Customer Relationship Management

CRM.Web Offline Installation Guide

Version 14.1

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

Learn to set up Aurea CRM Web in an offline environment.

This document provides all information necessary to set up and run Aurea CRM web in an offline environment. It describes the following:

- all necessary preparation steps on the server, see Preparations on the Aurea CRM Server on page 14
- installation of Aurea CRM web offline on the clients, see Setting Up Offline Clients on page 31
- Administrative tasks when running Aurea CRM web offline, see Administration on page 52.

For additional information, see the article “web offline” at https://support.aurea.com.

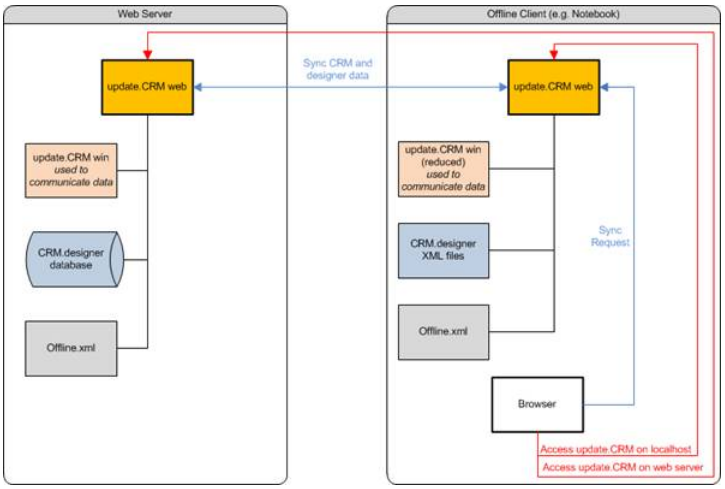
Some of the steps described in this document require a detailed knowledge of administrative tasks in Aurea CRM (communication, CRM.designer etc.). Please refer to the respective documentation for information.

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

All screenshots and examples in this manual apply to the BTB vertical.

Aurea CRM Web Offline Overview

Aurea CRM web offline enables users to continue working with Aurea CRM web without an internet connection.



In offline mode, Aurea CRM web is connected to a local Aurea CRM database and uses CRM.designer configurations stored locally as (read-only) XML files. A synchronization mechanism ensures data transfer between offline clients and the server. Synchronization of the CRM web offline and the database can be configured to use the CRM.Launcher cookies for database synchronization.

An Aurea CRM web offline implementation consists of:

- Aurea CRM web server installation containing:
  - an installation of Aurea CRM win
  - an installation of Aurea CRM web
  - an Aurea CRM database
  - a properly configured `Offline.xml` file
- Number of Aurea CRM web offline client installations containing:
  - a local installation of MS SQL 2012 LocalDB for the local Aurea CRM database
  - a local installation of IIS Express 8.0 allowing only localhost access
  - a reduced installation of Aurea CRM win (containing `mmcfg.exe`, `mmco.exe`, `mmim.exe`, `mmRegServers.exe` and `mmri.exe`)
  - a local installation of Aurea CRM web

- a properly configured `Offline.xml` file
- a local installation of CRM.launcher (handling the initialization and patching of Aurea CRM web offline)

Aurea CRM web offline Functionality

Learn about the functions supported by Aurea CRM web offline.

Using Aurea CRM web offline provides both an online and offline mode of Aurea CRM web offering the same browser-based user interface. Thus, users can continue to work offline with only minimal functional restrictions.

Aurea CRM web offline supports the following functions:

- Searching, viewing, editing, creating and deleting records
- Executing predefined queries and analyses
- Creating reports
- Using processes

The following functions are not supported:

- Defining new queries and analyses

Note: If queries are saved in the Aurea CRM database, users can create/edit queries in offline mode as well. Only queries stored in the CRM.designer database are read-only. For further details, see Query.SaveTo and Query.LoadFrom in the Aurea CRM web Administrator Guide.

- Accessing 3rd-party applications that require online access (e.g. ERP or document management system)
- Modifying individual user settings that are saved in the CRM.designer configuration (e.g. customizing lists or re-arranging Expand views)
- Executing tasks that require CRM.server (e.g. generating a mail merge letter for a marketing activity)

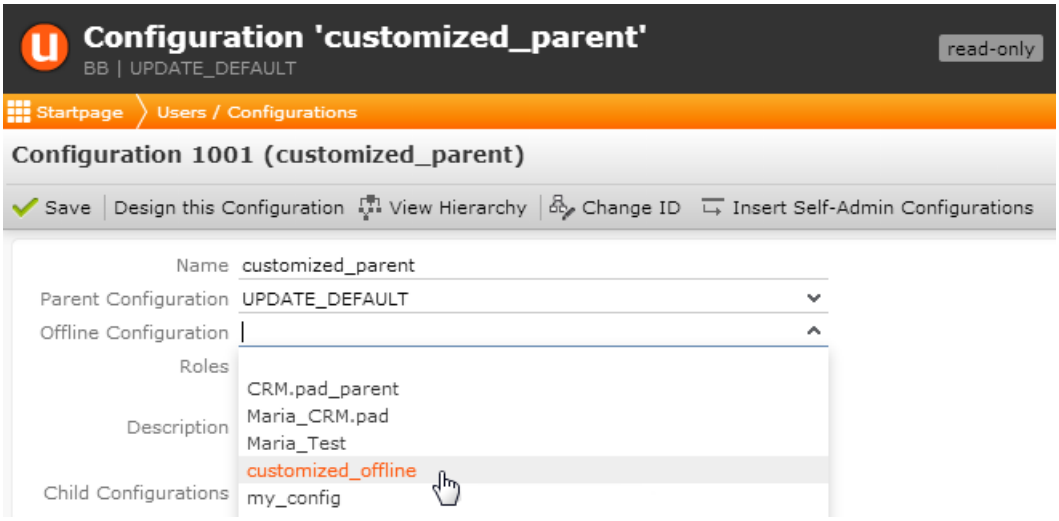
Creating Aurea CRM web offline Configurations

You can create dedicated offline configurations in CRM.designer e.g. to hide menu entries or buttons calling functions that are not supported in Aurea CRM web offline (e.g. hiding the **Modify List** button from `SearchResults` headers).

For general information on CRM.designer configurations, see Users and Configurations in the *Aurea CRM web Administrator Guide*.

To create and assign an Aurea CRM web offline configuration:

