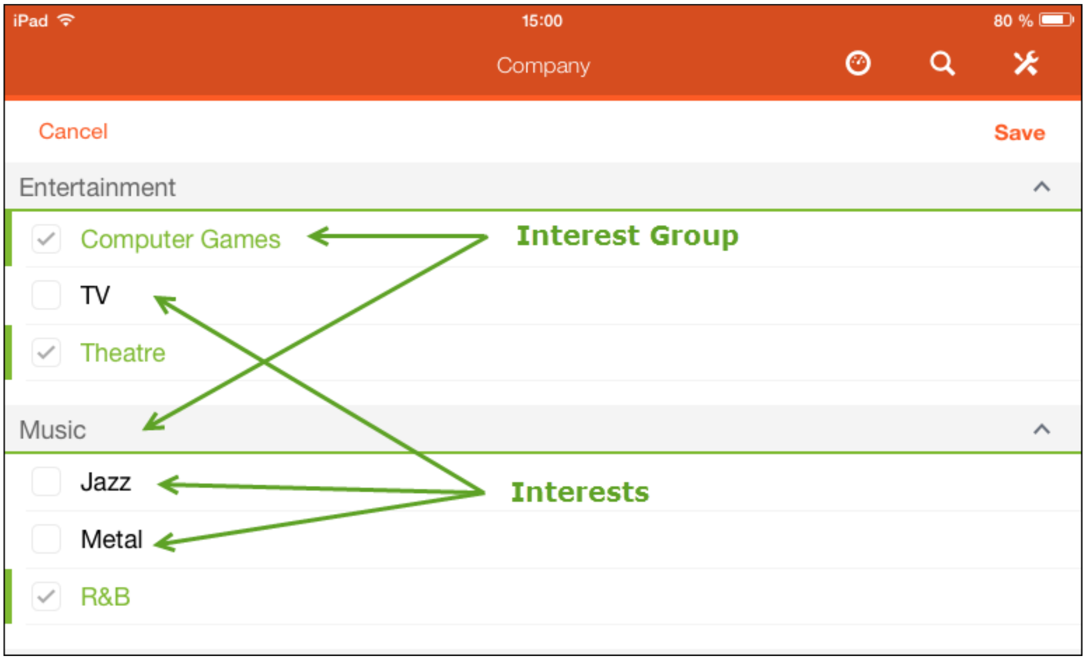
1. Ensure that a List control exists in the IT info area and assign the following **Functions**:

Field	Function
Interest Group (0) (IT)	Group
Interest (1) (IT)	Item
Date and FreeC1	No function required
Any other field	ShowAdditionalFields

2. Assign the `CharacteristicsEdit` action (usually to a button in the FI or KP info areas) and to set the following parameters:

   - `DestinationFieldGroup`: Enter the field group you previously defined.
   - `RecordId`: Set to `Record`.

   When the user taps on the button to edit the interests/characteristics, check boxes are displayed allowing the user to select the desired data:



Configuring Interests/Characteristics by Adding Info Areas to the Data Model

Interests/Characteristics can be added by adding additional info areas.

For example, if you work with the LSI industry solution, the default configuration already contains Characteristics (Interests are called Characteristics in LSI) using info areas that are not part of the Aurea CRM default data model.

The following example describes how to configure interests by adding two additional info areas. All given names are examples and are not included in the default configuration.

Adding Info Areas

Add two info areas, e. g.:

- C001 - Characteristics Group:

  Add all the required fields. The info area must contain the Interest Group (45) catalog.

C001

	ID	Name	Typ	Referenz	Katalog	DB-Spalte	Zeitste	Länge	Eig
0	4001	StNo.	sno(2Byte)			ID		5	no
1	4002	lNo.	lno(4Byte)			ID		9	no
2	4010	Neu	date(4Byte)	Erstellt um		NEW_DT		10	
3	4011	Upd	date(4Byte)	Aktualisiert um		UPD_DT		10	
4	4012	Astnr.	sno(2Byte)			NEW_SNO		5	no
5	4013	Löskz.	y/n			DEL		1	
6	4014	Komkz.	y/n			COM		1	
7	4015	Erstellt um	me(hh:mm:ss.ttt) T.	Neu		NEW_DT		12	rea
8	4016	Aktualisiert um	me(hh:mm:ss.ttt) T.	Upd		UPD_DT		12	rea
9	7001	Merkmalgruppe	var catalog		(22) Interessensgr.	F7001	✔	80	
10	7002	Merkmal	var catalog(child)	Merkmalgruppe	(23) Interesse	F7002	✔	80	
11	7003	Gruppe	y/n			F7003	✔	1	
12	7004	Einzelauswahl	y/n			F7004	✔	1	
13	7005	Sperre	y/n			F7005	✔	1	
14	7006	Gruppenname	text(80)			F7006	✔	80	
15	7007	Anmerkung	text(200)			F7007	✔	200	
16	7008	Aufklappen	y/n			F7008	✔	1	

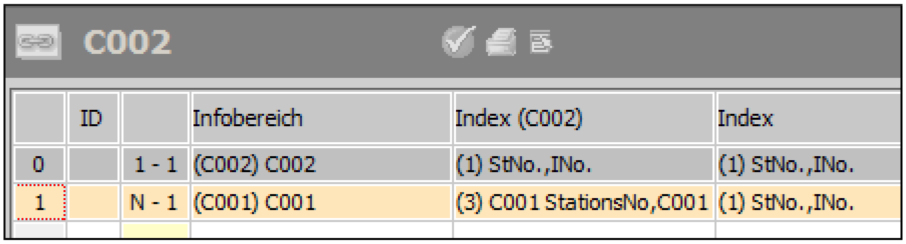
- C002 - Characteristics:

  Add all the required fields. The info area must contain the Interest Group (45) and Interest (46).

C002

	ID	Name	Typ	Referenz	Katalog	DB-Spalte	Zeitste	Länge	Ei
0	4001	StNo.	sno(2Byte)			ID		5	n
1	4002	lNo.	lno(4Byte)			ID		9	n
2	4010	Neu	date(4Byte)	Erstellt um		NEW_DT		10	
3	4011	Upd	date(4Byte)	Aktualisiert um		UPD_DT		10	
4	4012	Astnr.	sno(2Byte)			NEW_SNO		5	n
5	4013	Löskz.	y/n			DEL		1	
6	4014	Komkz.	y/n			COM		1	
7	4015	Erstellt um	me(hh:mm:ss.ttt) T.	Neu		NEW_DT		12	re
8	4016	Aktualisiert um	me(hh:mm:ss.ttt) T.	Upd		UPD_DT		12	re
9	7001	C001 StationsNo	sno(2Byte)	Link->(C001) C001: StN		ID_C001	✔	5	n
10	7002	C001 LaufNo	lno(4Byte)	Link->(C001) C001: lNo		ID_C001	✔	9	n
11	7003	C001 Link	reference (table)	Link->(C001) C001				60	
12	7004	Interessensgrup	var catalog		(22) Interessensgr.	F7004	✔	80	
13	7005	Interesse	var catalog(child)	Interessensgruppe	(23) Interesse	F7005	✔	80	
14	7006	Interaktive Admi	y/n			F7006	✔	1	

Define a link to the C001 info area:



C002

	ID		Infobereich	Index (C002)	Index
0		1 - 1	(C002) C002	(1) StNo.,INo.	(1) StNo.,INo.
1		N - 1	(C001) C001	(3) C001 StationsNo,C001	(1) StNo.,INo.

Note: See Data Model in the CRM.core Administrator Guide for information on how to add info areas to the default data model and CRM.bulkloader in the Aurea CRM web Administrator Guide on how to transfer the data model changes to the CRM.designer database.

Defining the Field Groups

You need to define the following field groups:

- **C001 (C001 info area):**
  - **Details control:**

    Add at least the Interest Group field.
  - **List control:**

    Add at least the Interest Group field. Assign the following functions:

Field	Function	Description
Interest Group	`Group`	Defines the Interest Group.
A boolean field	`Single`	Defines that the user can only select one item within an Interest Group.
A boolean field	`ShowGroupExpanded`	Defines that this group is displayed expanded.

- **C002 (C002 info area):**
  - **List control:**

    Add at least the Interest Group field. Assign the following functions:

Field	Function	Description
Interest Group	`Group`	Defines the Interest Group.
Interest	`Item`	Defines the Interest.
A boolean field	`ShowAdditionalFields`	Defines that additional fields are also displayed.

- **IT (IT info area):**

- **Details and List control:**

  Add at least the Interest Group and the Interest fields. If you have added other fields that have the `ShowAdditionalFields` function assigned, add them as well. Assign the functions according to the List control of the C002 info area.

- **Edit control:**

  Add at least the Interest Group and the Interest fields and assign the `Group` and `Item` functions. If you want to add further fields, you need not assign the `ShowAdditionalFields` function.

  Search control:

  Add the Interest Group and the Interest fields, you need not assign a function.

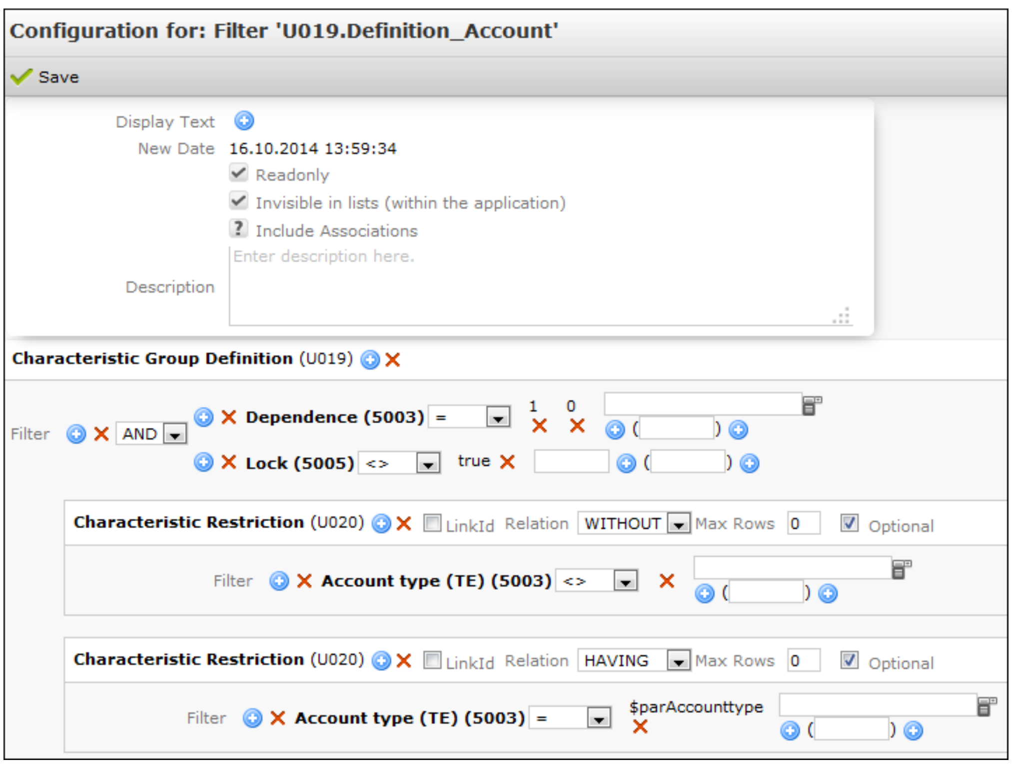
Note: Ensure that also a Header Group, an Expand configuration and Table Captions exist for the field groups.

Defining the Filters

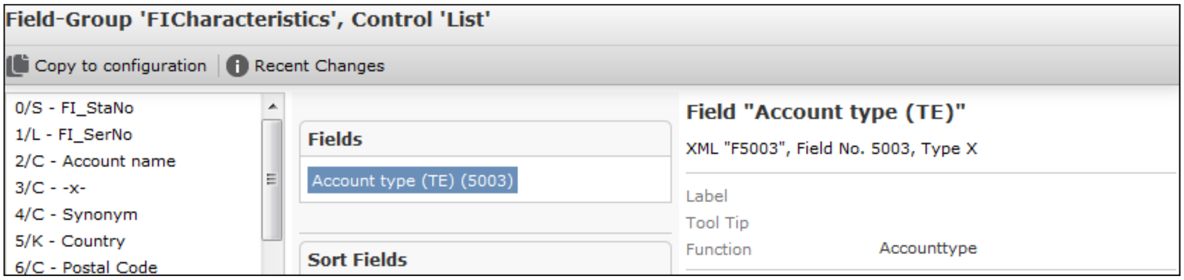
You can define that filters are also applied.

The following example shows filters that are applied in the default LSI industry solution:

The `U019.Definition_Account` filter uses the `Accounttype` function name as `$parAcccounttype` parameter:



However, to apply this filter, you must add a field group to the FI info area and add the following field and function:



Defining the Search&List Configurations

Define Search&List configurations for the C001, C002 and the IT info areas by assigning the desired field groups, header groups, filters etc.

Assign these Search&List configurations to the following input arguments in the `Characteristics` action call, see Characteristics on page 76:

Input Argument	Search&List
GroupSearchAndList	C001
ItemSearchAndList	C002
DestinationFieldGroup	IT

Editing Interests/Characteristics

Learn how to edit interests and characteristics.

To allow the user to edit interests/Characteristics assign the `CharacteristicsEdit` action call to a button or menu action. For further details, see Defining the Buttons and Context Menu Actions and CharacteristicsEdit on page 76.

Displaying All Fields of the Field Group

Learn how to display all fields of the field group.

If you have added more than the Interest and Interest Group fields in your field group, and you want to display all fields, ensure that `Characterristics.HideAdditionalFields` Web Configuration parameter is **NOT** set. For further details, see Characteristics.HideAdditionalFields.

Queries and Analyses

Learn how to include queries and analyses in CRM.pad.

Note: Analyses are available as of CRM.pad 2.1, queries as of CRM.pad 2.1.1.

Analysis and query names as well as button and menu names **must not contain underscores**, as underscores are used as delimiters.

Configuring Analyses

This chapter only describes how to configure an analyses to be displayed in CRM.pad. See the Aurea CRM web Administrator Guide on defining an analyses (e.g. the included values and categories).

Analyses are executed locally on your iPad.

To add an analysis:

1. Define a configuration unit that calls the analysis:
   - Button, see Buttons on page 63.
   - Menu action or context menu action, see Menus on page 62.
   - A panel in the Details control, see ANALYSES_<AnalysisName>.

     Users cannot manipulate the analysis in this panel. However, you can add a button to the page's header allowing the users to switch to the analysis page where the available options like drill-down, drill-up, etc. are available.

2. Assign the `Analysis` action template to the desired configuration unit, see Analysis on page 73.
3. Add the configuration unit to the desired info area.

Configuring Queries

See the Aurea CRM web Administrator Guide on defining a query.

Queries can be executed on- and offline. However, if a query was not explicitly set to `RequestOption = online`:

- Multiple child relations are not supported.
- `SUMMED` relations are not available.

To add a query:

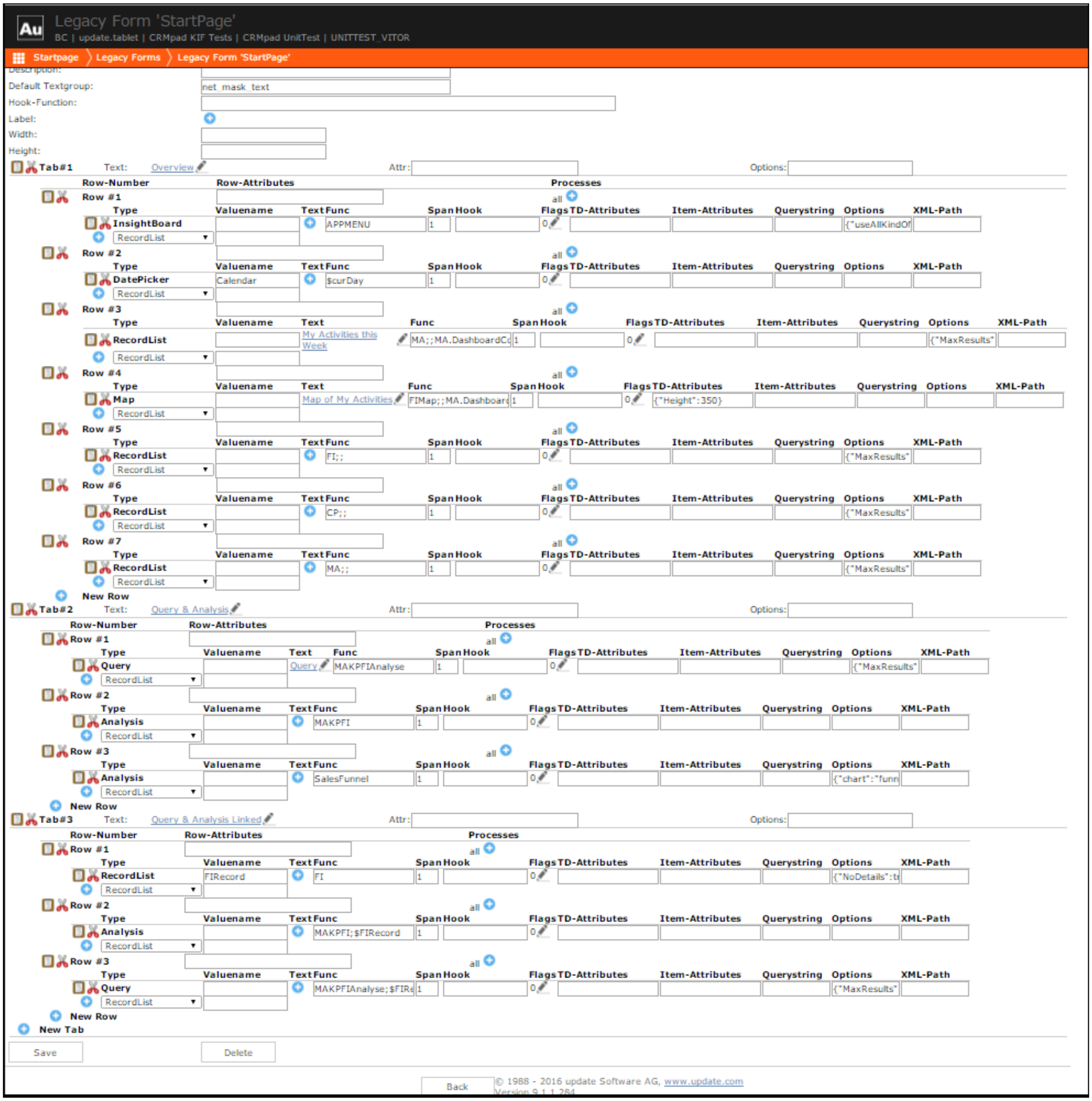
1. Define a configuration unit that calls the analysis:
   - Button, see Buttons on page 63.
   - Menu or context menu action, see Menus on page 62.
   - A panel in the Details control, see QUERY_<QueryName>_<No>.
2. Assign the `Query` action template to the desired configuration unit, see Query on page 113.
3. Add the configuration unit to the desired info area.

Adding pre-defined Analyses and Queries to CRM.pad Dashboard

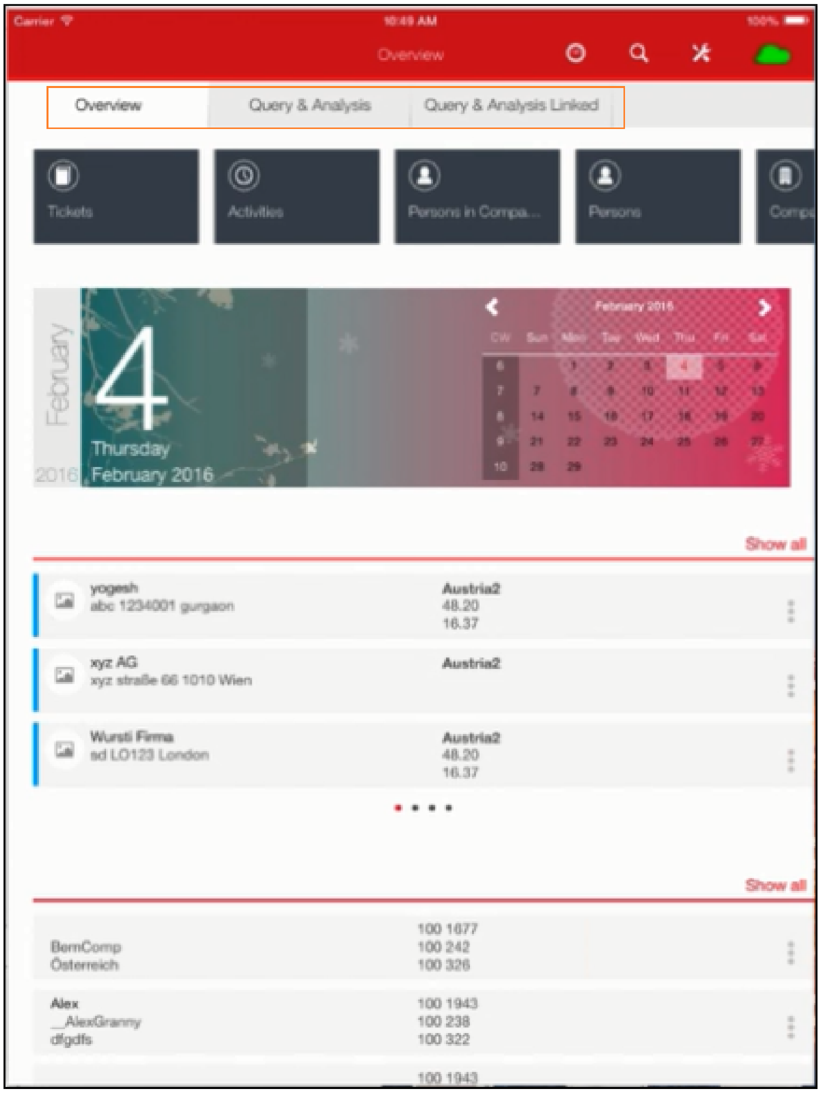
To add Analysis and Queries to a CRM.pad application dashboard, perform the following steps in the CRM.designer:

1. Click **Legacy Forms** on the **Start page** of the CRM.designer.
2. Enter a name for the legacy form that displays the Queries and Analysis in the CRM.pad.
3. Use the **copy from** drop-down list to select an existing legacy form as a template to create the new form. Click the plus button to add the new form.
4. Click the Edit button to edit the new form and add the pre-defined Queries and Analyses to the form.
5. Select the Mask Type and enter the required details for the Legacy forms. For more information on working with Legacy forms, see Legacy Forms.
6. Click on **New Tab** and provide the **Text** for the tab in the CRM.pad.
7. Click **New Row** to add a new record to the tab, this record describes the information that needs to be displayed in the Tab for the Start page of the CRM.pad. For more information on configuring a row, see Analysis and Query Types.

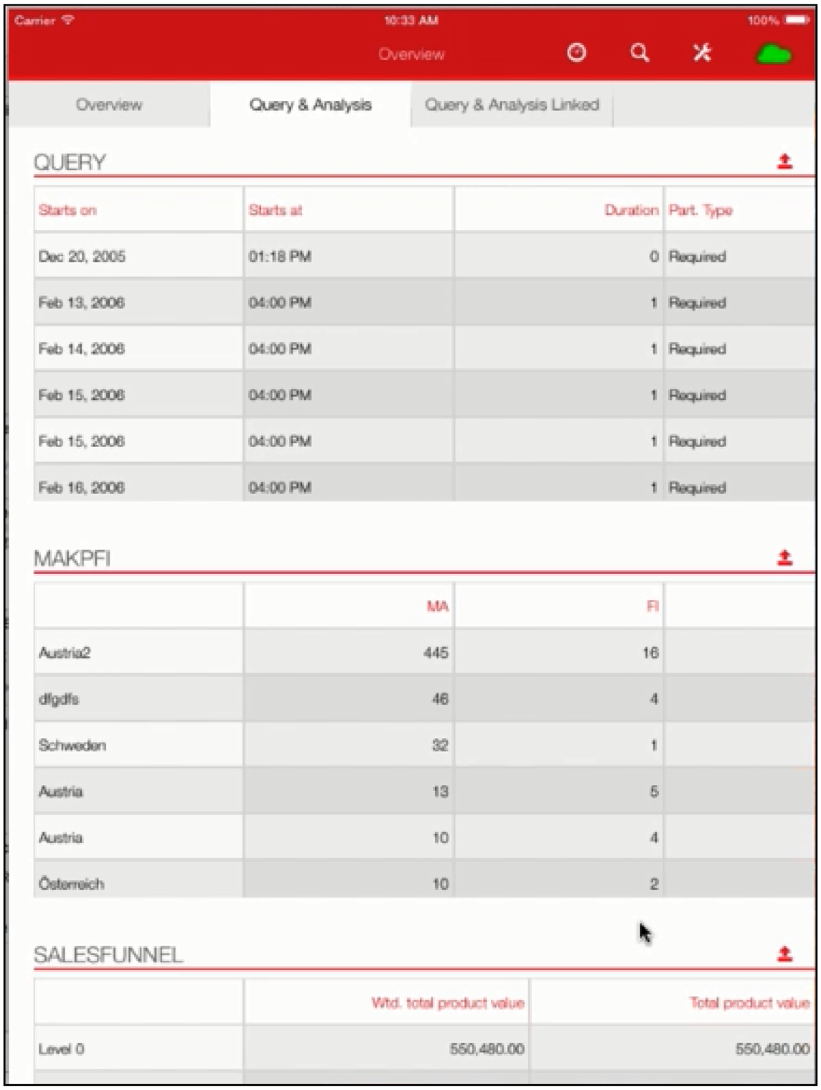
   The following screenshot shows a fully configured legacy form called **start page** for the CRM.pad application. Three tabs are defined in this configuration: **Overview**; **Queries and Analyses**; and **Queries and Analyses Linked**. The subsequent screenshots shows the corresponding start page with the corresponding tabs open, in the CRM.pad application running on an iPad.


Legacy Form 'StartPage'
BC | update.tablet | CRMpad KIF Tests | CRMpad UnitTest | UNITTEST_VITOR
Startpage
Legacy Forms
Legacy Form 'StartPage'
Default Textgroup:
net_mask_text
Hook-Function:
Label:
Width:
Height:
Tab#1
Text:
Overview
Attr:
Options:
Row-Number
Row-Attributes
Processes
Row #1
Type
Valuename
Text Func
Span Hook
Flags TD-Attributes
Item-Attributes
Querystring
Options
XML-Path
InsightBoard
APPMENU
RecordList
Row #2
DatePicker
Calendar
$curDay
Row #3
RecordList
My Activities this Week
Row #4
Map
Map of My Activities
{"Height":350}
Row #5
FI;;
{"MaxResults"
Row #6
CP;;
Row #7
MA;;
New Row
Tab#2
Query & Analysis
Query
MAKPFIAnalyse
Analysis
MAKPFI
SalesFunnel
Tab#3
Query & Analysis Linked
FIRecord
FI
MAKPFI;$FIRecord
New Tab
Save
Delete
Back
© 1988 - 2016 update Software AG, www.update.com

The following screenshot shows the corresponding tabs defined in the designer above displayed in the CRM.pad application. This screenshot shows the **Overview** tab. It is displaying the records as defined in the rows for the **Overview** tab in the designer.



The following screenshot shows the **Queries and Analysis** tab. It is displaying the **Query** and the two Analyses **MAKPFI** and **SALESFUNNEL** as defined in the designer.



5

Enterprise Features

Enterprise features are enabled if you have an Enterprise license for Aurea CRM.

Below topics describes enterprise features.

Google Analytics Integration

Learn how to integrate google analytics on apple device.

CRM.pad application running on Apple devices is integrated with Google Analytics. Using Google analytics you can track and understand the usage of the CRM.pad application by your users. This can provide you with valuable insights into your business as well as measure the results of your marketing campaigns, behavioural flow patterns of CRM.pad users, help develop KPIs by collecting data ratios and counts.

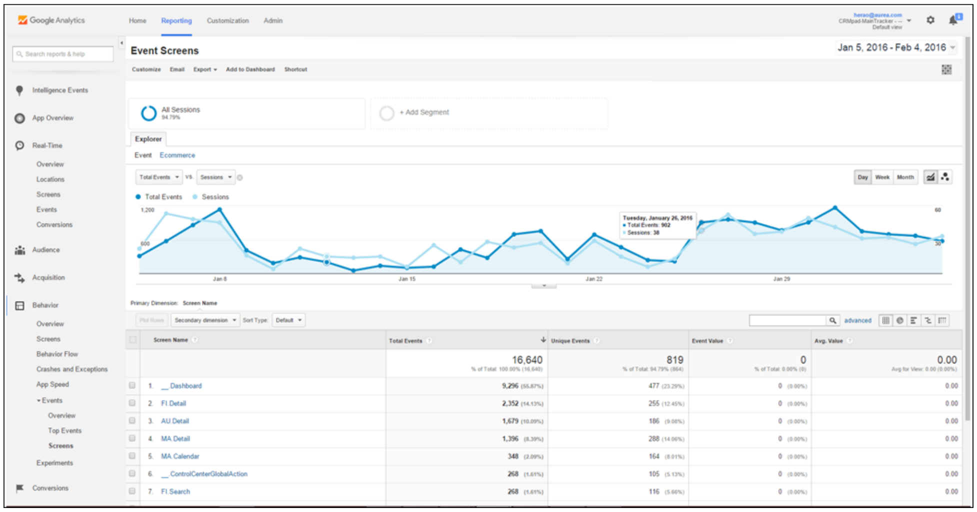
Some of the key information that can be extracted by the Google Analytics integration are listed below:

- Usage frequency of the CRM.pad application.
- The version of the CRM.pad application being used
- The devices and their versions used to access the CRM.pad application
- The ratio of crashes of the application versus the iOS version used
- Average time spent on a screen by users
- Behaviour flows that show how users move to their target screens
- Real time information on events triggered by users

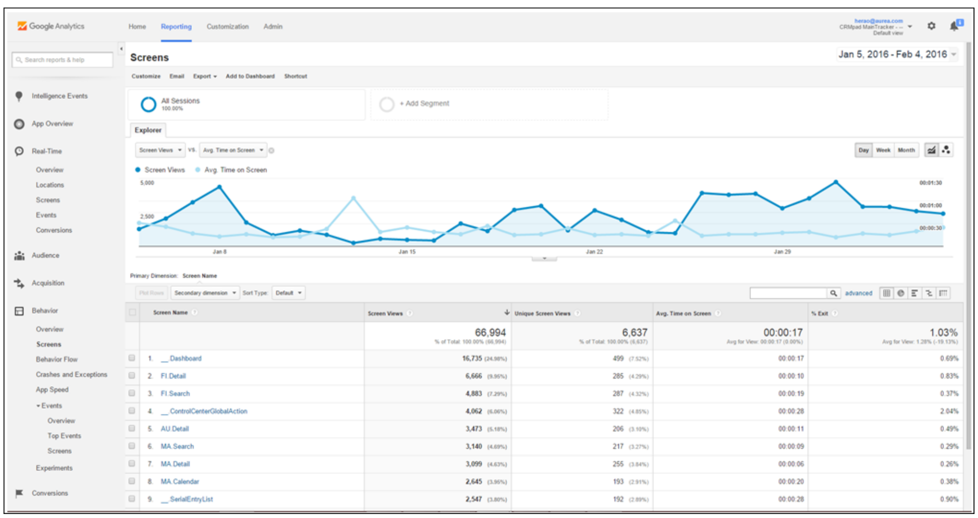
Note: Enterprise users can use their own Google Analytics tracker and can further customize the tracked components. Standard users of Aurea CRM can use Google Analytics tracker provided by Aurea.

The following screenshots show the metrics captured on screen events, user sessions, and screen behavior flows of the CRM.pad users by Google analytics.

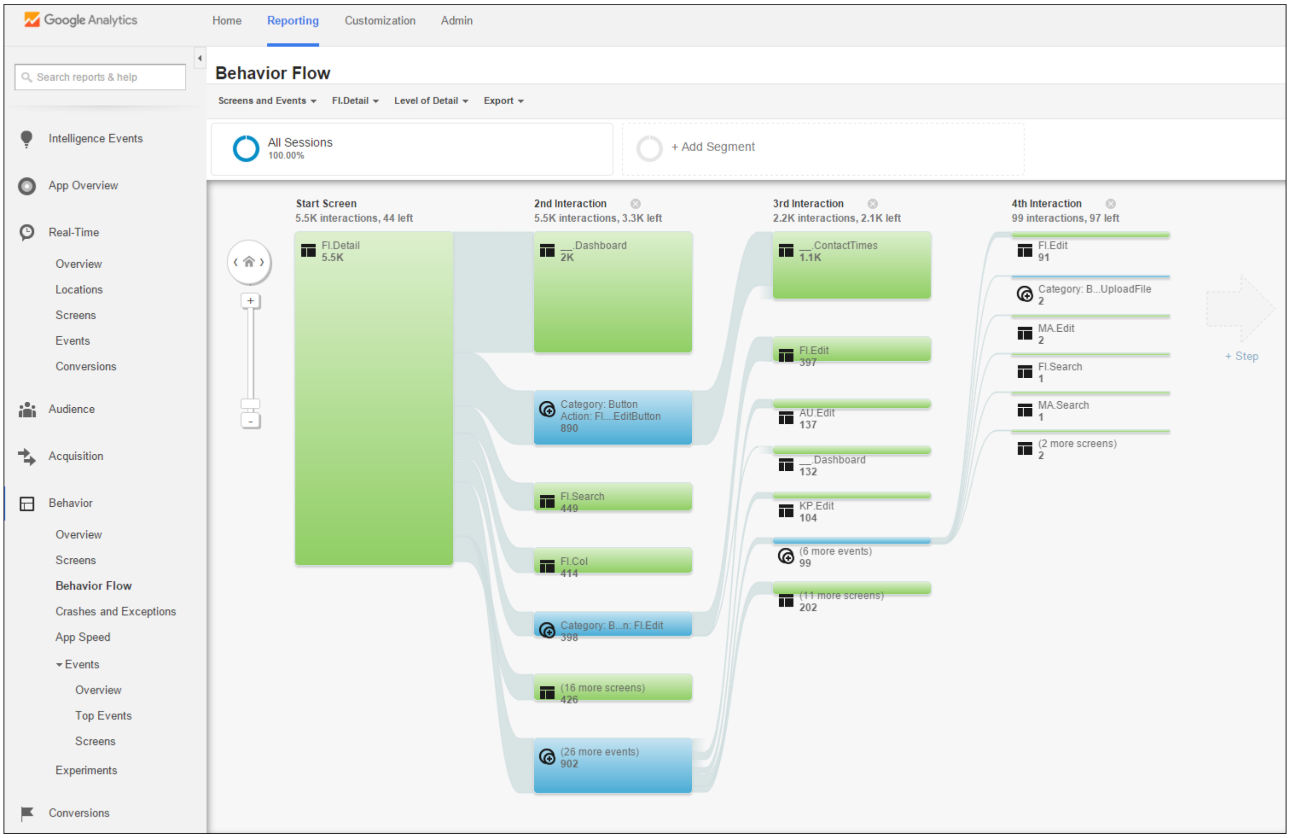
Total number of screen events Vs the total number of user sessions



Total number of Screens Viewed Vs Average time spent on a screen



Behaviour flow showing screens and action events for company detail



Configuring the Google Analytics Tracker

To track the CRM.pad application using Google Analytics, you can:

1. Use the main tracker provided by Aurea CRM. This tracks all aspects of CRM.pad application. Use the `GoogleAnalytics.MainTrackerEnabled` parameter to disable tracking by Google Analytics tracker provided by Aurea. Tracking for CRM.pad application is enabled by default.
2. Enterprise customers can use their own Google Analytics tracker and they can specify which of the menus, screens, and buttons of the CRM.pad application can and cannot be tracked.

To use your own Google Analytics tracker (enterprise customers), perform the following steps:

1. Disable tracking by Google Analytics tracker provided by Aurea by setting the parameter `GoogleAnalytics.MainTrackerEnabled` to `false`.
2. Set the value of your google analytics tracker id using the `GoogleAnalytics.CustomerTrackerId` web parameter in the designer.
3. Set tracking controls on the screens, menus and buttons of CRM.pad application by adding them to the white list or black list web configuration parameters. For more details on Google Analytics web configuration parameters, see Google Analytics Parameters.

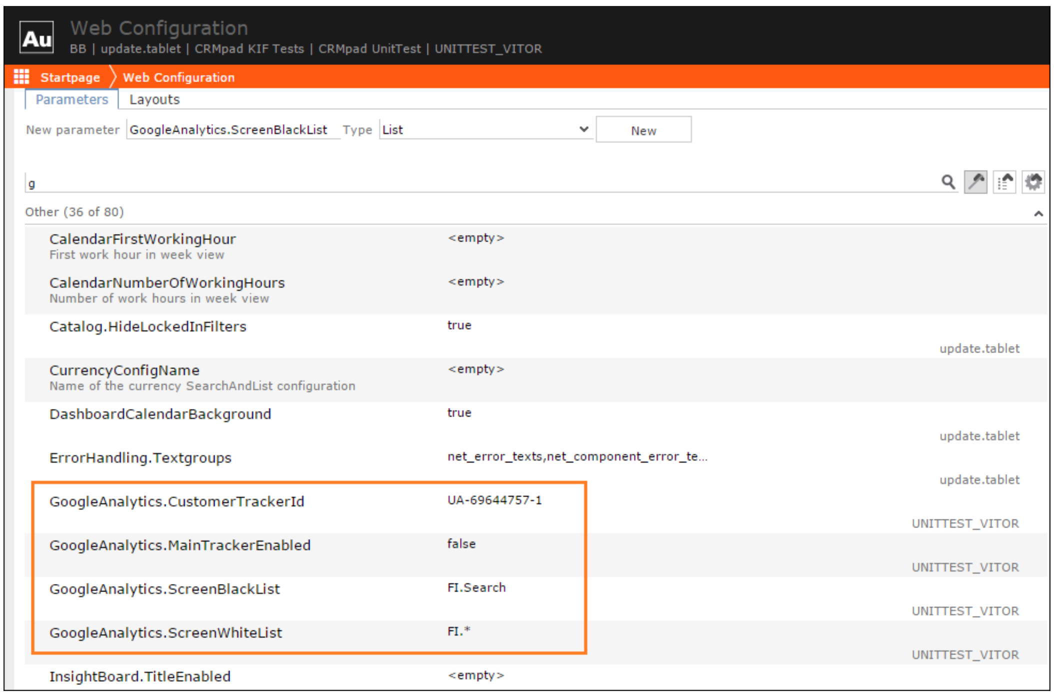
Consider the following examples:

a. Adding FI.Search and MA.Edit to the parameter `GoogleAnalytics.ScreenBlack-List` ensures that all screens are tracked except the Search Company and Edit Activity screens.

b. Adding FI.* to the parameter `GoogleAnalytics.ScreenWhiteList` ensures that only the screens from Company info area is tracked.

c. Now, adding FI.Search to the `GoogleAnalytics.ScreenBlackList` ensures that only the Company search screen is not tracked, while the condition defined in point 2 ensures that all the other screens from the company info area area are tracked.

A sample configuration screenshot is provided below:



MobileIron Support

MobileIron is a leading provider of Mobile Device Management (MDM) software and solutions.

MobileIron integration with CRM.pad application

MobileIron solution provides security for CRM.pad application data when the user is mobile or at rest. It also provides data loss prevention control.

MobileIron provides the following advantages:

- Security & Usability improvement

CRM.pad application can run automatically using secure VPN (appconnect). Users do not have to manually configure VPN connection, for example using Cisco AnyConnect.

- AppConnect containerization technology

  The CRM.pad application runs in a container, which allows exchange between multiple business applications like CRM.pad, notes, email, and data using the connected container. This also reduces your total cost of ownership.

- Allows you to control certain functions the user can perform. For example, Copy & Paste, Print, and Open In other applications.

  This allows you to control if the user can print CRM.pad data, or open the data in other related applications. This is very useful in restricted environments.

- Selective wipe of CRM.pad application

  This is very useful if you are adapting BYOD strategy and allowing your users to bring their own devices.

If you already use CRM.pad application and want to integrate it with MobileIron, perform the following:

1. Get the unsigned application archive file from Aurea.
2. Use your Apple Enterprise License to sign the archive.
3. Use the AppConnect wrapper for iOS to wrap the application archive.

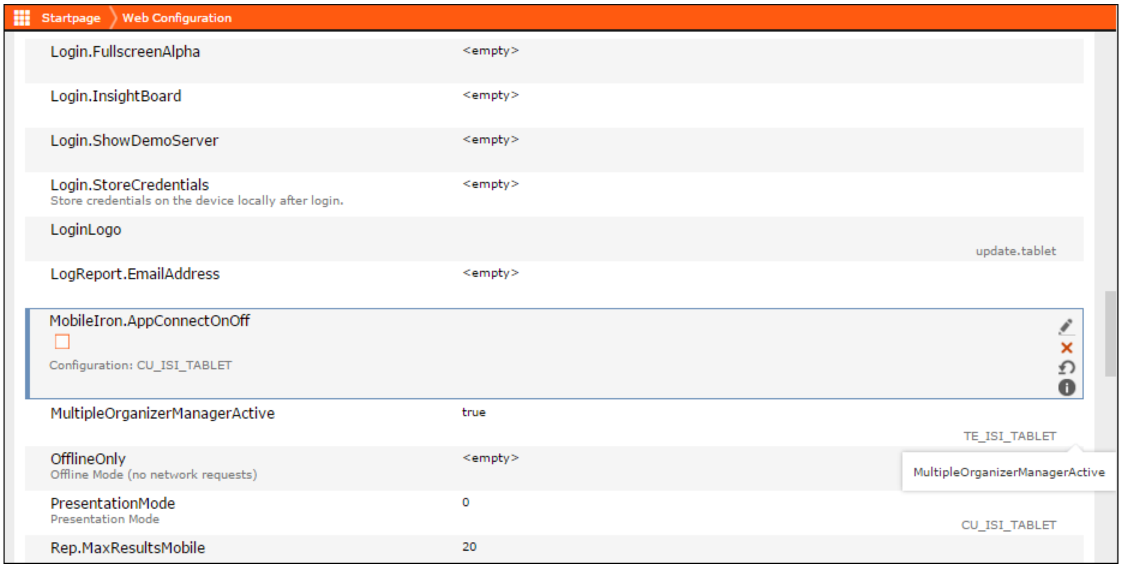
New users can install CRM.pad application version with the AppConnect integrated from the AppStore. The AppConnect integrated version detects if MobileIron is used and uses it to securely communicate with the CRM server.

Note: If your users are using MobileIron Sentry, then append `enablemobileiron=true` the URL to ensure that you are able to connect to the CRM Server. This ensures that MobileIron is started before the connection with CRM Server is initiated. Without this parameter the MobileIron Sentry blocks CRM.pad from connecting to the CRM Server. This configuration is valid only for CRM.Pad package released by Aurea with the AppConnect integration.

Integrating CRM Server with MobileIron

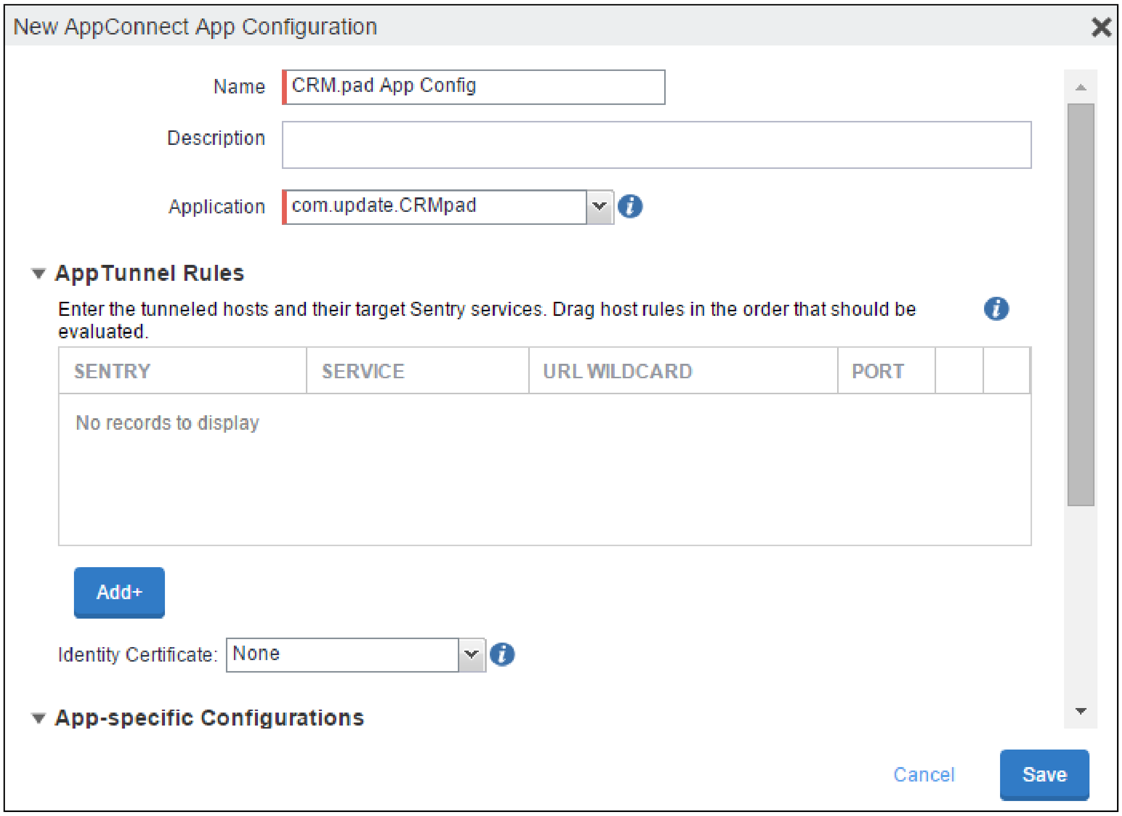
To integrate MobileIron with the CRM Server, perform the following:

1. Enable **MobileIron.AppConnectOnOff** web configuration parameter for CRM.pad designer configuration. To add the parameter and then enable it, perform the following:
   a. Login to the CRM.designer.
   b. Select **CRM.pad** from the **Application** dropdown list and select the appropriate **Configuration** to configure.
   c. Click **Web Configuration** from the **Configuration** menu.
   d. Add the **MobileIron.AppConnectOnOff** parameter and click the checkbox to enable it.



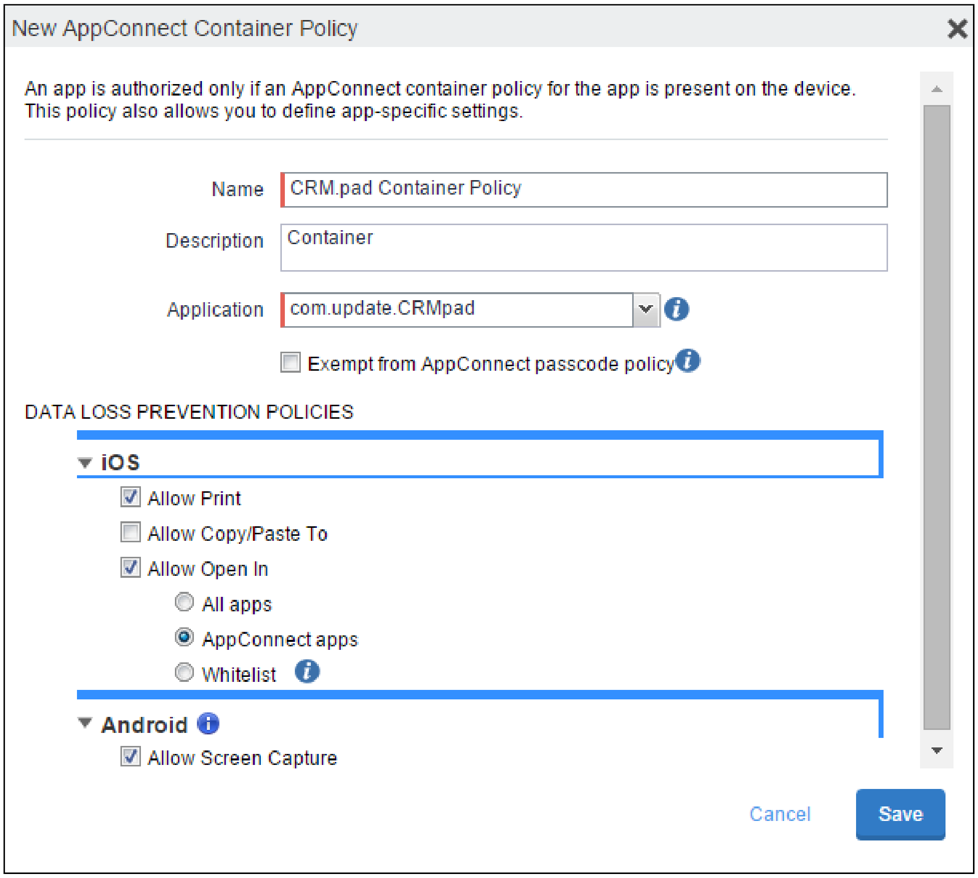
2. Click **Policies and Configs** and set up CRM.pad application configuration and container policy. Perform the following steps:

   a. To set the App configuration, from the **Add New** dropdown list, select **AppConnect** > **App Configuration**. The **New AppConnect App Configuration** window opens. Enter a **Name** for the application and enter `com.update.CRMpad` in the **Application** field. Provide other information as necessary and click **Save**.



b. To set the App container policy:

   a. From the **Add New** dropdown list, select **AppConnect** > **Container Policy**. The New AppConnect Container Policy window displays.

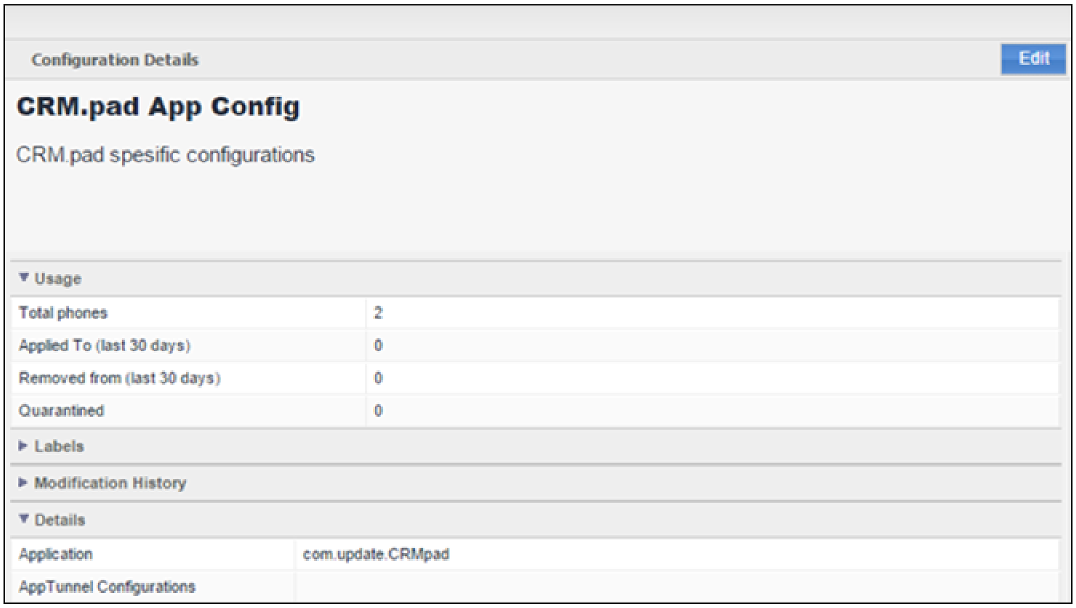


b. Enter a **Name** for the policy and enter `com.update.CRMpad` in the **Application** field.

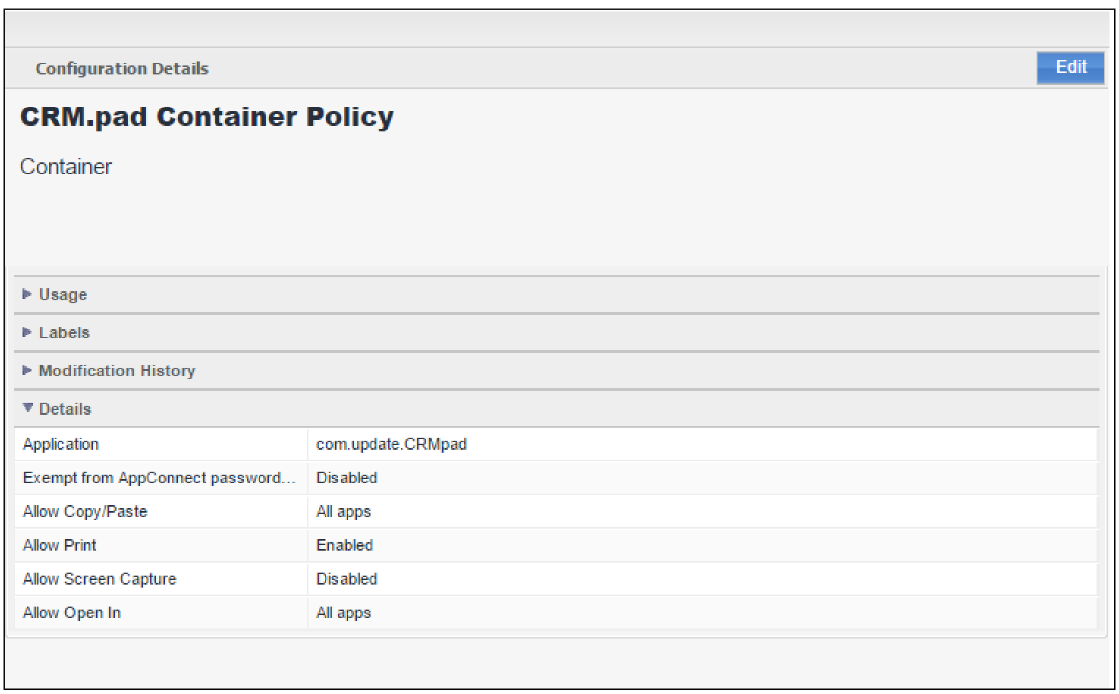
c. You can set the data loss prevention Policies by setting the appropriate permissions for CRM.pad data. For example, selct the **Allow Open In** checkbox, and select **AppConnect apps** or you can select **Whiltelist** and provide a list of applications that can open CRM.pad data.

c. You can view the details of a configuration or policy by click the desired item from the list. The details are listed in the right side of the window.

The following screenshots show the app configuration properties. You can expand the listed property to view its details:



The following screenshot shows the details of the App container policy:



Spotlight Search Support

Spotlight search allows you to search for and access information from within your app.

CRM.pad supports the Spotlight search introduced in IOS 9.0. It allows you to access activities and contents deep within your Aurea CRM application.

CRM.designer allows you to configure the following:

- The content to be indexed
- The content that can be searched
- The event to be triggered when a search result is being selected.

The following types of indexes are made available by IOS 9:

- On-device index – provides access to the private data present on your device.
- Server-side index – stores only publicly available data that you've marked appropriately on your website.

Note: At present only On-device index is supported.

CRM.pad does not support spotlight search on all apple devices. Refer to the table below for supported devices.

Note: You have to upgrade to iOS9.0 to be able to use this search feature.

Model	Initial Operating System	Support status
iPad Mini (first gen)	`iOS 6`	Not supported
iPad Mini 2	`iOS 7`	Supported
iPad Mini 3	`iOS 8.1`	Supported
iPad Mini 4	`iOS 9.0`	Supported
iPad (first generation)	`iOS 3.2`	Not supported
iPad 2	`iOS 4.2.1`	Not supported
iPad 3	`iOS 5.1`	Not supported

Model	Initial Operating System	Support status
iPad 4	`iOS 6.0`	Supported
iPad Air	`iOS 7.0.3`	Supported
iPad Air 2	`iOS 8.1`	Supported

CRM.designer Spotlight configuration parameters

The CRM.pad application uses the NSUserActivity API and CoreSpotlight API provided by iOS 9 to configure search and ensure that the CRM.pad information is searchable on your device.

The following web configuration parameters influence CRM.pad. Information on how to set parameter values can be found in the Web Configuration topic of the Aurea CRM web Administrator Guide.

Following web configuration parameters need to be defined in the CRM.designer:

SpotlightSearch.ConfigOnOff

The fields are not searchable. This parameter is ignored under iOS6.

Enables or disables the searchability of data with this parameter. Set the value to “True” to make data searchable outside the CRM.pad application.

Type: checkbox

Default: Off

SpotlightSearch.FieldGroups

Use this to specify the field groups that are searchable. Multiple fieldgroups can be separated by a comma. For example, for infoarea “KP” give a name “KP:spotlight”, and for “FI” give the name “FI:spotlight”. The value field is provided as a comma separated values: “KP:spotlight, FI:spotlight”.

Type: Character

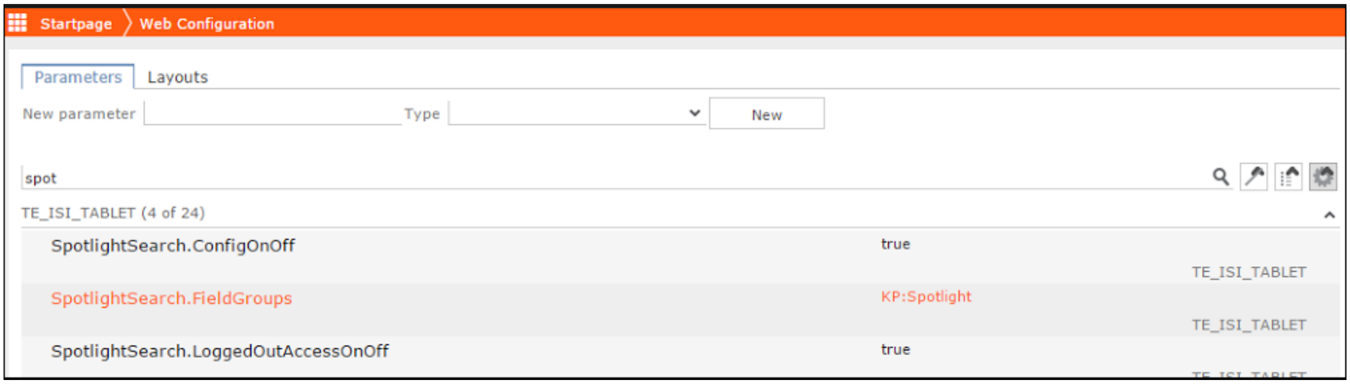
SpotlightSearch.LoggedOutAccessOnOff

Enables or disables access to CRM.pad application data even when a user is not logged in. Set it to “True” to enable such access to data.

Type: Checkbox

Default: False

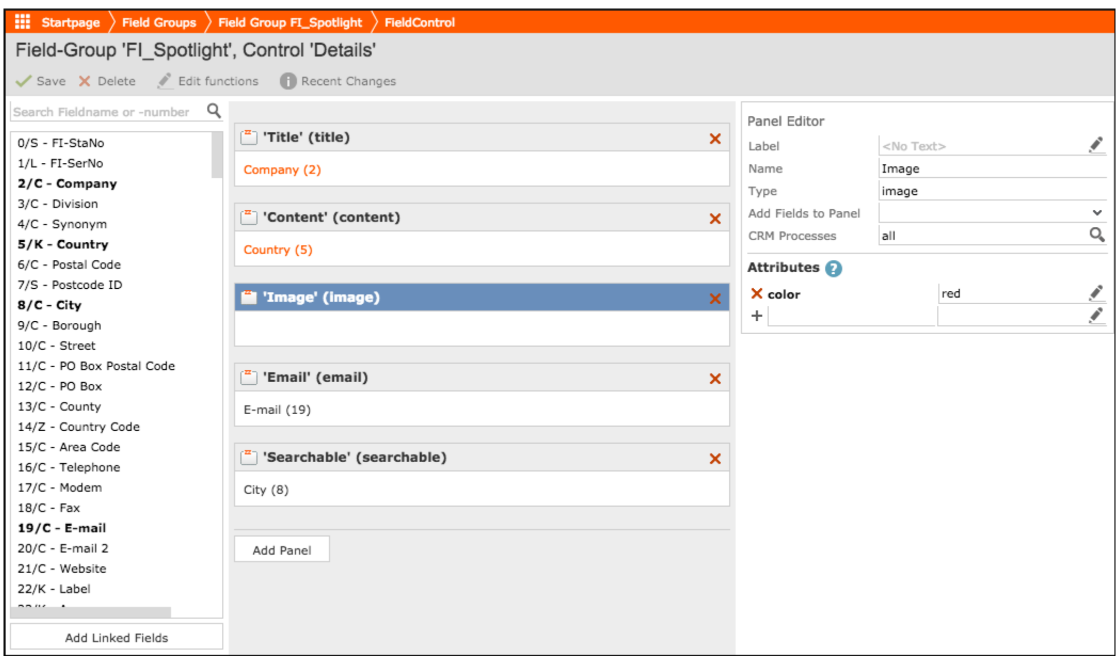
The screenshot below shows a sample configuration.



Example – Configuring a FieldGroup to be searchable

In this example configuration a “FI_Spotlight” field group is created with a “Details” control for each information area that is to be searched.

See the screenshot below:



To configure the search you have to perform the following steps:

1. For each information area that is to be searched, create a field group with “Details” control.
2. Define the following panel types:
    - **Title** panel -
    - **Content** panel

Note: To have a valid configuration you should atleast define the Title and Content panels and ensure that they contain atleast one field in each.

After adding these panels, synchronize and check. You need to have an Enterprise license for this feature to work.

- **Image** panel
- **Email** panel (optional)
- **Searchable** panel (optional)

Note: The Email panel is currently not supported and we are awaiting Apple's support for local index implementation.

The following are the descriptions of the different panels:

Title panel

Description: Search result title area fields are located in this panel.

Type: Set the "Type Identifier" to "title".

Content panel

Description: Search result content area fields are defined with content panel.

Type: Set the "Type Identifier" to "content".

Note: To have a valid configuration you should atleast define the **Title** and **Content** panels and ensure that they contain atleast one field in each.

After adding these panels, synchronize and check. You need to have an Enterprise license for this feature to work.

Image panel

Description: You can define thumbnail images and their attributes in this panel. It is an optional panel and allows you to define the following thumbnail images:

- Record photo/logo: Defined as field in the panel.
- App Icon Color: Defined as "color" attribute for panel with 5 system colors or hex color codes.
- Field group icon: Defined as "image" attribute for panel configured by name of the image defined in images area and copied to resources folder.

Type: Set the "Type Identifier" to "image".

Email panel

Description: Email data is retrieved from the field groups define in this panel. This is an optional panel. It is possible to combine all rich result parameters as one panel and change the panel identifier to “solution”.

Type: Set the “Type Identifier” to “email”.

Searchable panel

Description: This panel provides additional searchable fields, which is not visible in title or content area.

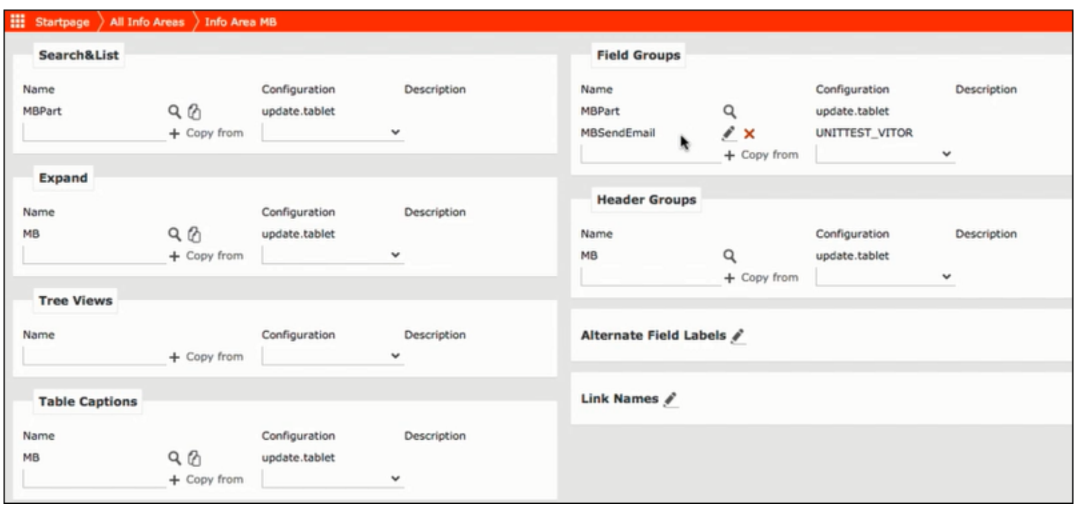
Type: Set the “Type Identifier” to “searchable”.

Sending Email to Multiple Activity Participants

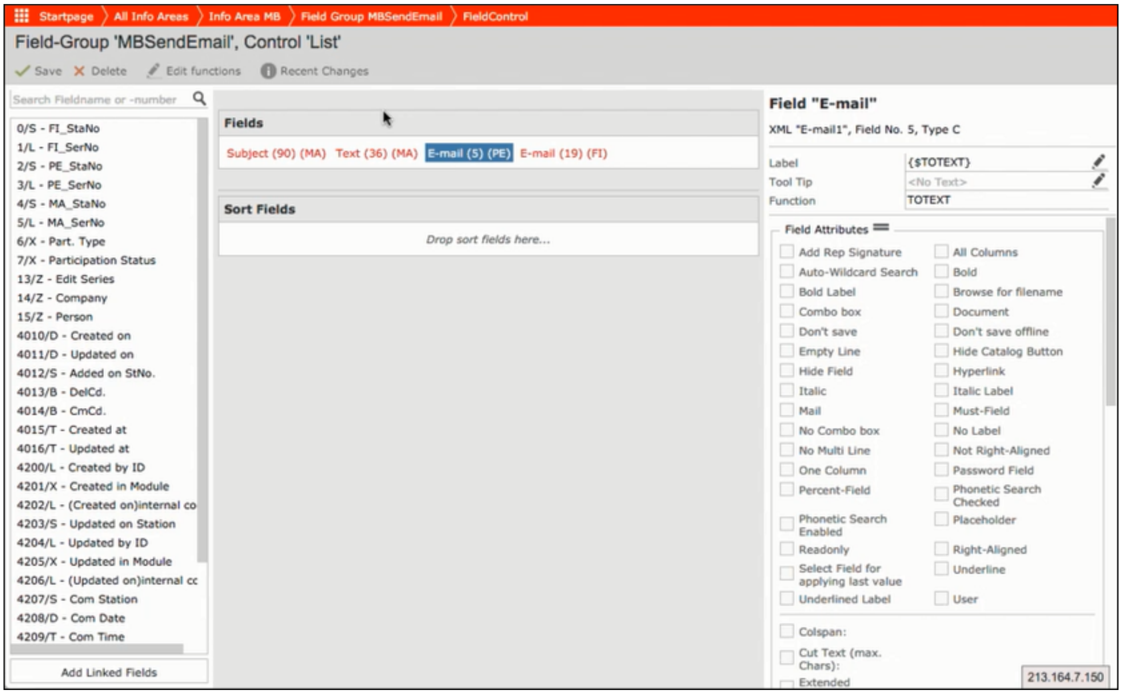
Enterprise customers can allow their users to send an email to multiple Activity participants.

To configure this, perform the following steps:

1. In Start page of the CRM.designer configuration for the CRM.pad application, click **Action Templates**.
2. Copy the **ClientEmail** action template to the configuration. For more information on this action template, see ClientEmail.
3. Create a field group in the Activity Participant (MB) infoarea. For example, the following screenshot shows the **MBSendEmail** field group added to the Activity Participant (MB) infoarea.



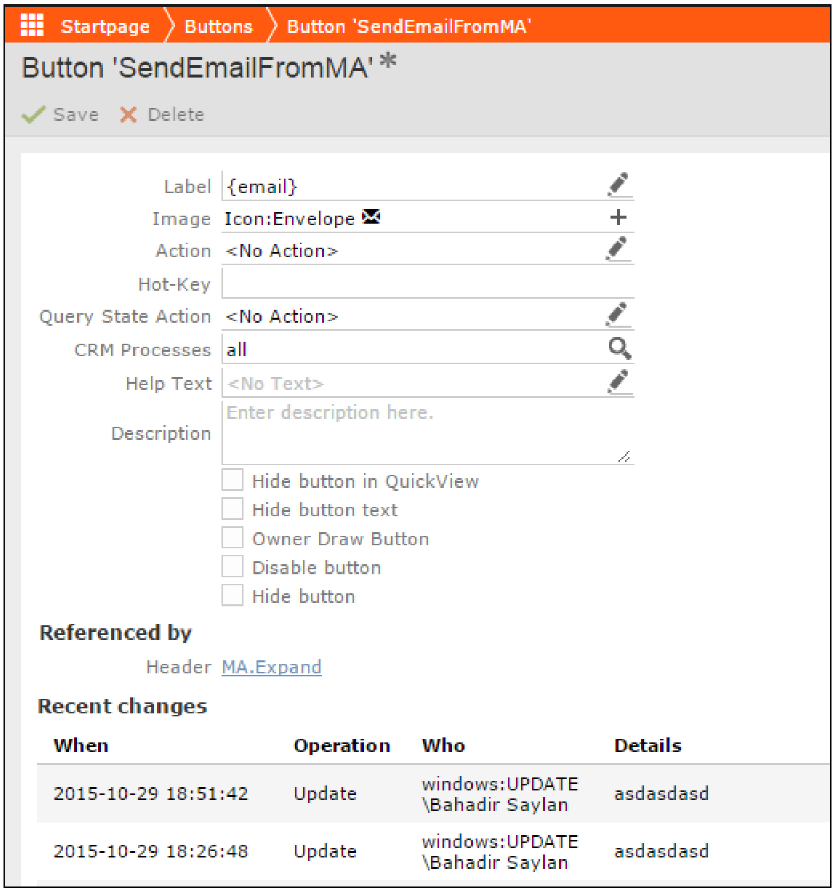
4. Edit the **MBSendEmail** field group and add a List control to the field group.
5. Edit the List control and add the linked e-mail fields to the list control. For more information on the email fields and the pre-defined function names that can be used for each field, see Defining E-Mail Fields.



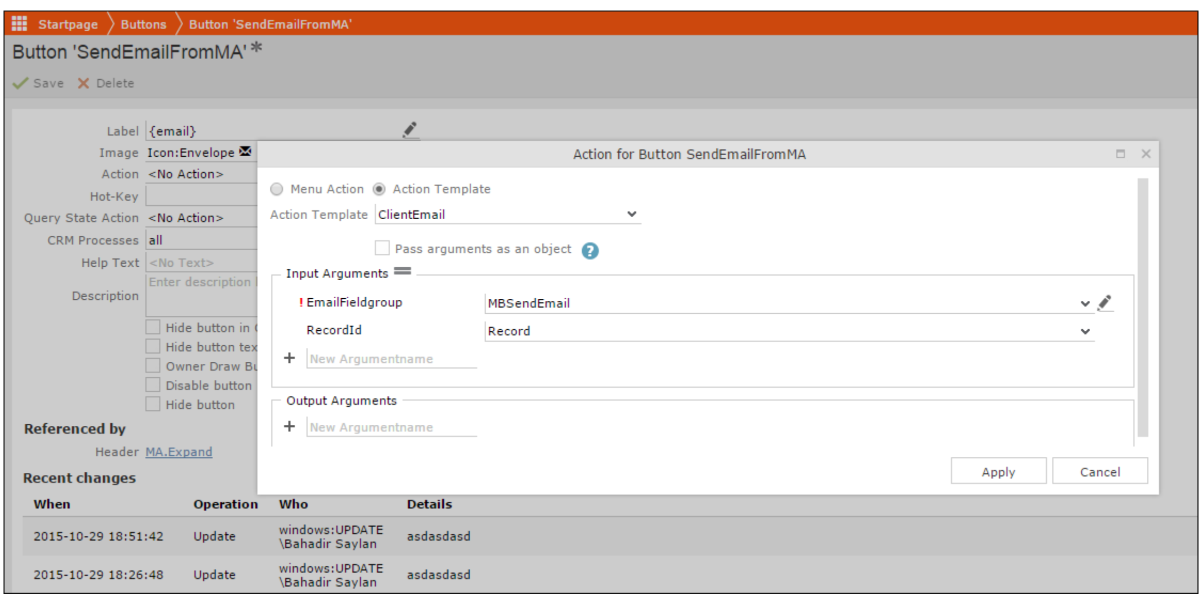
Note: You can also add the linked email field from the Company (F1) info area using the company field. This adds the company email id for the activity.

---

6. Create a button for the email action that is visible in the user when he logs into Aurea.CRM from his iPad. The following screenshot shows the configuration window for the **SendEmailFromMA** button added to the Expand Header group of the Activity Infoarea.



7. Edit the **Action** field to open the “Action for Button SendEmailFromMA” window.



Perform the following steps:

a. Select the **Action Template** option.
b. Select **ClientEmail** from the drop-down list for the **Action Template** field.
c. Select **MBSendEmail** from the drop-down list for the **EmailFieldGroup** field.

Note: All activity participants are searched and can be added to the Activity email.

d. Select **Record** from the drop-down list for the **RecordId** field.
e. Click **Apply**.

8. Click Save to complete the configuration for multiple email recipients.

Note: If you do not have an Enterprise license, this feature can still be configured but it does not work. Enterprise users can send an Email to multiple users or a single user.

6

Additional Information

Explore additional capabilities offered by Aurea CRM.pad solution.

CRM.Intelligence

CRM.intelligence is an analysis tool based on the QlikView technology.

You can define that users can access CRM.intelligence either

- from a context menu
- from a panel in an info area

To add CRM.intelligence:

1. Define either a context or application menu action and assign the `QlikView` action see QlikView on page 113.
2. If you want to add it to a panel, you need to add a panel to the desired info area control and assign the `WEBCONTENT_<ContextMenuName>` type, where you must replace `<ContextMenuName>` with the name of the context menu you defined previously.

Synchronizing Favourites and Related Data

Records that are added to Favorites can be synchronized along with their related data (linked records).

For example, if Company and Person infoarea records are added to Favorites and the Appointments infoarea records are linked to the Company and Person records, then the linked Appointments records are also synchronized along with the Company and Person records.

Note: Some linked records may not be synchronized if there are custom filters defined on the record. .

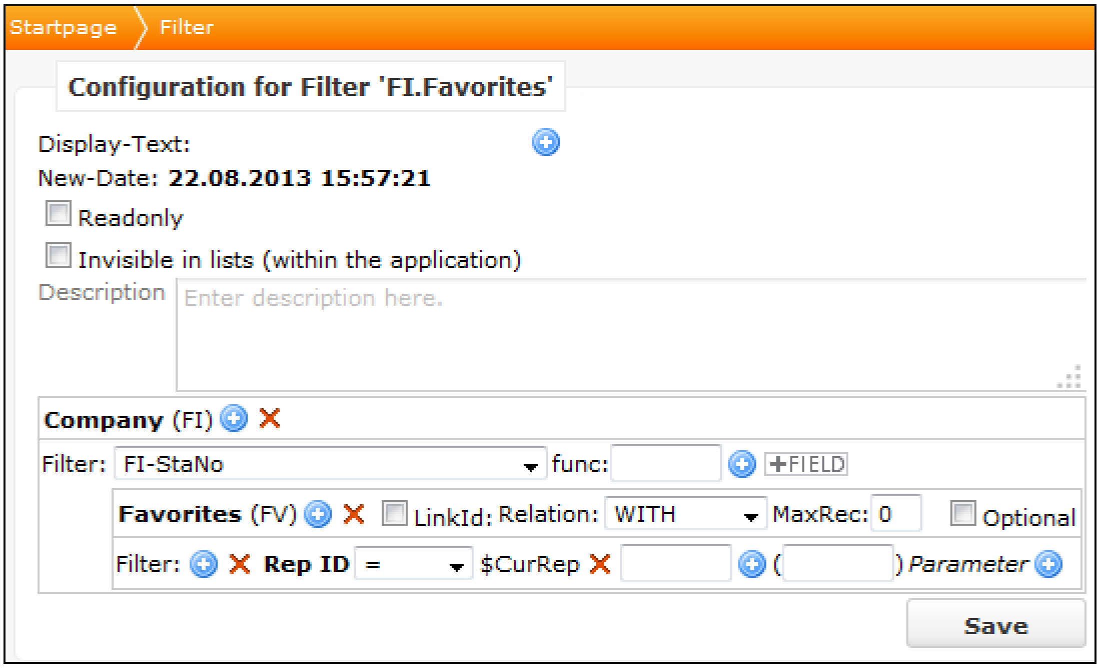
Example:

Companies and persons that were defined as Favorites and appointments linked to these records are synchronized.

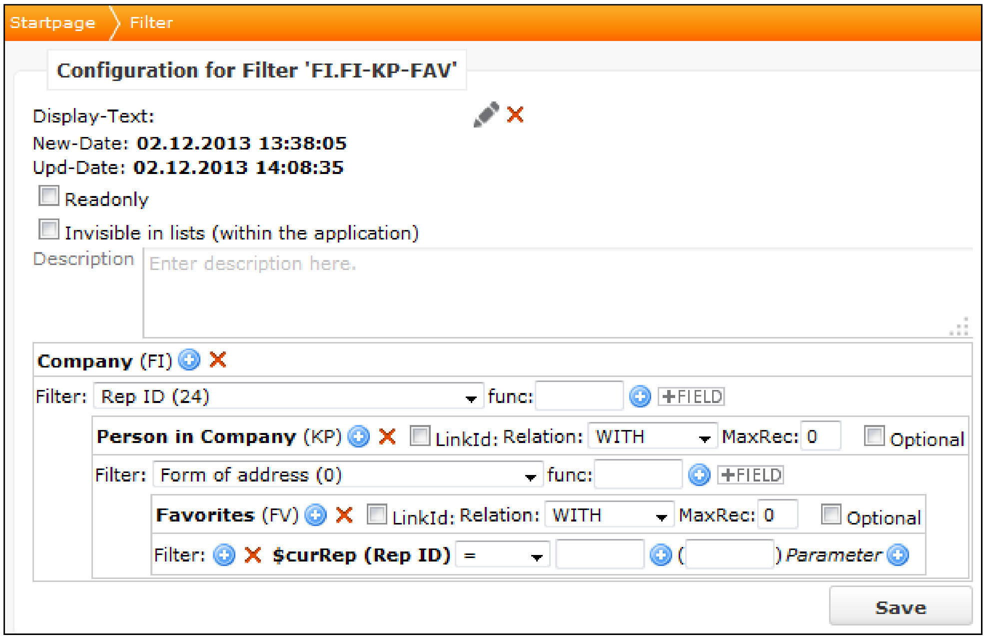
To configure this example:

1. Define the following filters for the FI info area:

   - `FI.Favorites` (included in update.tablet): Filters all companies that the user has defined as Favorites:

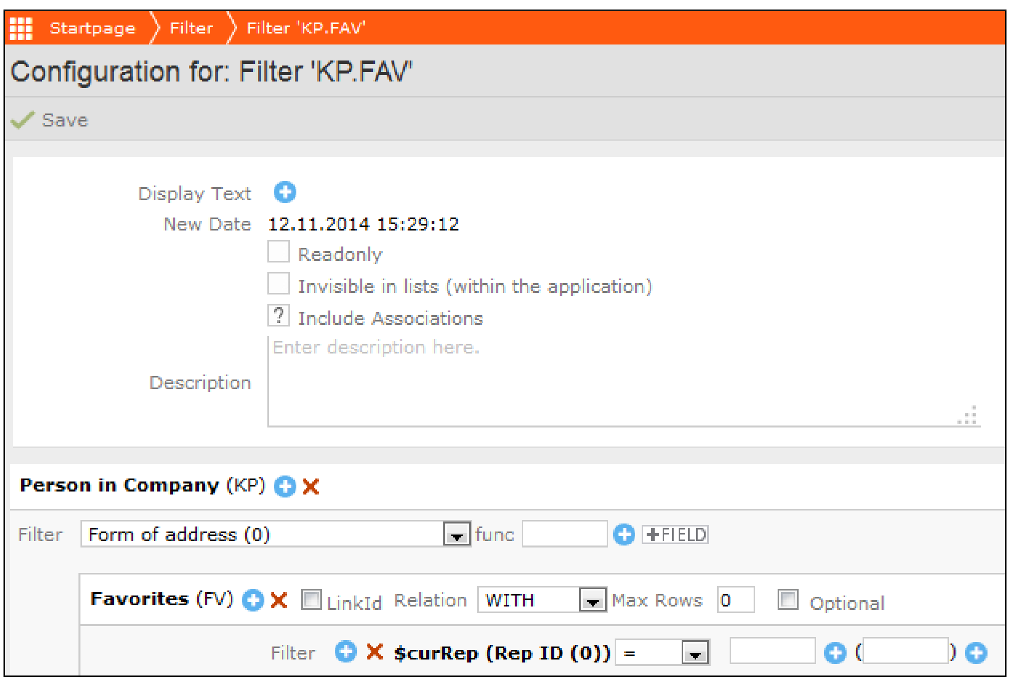


   - `FI.FI-KP-FAV`: Filters all companies related to all persons that were defined as Favorites:

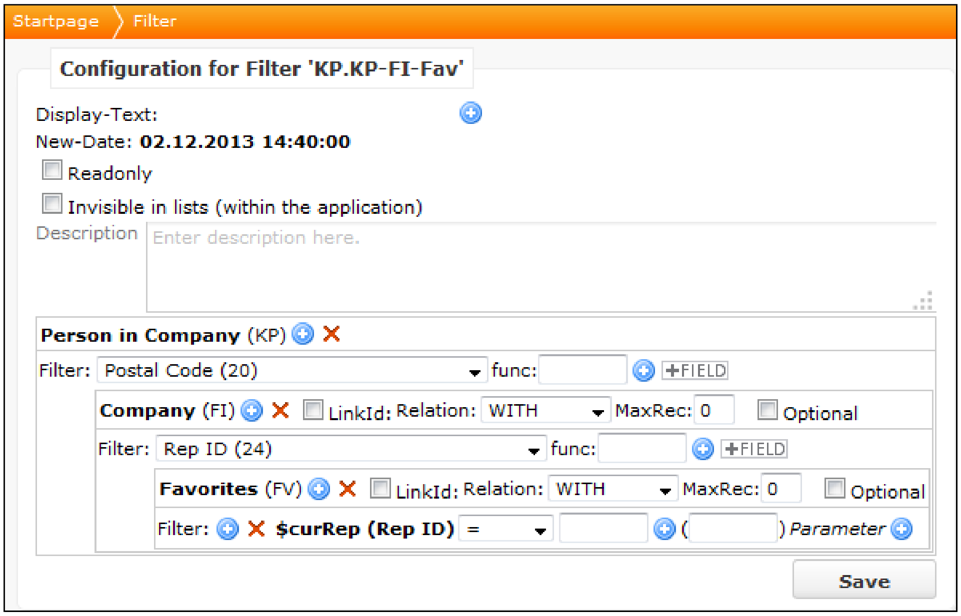


2. Define the following filters for the KP info area:

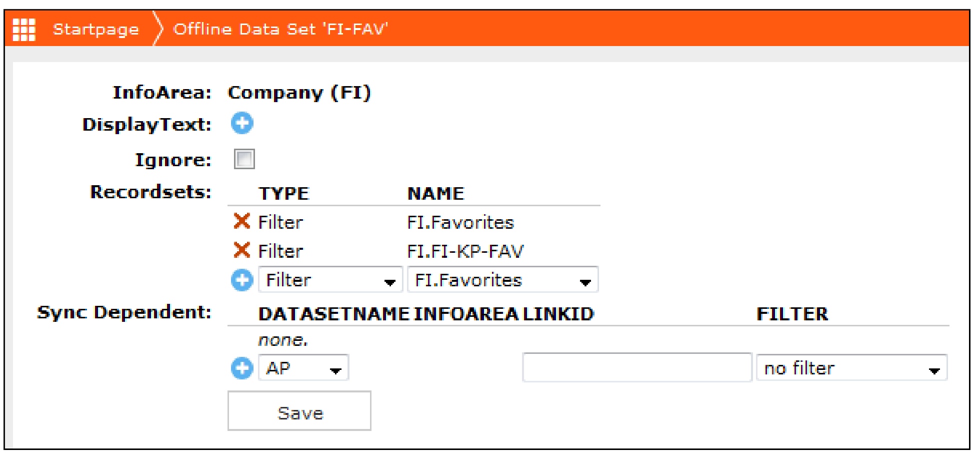
- `KP.FAV`: Filters all Person (KP) records that were defined as Favorites.



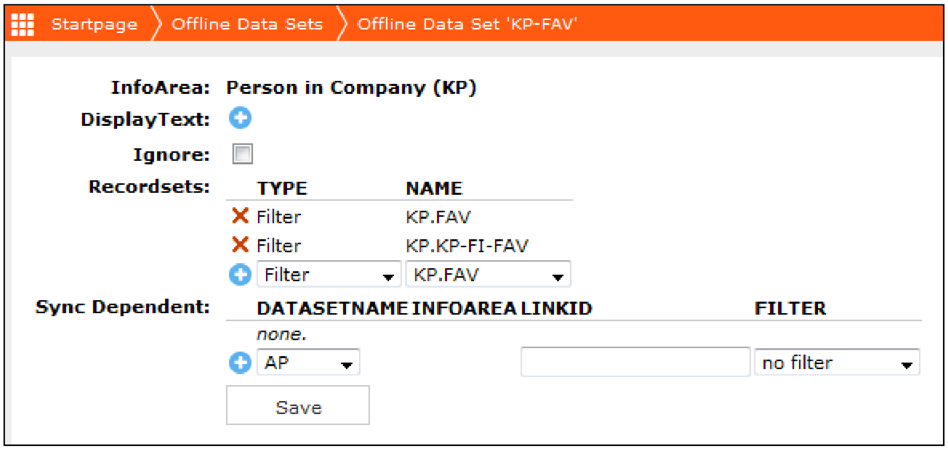
- `KP.KP-FI-FAV`: Filters all persons not defined as Favorites that are linked to a company in the Favorites.



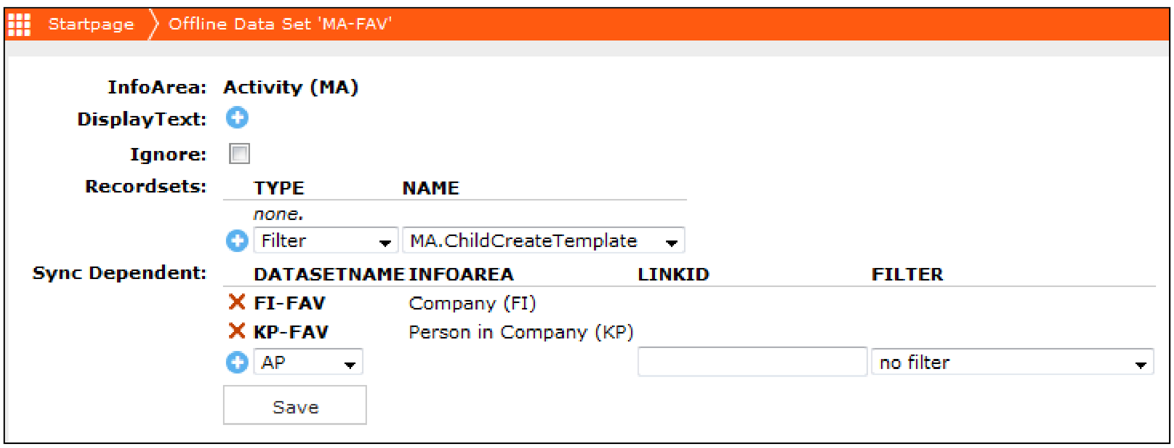
3. Define the following offline data sets:
   - For the FI info area:



   - For the KP info area:



- For the MA Info area (if you want to synchronize the activities of the companies and persons included in the Favorites):



Executing Server Workflows with Parameters

Learn how to execute workflows with parameters.

Two form types are available for Legacy Forms to define the execution of server workflows:

- `ActionForm`, see Defining the Parameters on page 368.
- `TriggerExecution`, see Executing Server Workflows on page 368.

Defining the Parameters

If you assign the `ActionForm` type to a row in a Legacy Form, a panel containing the configured Edit control is displayed.

Define the following options for this legacy form type:

- **Valuename**: Name of another form item that is used for entering data.
- **Func**: You can specify the following (separated by semicolon):
  - Name of the field group containing the desired Edit control (required). Enter `<fieldgroup>_<TabId>` to display a specific panel instead of the first one.

    The Edit control is used to display an edit mask and to define the fields (and their **Function** names). The function names given are used to transfer data to other controls.
  - Name of a Template filter used to apply predefined values (optional).

Executing Server Workflows

If you assign the `TriggerExecution` type to a row in a Legacy Form, a group containing a text, an optional description of the action and a button to start the execution of the action are displayed.

The button is not displayed if a field defined as **Must Field** is left empty.

Define the following options for this legacy form type:

- **ValueName**: Name of the form row (can be referenced in other form items).
- **Func**: You can define the following (separated by semicolon):
  - Source parameter (required), e. g. `$CopyDate` if the **ValueName** of the referenced `ActionForm` is `CopyDate`.
  - Record for which the action is executed (required), e. g. `$Record`.
  - Action type (`copyRecord` or `Workflow`) (required).
  - An additional configuration for the action (required if the `copyRecord` action type is assigned), either the workflow's name or a filter (if you use `CopyRecord`, see Copying Data Using a Trigger on page 370).

  Example: `$CopyDate;$Record;CopyRecord;PR.CopyFilter`
- **Text** (separated by semicolon): Five texts are used in the area that is displayed in the application (where the user can start the workflow executions):
  - First text: The header name
  - Second text: Text displayed on the button before starting the execution
  - Third text: Text displayed on the button during the execution

  - Fourth text: Text displayed on the button after the execution
  - Fifth text: Text describing the executed action.

  If less than five texts are given, the last one is always used as description.

- **Options**: If no additional configuration was given, the first record that was changed is transferred to the next dependent row in the form. You can define the result set with the following two options:
  - `{"OutputInfoAreaId":[infoAreaId]}`

    Example: `{"OutputInfoAreaId" :MA }` results in the first changed MA record being returned. It can be used by the next dependent row in the form.

  - `{"FinishedAction":"[ContextMenuActionName]"}`

    Example: `{"FinishedAction":"SHOWRECORD"}` results being passed to the given subsequent action (in this example to `SHOWRECORD`).

  - `{"OutputMultiple": true}`

    Example: `{"OutputInfoAreaId":"MA", "OutputMultiple": true}` results in all changed MA records being returned.

Viewing Results

Learn how to define how results are displayed.

To define that the results are displayed:

1. Add a new form row e.g. of type `RecordList`.
2. Enter the following options in the **Func** field, separated by semicolon:
   - Name of the Search&List configuration that references the List control to be displayed.
   - `$[Valuename]`: Specifying this option hands the result over to the given Search&List configuration.

   Example: The following configuration hands the result of the `$RecordCopy` function over to the Search&List called PR.

Copying Data Using a Trigger

If you want to copy data (i. e. handing data over to another info area), add a row of type `TriggerExecution` to the desired Legacy Form and enter `CopyRecords` in the **Func** column.

This results in a filter being used instead of a workflow. This filter defines which data of the records and their children is copied.

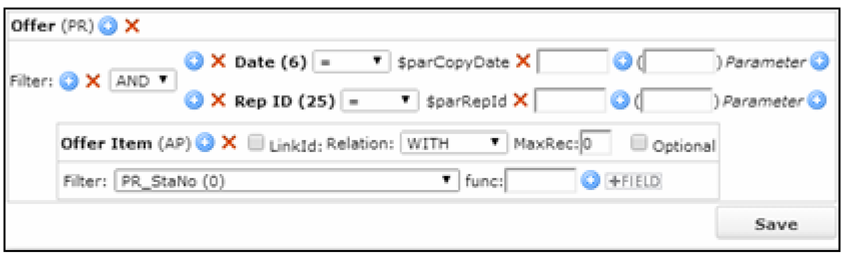
Copying Records from the Same Info Area

If the source and target record are from the same info area, the filter needs to define either predefined values to be applied or the child info area(s) where to you want to copy data.

Example:

Func parameter for `TriggerExecution`: `$CopyDate ;$ Record;CopyRecord;PR.CopyFilter`

`PR.CopyFilter`:



This filter defines that a new offer (PR) record is automatically added containing the following values:

- All fields that are specifically stated in the filter with the corresponding values
- All fields of the referenced List control (set with `Parameter :SourceConfig =[Search&List name]`, >> see the example in Copying Records from a Different Info Area).
- The links of all linked fields of the references List control.

A new AP (Offer Item) record is added for each offer item of the source (PR) record and the same rules are applied (i. e. the link to the new PR record is automatically set).

Note: If (like in the example above) no Search&List configuration was given (and therefore, no List control is referenced), the List controls with the same name than the info area (PR and AP in the example) are used. If you want to reference a different Search&List configuration (and its List control), use `Parameter :SourceConfig =<Search&List name<`, >> see the example in Copying Records from a Different Info Area).

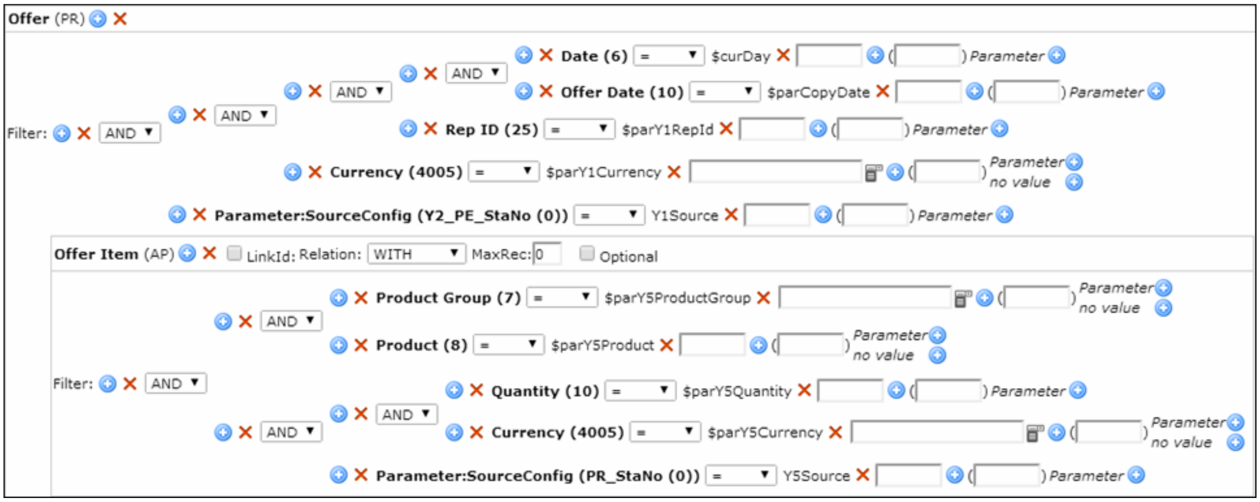
Copying Records from a Different Info Area

If the source record belongs to a different info area, ensure that the fields you want to copy are set in the filter.

Example:

Copying an opportunity (Y1) to an offer (PR):

Func parameter for TriggerExecution: `$CopyDate ;$ Record;CopyRecord;PR.CopyFromY1`



This filter defines that a new offer (PR) record is automatically added containing the following values:

- All fields that are specifically stated in the filter with the corresponding values
- The links of all linked fields of the referenced List control (set with `Parameter:SourceConfig=<Search&List name>`

A new AP record is created for each Y5 record of the copied Y1 record (not filtered by the Y5 source Search&List configuration) with the following values:

- All fields that are specifically stated in the filter with the corresponding values
- All links of link fields in the referenced List control
- The link to the parent records (in this example PR) is set automatically

Note: If you copy records from several several you must define a Search&List configuration using `Parameter :SourceConfig`.

Preventing Duplicates

Learn how to prevent duplicate requests.

If a request is handled several times, duplicates might occur. You can prevent duplicates as follows:

- `RequestControl`: Requests sent by CRM.pad are (automatically) sequentially numbered. If e.g. the network connection is interrupted and the information about a successfully processed request could not be sent to the client anymore, the server recognizes (based on the numbering) that this request was already processed.
- Key fields: If key fields are applied, the existence of these fields can be checked. If a record with the same key field values and the same link records already exists, the existing record is used.

  If you use key fields, you must define Template filters that apply the key property. You can assign these Template filters in the following input arguments of the `SerialEntry` action call. For further details, see SerialEntry on page 127.

  - `DestinationTemplateFilter`
  - `DestinationChildTemplateFilter`
  - `DestinationParentTemplateFilter`

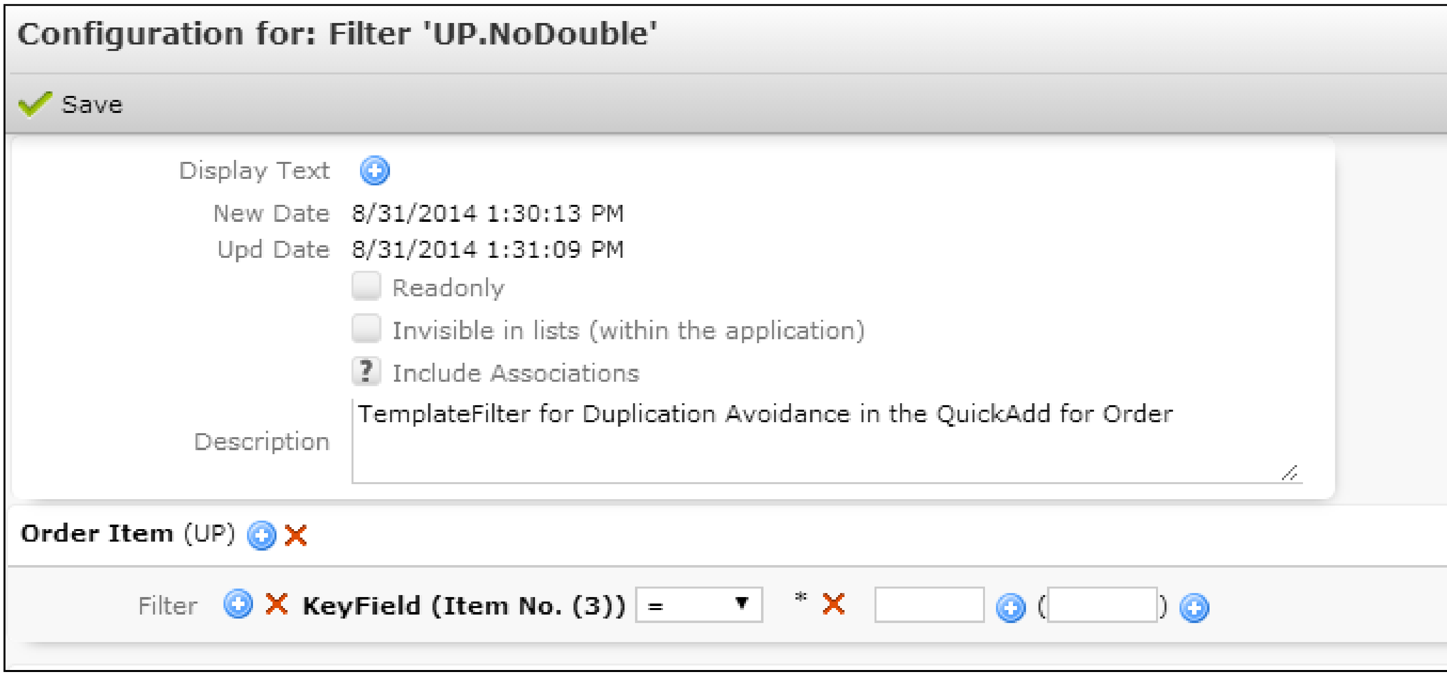
Example: Duplicate Prevention

The following example shows how Template filters prevents duplicates in the order entry quick add.

Define the following configuration units:

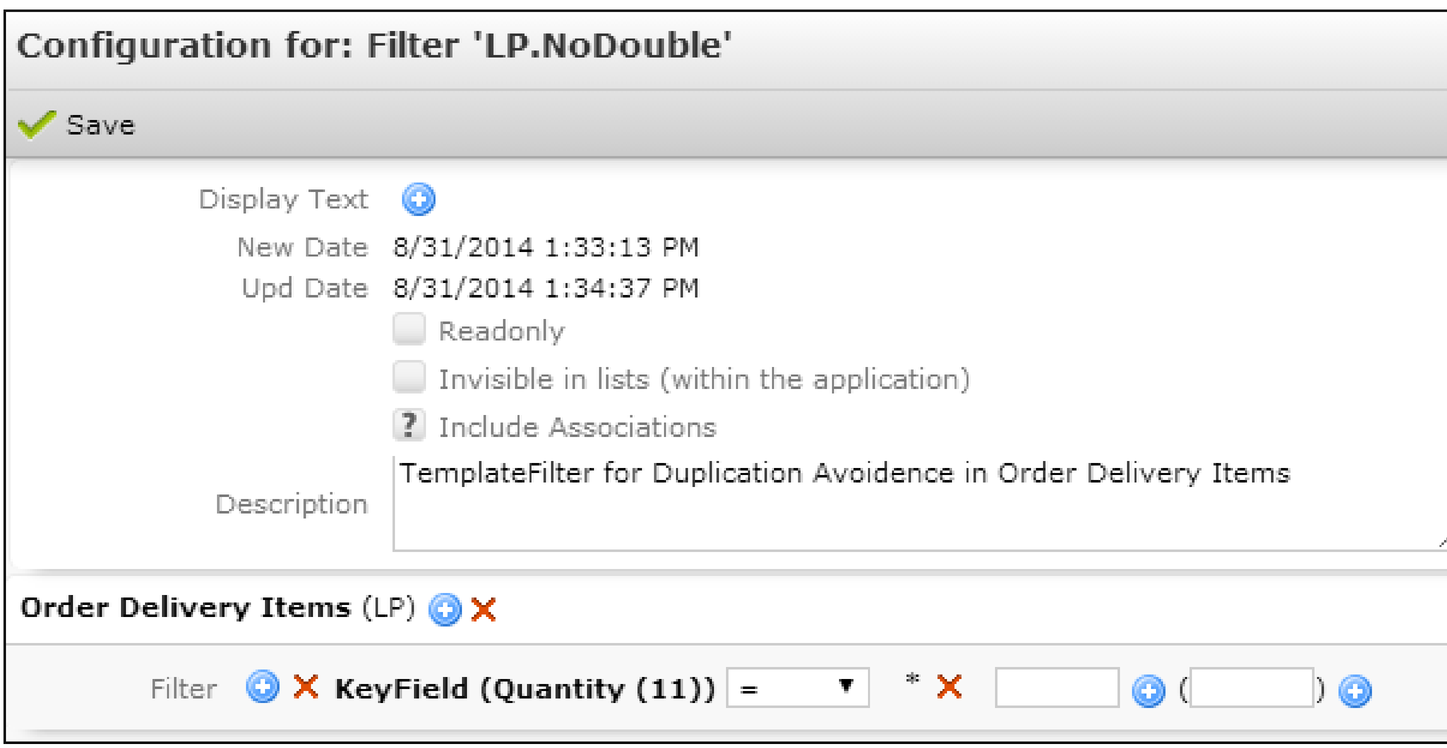
- `UP.NoDouble` filter: Defines that for each order and article only one record can exist.

  Note: This example definition does not apply in every implementation, therefore, you might have to adapt it.



You must define at least one value to be able to save the filter. "`*`" indicates that no pre-defined value is available (because if you define a value, e.g. "`123`", this value is written to every record).

- `LP.NoDouble` filter: This filter is applied to Delivery Items (LP) defining that for each Delivery Date (LD) and Order Item (UP) only one records is created.



The match-up is carried out using both links in the LP record, but to apply the key field logic at least one key field must defined.

You can use the delivery date from the LP record, but in this case you need to define that it is entered by the quick add functions. In this example, the Quantity field is checked as in case that a duplicate occurs this field contains the same value as the original value.

Set the following `SerialEntry` input arguments in the action call as follows:

- `DestinationTemplateFilter = UP.NoDouble`
- `DestinationChildTemplateFilter = LP.NoDouble`

Troubleshooting

In this topic you find information on how to solve known issues.

Adding a Debug Page

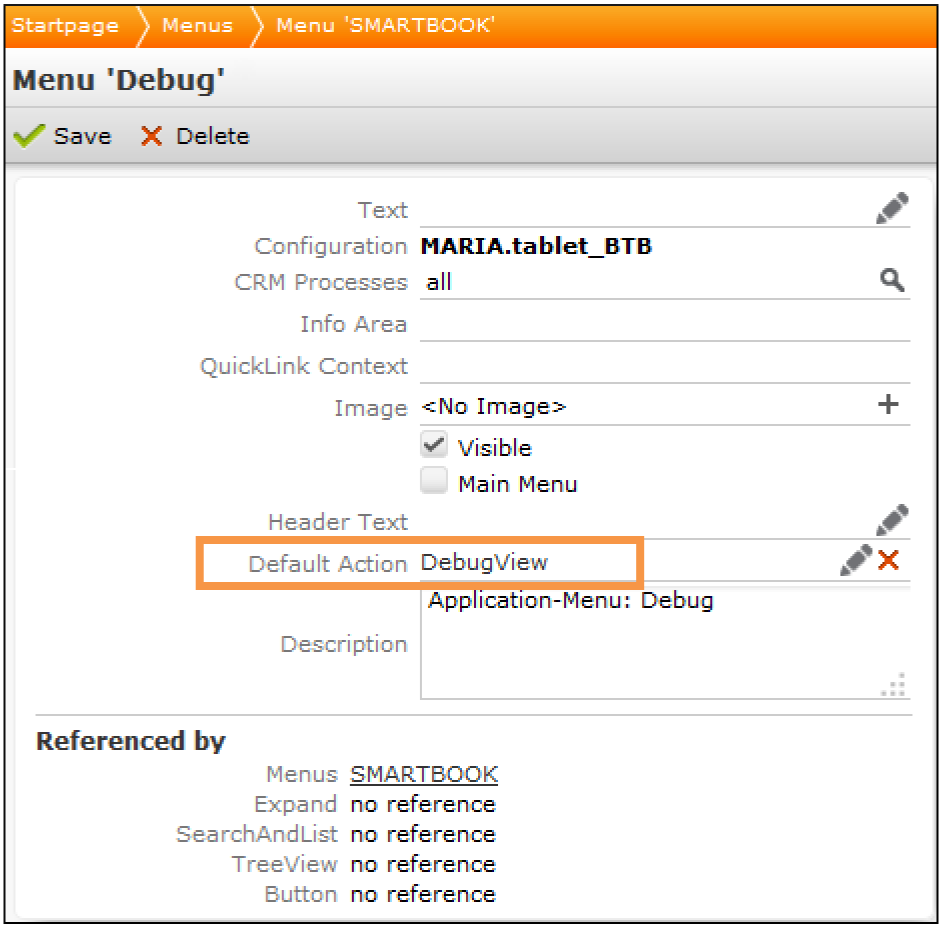
You can add a menu item that displays a debug page.

You can use the debug page e. g. to submit SQL statements or configure the logging.

Note: This page is only designed for development or configuration issues. Remove this page from user configurations.

To add the debug page:

1. Switch to the **Application Menus** area.
2. Add The `Debug` menu action to the `Smartbook` menu and assign the `DebugView` action (no parameters required):



Defining Error Messages

If a request was rejected by the server, by default, an error message is displayed. You need to analyze the log file to identify the problem.

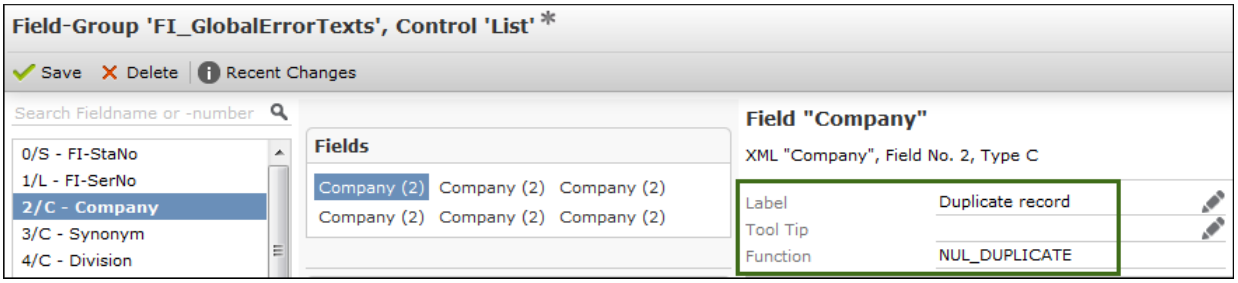
You can also define a more specific error message based on the technical error text.

To define error messages:

1. Define a List control.

   Example:

   Add the `FI_GlobalErrorTexts` field group, and define the List control (as the List control is applied by default) if no specific control is given. Add the desired fields (you can also add one field multiple times), e.g. the company field and assign the desired technical error message (e.g `NUL_DUPLICATE`) as function:



`NUL_DUPLICATE` defines that if `NUL_DUPLICATE` is part of the technical error message text, the **Label** text is displayed instead.

You can enter more than one text separated by semicolon into the **Function** field, in this case the error message is only displayed if all given strings can be found in the technical error message.

2. Enable the `System.ErrorTexts` Web Configuration parameter to display the alternative error texts, see System.ErrorTexts.

You can also assign such a field group to an action call. For example, the if `NUL_DUPLICATE` occurs for an order, it can indicate that the entered delivery date already exists.

1. Add a group error for the text field in the AU info area, e.g. AU_Errors.
2. Define a more specific error text in the **Label** field, e.g. "Record could not be saved, maybe the order date already exists?".
3. Assign the desired field group in the `ErrorTranslation` input argument of the `AddOrder` action call, e.g `AU_Errors.List`..

   All the information on syntax etc. also applies when specifying the `ErrorTransla-tion` argument; see System.ErrorTexts.

   If no hits are found in the `ErrorTranslation` argument's field groups, the field groups defined in the `System.ErrorTexts` Web Configuration parameter are also applied.

Note: These error messages are also displayed if synchronization conflicts occur

General Checks

Learn about the general checks in case any problem occurs.

If problems occur, check the following:

- Check the configuration of the `web.config` file.
  - For `sso`:

```
crmpad://configureserver?identification=<ServerID>&name= <Server-
name>&url=<web_URL>&authenticationType=sso&useragent=ipad
```

- For `ssocredentials`:

```
crmpad://configureserver?identification=<ServerID>&name= <Server-
name>&url=<web_URL>&authenticationType=ssocredentials&useragent=ipad
```

- Check if single sign-on is enabled and configured for the user (if you want to use it).
- If an error occurs, always check the `update_net.log` of the Aurea CRM web installation.
- Check whether .net 4.0 is installed and used for the application pool.
- Check whether the **ISAPI and CGI Restrictions** for the application pool's .net Framework 4.0 are set to `allowed`.

Problems concerning Rights

Learn to troubleshoot rights issues.

If right problems occur:

- Check if the Mobile Services right was assigned to the user in the login configuration (`mmcfg.exe`).
- Check if single sign-on was assigned to the user.
- If messages in the `update_net.log` state that an info area or a field does not exist, usually the field is missing in the CRM.designer data model. Try to upload the data model to the CRM.designer database using Aurea CRM web and carry out your last action again. You must also update CRM.pad's data model, see Data Model on page 49.

Connection Problems

Learn to troubleshoot server connections issues.

If server connections problems occur, check the following:

- Check in the iPad's **Settings** whether you are connected to the internet or to corporate WLAN. You can also try to open a web site (e. g. www.aurea.com) with a browser (e.g. Safari).
- If you can access a web site, check if you can access Aurea CRM web with the browser by entering the URL. If an error message is displayed stating that Aurea CRM web does not support Safari, you were able to connect to the server.

Login Errors

Learn to troubleshoot login errors.

The following errors may occur during login:

- If a user logs on the first time and "Hello WWW" is displayed, the server registration link was probably wrong.

  A server can be de-registered with:

  ```
  crmpad://removeServer?name=<Servername>
  ```

- If the initial login failed (error message: "Login failed – No data has been synchronized for this user yet. Please check internet connection & user name" or similar), you can check whether authentication was successful by entering one of the following URLs into Safari:
  - For anonymous access (with username/password):

    ```
    <myServer>/mobile.axd?Service=Authenticate&Username=<name>&Password=<password>&ClientVersion=<AppStoreVersionNumber>
    ```

    AppStore version number: e.g. 1.1
  - For authenticated access (with sso):
  - ```
    <myServer>/mobile.axd?Service=Authenticate&ClientVersion=<AppStoreVersionNumber>
    ```

    AppStore version number: e.g. 1.1

  The authentication was successful if content similar to the following is displayed:


```
{"authenticated":true,"sessioninfo":["200",[["repName","Martin Praunias"],
["repId","010001909"],["repGroupId","000000000"],["repDeputyId",""],
["repSuperiorId",""],["repGroupId",""],["tenantNo","0"],["tenantName",""],
["tenantAdd",""],["datamodelVersion","1.0"],["clientRequestTimeout",""]]]}
```


- If the login failed after the user has already logged in successfully at least once (error message: "Login Failed – Incorrect user name or password. Please check our login data and try again" or similar), try the following:
  1. Check the client log file for a response from the server with an error.
  2. Check the server log file for the error message.
  3. If you could not identify the cause, try the links for the initial login fail find out if a correct response from the server is returned. If yes, try the following statement to find out if the data model or the configuration is corrupted:
```

```
CRM.server/mobile.axd?Service=Synchronization&SyncDataModel=true&
SyncConfiguration=true&SyncCatalogs=true
```

Performance Problems

Learn to troubleshoot performance problems.

CRM.pad Server Request Timeout

Each CRM.pad server request is subject to two timeouts. The lower value determines the maximum execution time:

1. The server's ASP.NET timeout after which the server request is canceled (default = 90 seconds). You can increase this value in the `web.Config` file: `<!-- Increase script timeout to five minutes --> <httpRuntime executionTimeout="300" ... other configuration attributes .../>`
2. The default client request timeout of the CRM.pad is 60 seconds (i.e. 60 seconds, CRMpad assumes that the server does not respond). You can increase this value with the `ClientRequestTimeout` Web Configuration parameter, see Web Configuration Parameters and Layouts on page 23.

   Note: This setting is available for CRM.pad version 1.3.1 and higher in iOS6.

3. For a thin connection (e.g. GSM) the timeout hardly ever occurs. A timeout occurs only if time runs out before the first response package is transferred. Still, a full synchronization on a thin connection probably takes a long time.

Note: Do not set the timeout without limits (although possible) as the server timeout restricts the server's workload and the client timeout is important for usability.

Asynchronous Server Requests

The `ClientRequestTimeout` Web Configuration parameter defines the default client timeout, see ClientRequestTimeout. This timeout is also applied to a full synchronizations. It is often set to several minutes to ensure, that a full synchronization can be carried out.

This timeout also defines the time span until a "Server not available" error is generated if the server does not respond to a request.

If the user connects to the server using a VPN, the scenario that a network is available but a network connection is not established hardly occurs, therefore, a long timeout interval is rather annoying.

You can define that requests are handled asynchronously. Define a timeout for the server and a retry interval. If the request has not finished processing within the server timeout, the server informs the client about this pending request. The server checks after the retry interval whether the request was processed. If the request is still pending, the server re-checks regularly until the process has finished.

To enable asynchronous server requests, set the following Web Configuration parameters:

- System.AsyncRequestWaitTime. This parameter enables this feature.
- System.AsyncRequestRetryTime.
- System.DisableAsyncExecution.

Note: If you set `System.AsynchRequestWaitTime`, you can assign the default value (60 seconds) or even a smaller value to the `ClientRequestTimeout` parameter. All synchronous server requests (including the synchronous part of asynchronous server requests) are carried out within this time span.

Synchronization Problems

Learn about synchronization problems and how to solve them.

Synchronization Conflicts

Users can work offline when no server connection is available.

The request is then stored on the device as an uncommitted request and the changes are applied in the local database. When a server connection is re-established, CRM.pad automatically sends the uncommitted requests to the server and updates the data in the Aurea CRM database.

However, when a request cannot be committed the server returns an error. In this case:

1. The error is stored locally and the request is marked as blocked.
2. This request's changes are reset locally (i. e. changed data is not available for the user anymore when searching or viewing data – both offline and online).
3. All dependent uncommitted requests (i. e. changes to the same record and new child records) are also blocked and are not available for the user anymore.
4. The user is informed about the synchronization conflict and can try to fix the problem.

Conflict Handling - Example

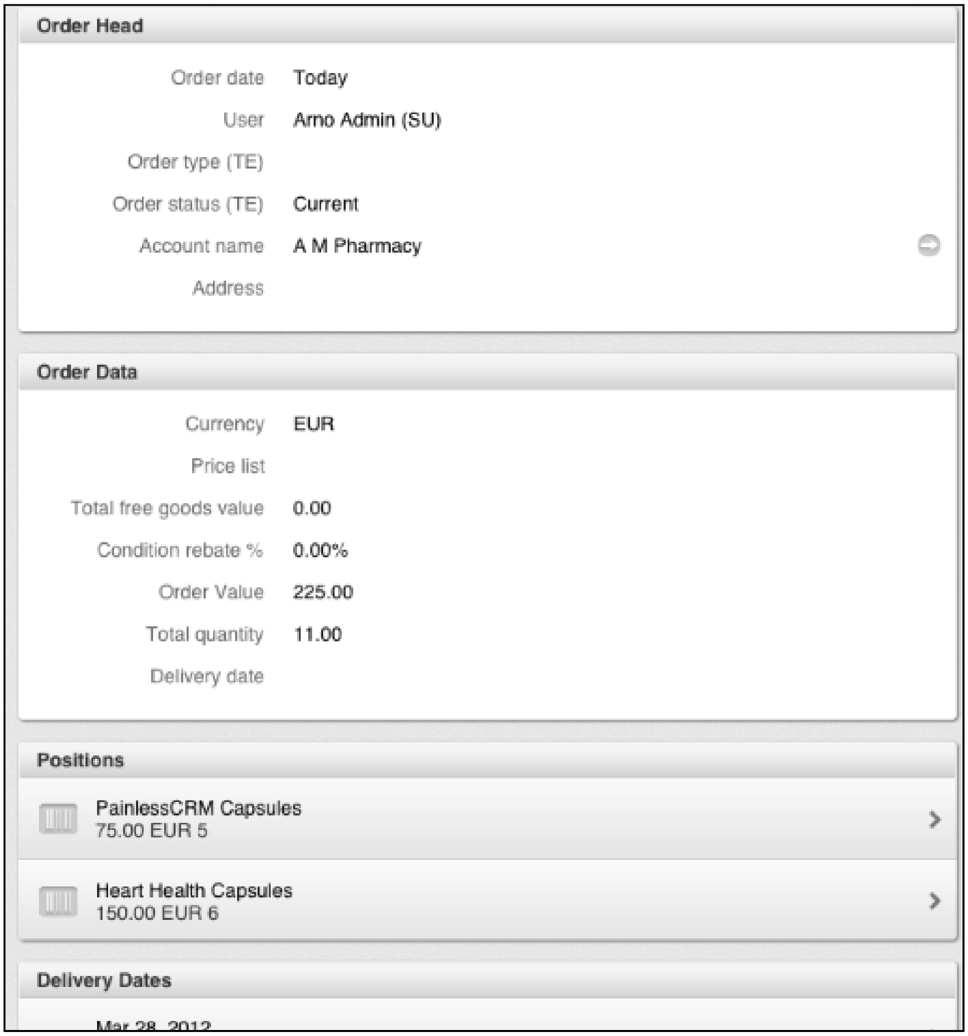
In the following example, creating a record results in a conflict and the error messages are displayed in CRM.pad:

1. A new order for a company is added containing two delivery dates on the same day.

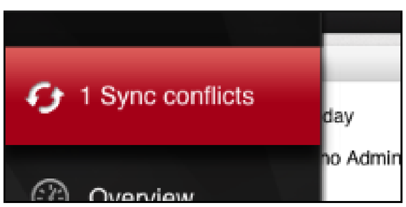
2. If a server connection exists, an error occurs:

The application's server log file shows a Duplicate-Key-Error thrown by Aurea CRM's business logic.

3. If no server connection exists, this error message is not displayed as CRM.pad does not check this occurrence. CRM.pad accepts this record in offline mode and the user can even add further order items. After entering the data, the record is probably displayed.

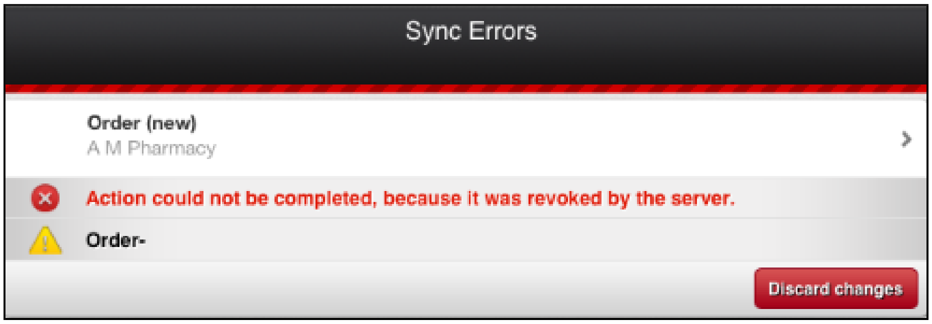


4. When CRM.pad is online again, the **Sync conflict(s)** are shown in the application menu.



The record created previously is not included in the order list anymore.

5. If the user taps on the **Sync conflict(s)**, the conflict's error message is displayed:



In addition to the error message, also the info area, the action (e. g. new) and whether a dependent process (e. g. quick add for offer items) exists are displayed.

6. The user can either click on **Discard changes** (the data is removed) or click on the conflict to fix it (in the example by changing the second date and saving the record again). Child processes are also automatically executed (and hopefully do not return another conflict).

Conflict Handling for Complex Server Requests

If your server requests are large and combined with complex business logic, they may take a long time until the server returns the commitment to the client. If the client does not receive or cannot process the commitment from the server (e.g. due to of the network connection loss), data inconsistencies can occur as the client tries to resend the request.

The following logic was implemented to solve this problem:

1. The client numbers each request and sends the numbers to the server.
2. The server stores each request number for each device and the IDs of the last request's affected records.
3. If the client sends a request number that already exists on the server, only IDs of changed records are transferred.
4. If the server request results in a conflict, a new request number is created.

To configure this logic:

1. Create a new info area with the following fields in the Aurea CRM data model:
   - `RequestNumber` (type: integer)
   - `RequestInformation` (type: text [as large as possible])
   - `RepId` (type: integer)
   - `LoginName` (type: text, min. 12 characters)
   - `ChildNumber` (type: integer)
   - `ParentKey` (type: text, min. 32 characters)
2. Add a new field group to this info area and define the Edit control as follows:

Field	Function
RequestNumber	RequestNumber
RecordIds	RecordIds
LoginName	LoginName
RepId	RepId
ChildNumber	ChildNumber
ParentKey	ParentKey

3. Add the field group name to the new `System.RequestControl` Web Configuration parameter (this parameter is not included in the default configuration).

Synchronization Timeout

Time-consuming single requests can occur during data synchronization.

All offline datasets in a request are queried during incremental synchronization base on a timestamp (the time of the last incremental synchronization). The server, therefore, only needs to check a small number of records and usually reacts within a short time period.

If timeout problems occur during incremental synchronizations (e.g. after many records were changed), carry out a full synchronization. If this problem occurs regularly, increase the request timeout.

Several requests are submitted one after another during a full synchronization:

- The first request transfers the complete application configuration, data model and catalog data to the client.
- The offline datasets that are synchronized are determined in CRM.designer. A request is submitted for each offline dataset, i.e. the full request timeout period is available for each offline dataset.

If problems occur during the first request, increasing the request timeout can help (although up to date no such cases are known). If the problem occurs while synchronizing one of the offline datasets, there are several options available to solve the problem. For further details, see Full Synchronization Terminates Due to the Timeout on page 385.

If you increase the timeout intervals, take into account that long synchronization requests result in a high load on the database server. Users are confronted with delays. Therefore, apply the other optimization options, in particular optimizing access rights and the database.

Change Requests after Client Timeout

If a Save request cannot be carried out due to a client timeout, the request is stored on the device.

All change request that occur after a client timeout are not carried out until the next UpSync is carried out.

Full Synchronization Terminates Due to the Timeout

If the full synchronization terminates, probably a timeout occurred for one of the datasets.

If this occurs, CRM.pad displays **Sync [info area]** for some time (until the request timeout occurs), then the synchronization stops. The info area displayed is usually the info area causing the problem.

To solve this problem try the following:

1. Set `LogLevel="Debug"` in the `settings.xml` file on the server and restart Aurea CRM web.
2. Synchronize again.
3. Analyze the log file (`update_net.log`) as described in the following chapters.

Analyzing the `update_net.log` Log File

Learn how to analyze the log file.

The below sections describe how to analyze the log file:

Finding the Cause

The `update_net.log` log file contains one line for each CRM.pad request, including the service name (in the following example "mobileService"):

```
2012.10.29 ;10:32:58 ; 0;mobileService ;Debug ;SID[41jep21jekzingzdalwtm-lxi];Service-Name: Authenticate,user=USER, configid=1015

[…]

2012.10.29 ;10:32:59 ; 938;mobileService ;Debug ;SID[41jep21jekzingzdalwtm-lxi];Service-Name: Synchronization,user=USER, configid=10000004
```

If multiple session IDs exist in the log file, Aurea recommends extracting the lines belonging to the same session. The timestamps on the left give information on how long each synchronization step took:

```
2012.10.29 ;10:33:39 ; 172;mobileService ;Debug ;SID[41jep21jekzingzdalwtm-lxi];Service-Name: Synchronization,user=USER, configid=10000004

2012.10.29 ;10:33:48 ; 500;mobileService ;Debug ;SID[41jep21jekzingzdalwtm-lxi];Service-Name: Synchronization,user=USER, configid=10000004

2012.10.29 ;10:33:54 ; 156;mobileService ;Debug ;SID[41jep21jekzingzdalwtm-lxi];Service-Name: Synchronization,user=USER, configid=10000004
```

```
2012.10.29 ;10:34:04 ; 172;mobileService ;Debug ;SID[41jep21jekzingzdalwtm-
lxi];Service-Name: Synchronization,user=USER, configid=10000004
```

The time between two requests is measured, i.e. the time span to execute the request on the server, to transfer the data and to process the data on the client. As client-side processing is quite quick, the difference gives a good estimation on how long processing the request takes in the network environment.

If synchronization problems occur, it is important to identify the request that causes the problem. In most cases this is the last request for the session in the log file. (Sometimes, additional requests from the same session are submitted afterwards, the runtime of the request can then be identified using the time stamps.)

You can find a semantic description of the submitted request (corresponding to the definition of the dataset and starting with `ReadByFilterMD`) under the line that corresponds to the request's start.

```
2012.10.29 ;10:34:05 ; 140;mobileService ;Debug ;SID[41jep21jekzingzdalwtm-
lxi];Service-Name: Synchronization,user=USER, configid=10000004
```

```
2012.10.29;10:34:05;63;mmObjects.Database;Info;SID[41jep21jekzingzdalwtm-
lxi]; ReadByFilterMD (C011='', HavingOptional(C011C011)[C011='F7028 >=
"20100101"', HavingOptional(C011C011FI)[FI='', HavingOptional(C011C011FI-
FI)[FI='',Having:0(C011C011FIFIFI)[FI='',Having:0(C011C011FIFIFIMA)[MA='(F0
>= "20121028" AND F0 <= "20121105")']]], HavingOptional(C011C011FIFI1)[FI='',
Having:0(C011C011FIFI1FI)[FI='', With:0(C011C011FIFI1FIFV)[FV='F0 = "$cur-
Rep"']]]]])
```

Reformat this entry to identify what occurred:

```
ReadByFilterMD (C011='',  HavingOptional(C011C011)[C011='F7028 >=
"20100101"', HavingOptional(C011C011FI)[FI='', HavingOptional(C011C011FI-
FI)[FI='',  Having:0(C011C011FIFIFI)[FI='',  Having:0(C011C011FIFIFI-
MA)[MA='(F0 >= "20121028" AND F0 <= "20121105")']]],  HavingOption-
al(C011C011FIFI1)[FI='',  Having:0(C011C011FIFI1FI)[FI='',
With:0(C011C011FIFI1FIFV)[FV='F0 = "$curRep"']]]]])
```

In this example, entries in the C011 info area that are linked to the FI info area are synchronized using a filter (defined in the C011 dataset). The FI records are determined according to the FI dataset definition based on the existence of MA records or Favorites.

However, the request's complexity and runtime are not immediately apparent from this information as it depends on the number of records in the corresponding tables, the defined access rights, and the size of any additional tables the access rights are applied to.

This is then followed by the SQL statement, which can be even more complicated because this statement needs to depict the access rights applied to the affected info areas (C011, FI, MA, FV) which are defined in the Rights module (`mmri.exe`).

```
2012.10.29;10:34:05;0;update.net.core.Sql;Info;SID[41jep21jekzingzdalwtm-
lxi];<RPACK=C011; Cur=1; SID=13A63FF0;
```

```
2012.10.29 ;10:34:05 ; 15;update.net.core.Sql ;Info ;SID[41jep21jekzingz-
dalwtmlxi]; SELECT [...]
```

"Z Fields" in Linked Info Areas

This problem can be identified by a quite high number of follow-up requests applied to other info areas (with conditions applied to ID) following the initial SQL statement in the synchronization request:

```
2012.10.30 ;13:50:36 ; 16;update.net.core.Sql ;Info ;SID[nqg-
pay55rl3iil45lagq1l45];<RPACK=MA; Cur=1; SID=07AF61D0;

2012.10.30 ;13:50:36 ; 15;update.net.core.Sql ;Info ;SID[nqg-
pay55rl3iil45lagq1l45]; SELECT [...]

2012.10.30 ;13:50:37 ; 765;update.net.core.Sql ;Info ;SID[nqg-
pay55rl3iil45lagq1l45]; EXEC=765; Conn=2; Cur=1; Rec=1024; Ret=0; Res=1024;

2012.10.30;13:50:37; 15;update.net.core.Sql ;Info ;SID[nqg-
pay55rl3iil45lagq1l45];<READ=FI; Cur=0; SID=07AF61D0; ' ,0,0';

2012.10.30;13:50:37; 0;update.net.core.Sql ;Info ;SID[nqg-
pay55rl3iil45lagq1l45]; SELECT [...] WHERE ID=?

2012.10.30 ;13:50:37 ; 0;update.net.core.Sql ;Info ;SID[nqg-
pay55rl3iil45lagq1l45]; V1='429496729665';

2012.10.30 ;13:50:37 ; 0;update.net.core.Sql ;Info ;SID[nqg-
pay55rl3iil45lagq1l45]; EXEC=0; Conn=2; Cur=0; Rec=1; Ret=0; Res=1;

;SID [nqgpay55rl3iil45lagq1l45];<READ=KP; Cur=0; SID=07AF61D0; ' ,0,0';

2012.10.30;13:50:37; 0;update.net.core.Sql ;Info ;SID[nqg-
pay55rl3iil45lagq1l45]; SELECT […] WHERE ID=?

2012.10.30 ;13:50:37 ; 0;update.net.core.Sql ;Info ;SID[nqg-
pay55rl3iil45lagq1l45]; V1='429496730029';

2012.10.30 ;13:50:37 ; 0;update.net.core.Sql ;Info ;SID[nqg-
pay55rl3iil45lagq1l45]; EXEC=0; Conn=2; Cur=0; Rec=1; Ret=0; Res=1;

2012.10.30 ;13:50:37 ; 0;update.net.core.Sql ;Notice ;SID[nqg-
pay55rl3iil45lagq1l45];timer=0; Ok; 0ms; SID=07AF61D0;>

2012.10.30;13:50:37; 0;update.net.core.Sql ;Info ;SID[nqg-
pay55rl3iil45lagq1l45];<READ=FI; Cur=0; SID=07AF61D0; ' ,0,0';

2012.10.30;13:50:37; 0;update.net.core.Sql ;Info ;SID[nqg-
pay55rl3iil45lagq1l45]; SELECT […] WHERE ID=?

2012.10.30 ;13:50:37 ; 0;update.net.core.Sql ;Info ;SID[nqg-
pay55rl3iil45lagq1l45]; V1='4294967308';

2012.10.30 ;13:50:37 ; 0;update.net.core.Sql ;Info ;SID[nqg-
pay55rl3iil45lagq1l45]; EXEC=0; Conn=2; Cur=0; Rec=1; Ret=0; Res=1;
```

You can avoid this problem by using the corresponding link fields in the configuration instead of the Z fields, i.e. the fields for the first name and last name from the KP info area instead of the "Person" Z field. These fields must subsequently be synchronized with the KP dataset.

This issue depends on the number of synchronized records. Reduce the offline dataset in order to execute a smaller number of additional statements.

The Runtime of the Initial Statement is too Long

The problem can be identified by the first statement taking a long time (or because it is the last entry for this session in the log file).

Processing the request on the database server takes a long time. The runtime is determined by the dataset and access rights definition. The runtime depends on the size of the involved tables and how easy it is to depict the conditions applied to these tables. Therefore, this problem can also occur if the synchronized data volume is quite small.

The Statement Returns too Much Data

The problem can be identified by the statement executing successfully but returning a very large amount of data - this is the case if the statement completes executing (i.e. a runtime and results set are specified) and is followed by several RPACK lines for the info area being synchronized.

```
2012.10.30 ;13:50:36 ; 16;update.net.core.Sql ;Info ;SID[nqg-pay55rl3iil45lagq1l45];< RPACK =MA; Cur=1; SID=07AF61D0;

2012.10.30;13:50:36; 15;update.net.core.Sql ;Info ;SID[nqg-pay55rl3iil45lagq1l45]; SELECT [...] FROM BTB70_MA […]

2012.10.30 ;13:50:37 ; 765;update.net.core.Sql ;Info ;SID[nqg-pay55rl3iil45lagq1l45]; EXEC=765; Conn=2; Cur=1; Rec=1024; Ret=0; Res=1024;

2012.10.30 ;13:50:40 ; 0;update.net.core.Sql ;Info ;SID[nqg-pay55rl3iil45lagq1l45];< RPACK =MA; Cur=1; SID=07AF61D0;

2012.10.30 ;13:50:40 ; 16;update.net.core.Sql ;Info ;SID[nqg-pay55rl3iil45lagq1l45]; EXEC=16; Conn=2; Cmd=FNEXT; Cur=1; Rec=1024; Ret=0; Res=1024;

2012.10.30 ;13:50:40 ; 0;update.net.core.Sql ;Info ;SID[nqg-pay55rl3iil45lagq1l45];< RPACK =MA; Cur=1; SID=07AF61D0;

2012.10.30 ;13:50:40 ; 16;update.net.core.Sql ;Info ;SID[nqg-pay55rl3iil45lagq1l45]; EXEC=16; Conn=2; Cmd=FNEXT; Cur=1; Rec=1024; Ret=0; Res=1024;

[...]
```

This problem can also cause CRM.pad to "crash" (it is terminated by the iOS operating system because the response cannot be processed due to a lack of memory resources). This occurrence depends on a record's width (the number of fields and their length) and on the number of records (usually more than 50,000 records) in one info area synchronized on the device. Try to reduce the data volume that is synchronized to improve the performance of the client.

Optimizing the Runtime of an Offline Dataset

Learn how to optimizing the runtime of an offline dataset.

The below sections describe how to optimize the runtime of an offline dataset:

Simplifying the Dataset Definition

A dataset determines all data in an info area that either is included in one of the filters specified in the dataset or is linked to at least one of the specified child records.

These conditions are included in a single SQL statement. In addition, the conditions and dependencies defined in child datasets as well as the access rights defined for all involved info areas are also included in this statement.

Combining child datasets and/or access rights definitions often results in complex database requests (which take a long time to execute).

Using filters instead of dependencies result in simpler SQL statements. Depending on the data volume you can also export all records the user is only allowed to view.

Splitting the Dataset Definition

More than one dataset can be defined for an info area and each definition is treated as an independent request. If you define complex filters (linked using "OR") or multiple child info areas (that are also linked using "OR"), each setting can be moved to a separate dataset.

The same record can be returned by multiple datasets, but avoid large overlaps as this data needs to be transferred over the network multiple times to the iPad.

If a dataset is split into multiple datasets, child datasets are only synchronized with the original dataset. For example, if KP is synchronized for all FI, and the FI dataset is split into FI, FI2 and FI3, both FI2 and FI3 need to be referenced by the KP dataset, or separate KP2 and KP3 datasets need to be defined for FI2 and FI3 to ensure that all KP records are transferred for all FI records.

Simplifying Access Rights

Only a user's view access rights are relevant for the synchronization. Bottom-up access rights (where the access rights depend on sub-tables) are performance-intensive, e.g. "companies with a linked SB record where the user is entered as a rep" or "companies that had contact with the user in the last 2 years". Multiple combinations of such bottom-up access rights result need more resources to process.

Try the following approaches (which can be combined):

- Simplification: Edit the access rights so that they are easier to evaluate, the trade-off being a semantic change to the access rights definition. Usually, data exists that the user may not view (set A), data that the user may view but is not interested in (set B) and data that the user may view and is interested in (set C). In this case, you can try to simplify the access rights for set B. The user still cannot view set A while having a complete full view of set C.
- Pre-triggering: Access rights can also be simplified by using triggers that create the necessary access information upon saving the record. In an extreme case, a record could exist in a child info area for each FI record and the user is allowed to view the record. In this case, reduce the access rights to a single bottom-up condition. However, triggers need to be used to ensure that these child info areas are kept up to date.

Measures that simplify access rights not only improve the performance of CRM.pad's synchronization but also the performance of queries in all applications that access the Aurea CRM database.

Optimizing the Database

Apply the following:

- Optimizing the database indexes (either manually or with a database tool)
- Maintenance (keeping database statistics up-to-date, avoiding the fragmentation of data files)

Server Problems

Learn how to identify and solve server problems in CRM.pad.

The following error message indicates that the problem is a server issue (and not a <CRM.pad> issue): "The action could not be completed because it was rejected by the server".

1. Check the server log file (in Debug mode with WebTrace) to identify the problem (especially when the user states remarks like "but in web it is working"). CRM.pad is not a web application, therefore functions may work in Aurea CRM web but not in CRM.pad
2. Compare Aurea CRM web statements with those from the CRMpad for differences. For example, some fields that are filled in web but not in CRMpad may cause the problems.
3. Enable the request and result log for information on the complete response from the server.

Lost Server Connection

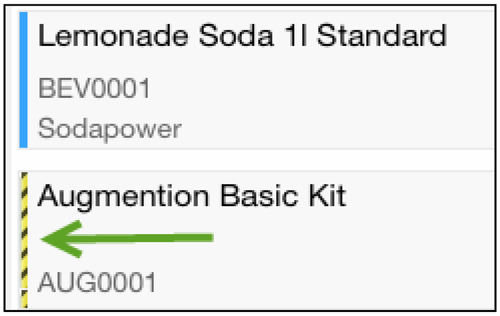
It can happen that a user cannot establish the connection to the server although an internet connection is available.

In this case after logging on only the message "Please wait, data is loaded" is displayed (instead of the overview dashboard). A user can try to switch to an info area (e. g. Persons) from the menu. If data is displayed, this indicates that he can access the offline data stored locally on the device.

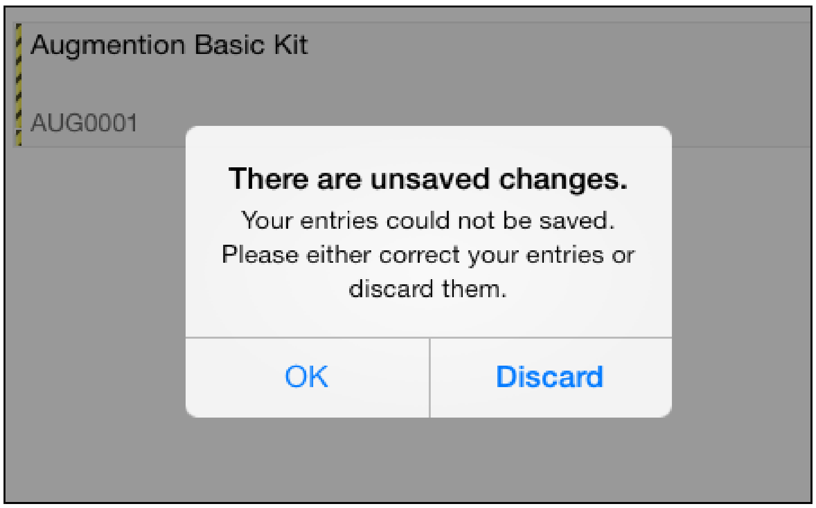
Quick Add Related Error and Conflict Handling

When a user is working offline, it might occur that some of the data entered causes a server error. Unfortunately, the user is not aware of that fact, the conflict only occurs in online mode during synchronization.

If the server rejects transferred rows, they are marked accordingly on the device:



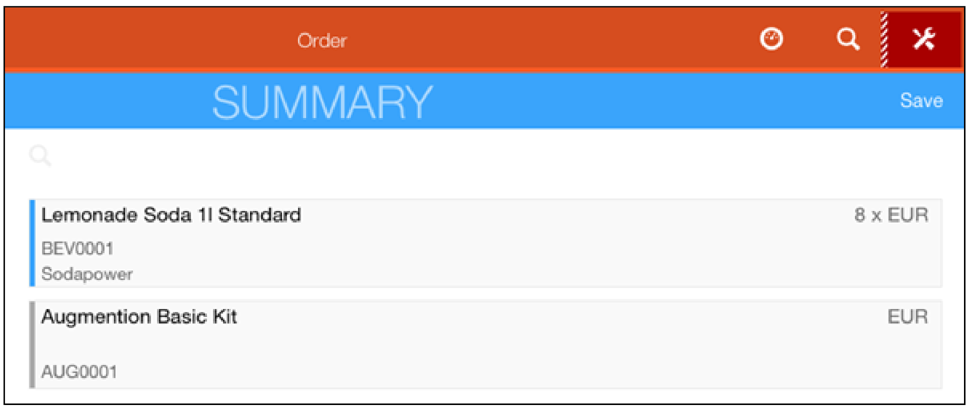
If the user tries to close the quick add, all unsaved changes are transferred to the server once more. If an error occurs, the user can either correct the error or discard the changes:



If the user taps on OK and saves the record again, the changes are applied on the device.

As soon as a server connection can be established, the changes are transferred to the server and - probably - rejected:

The changes can once more be discarded, or the user tried to solve the conflict. The record is then only transferred to the server after the user taps on the Save button.



"Language Mix"

Learn to troubleshoot language and region settings.

Sometimes strings are displayed in two languages, e.g. German and English.

Check the iPad's iOS language and the region settings. If these two settings do not comply, it might occur that e.g. calendar strings like weekdays or months are displayed in a different language.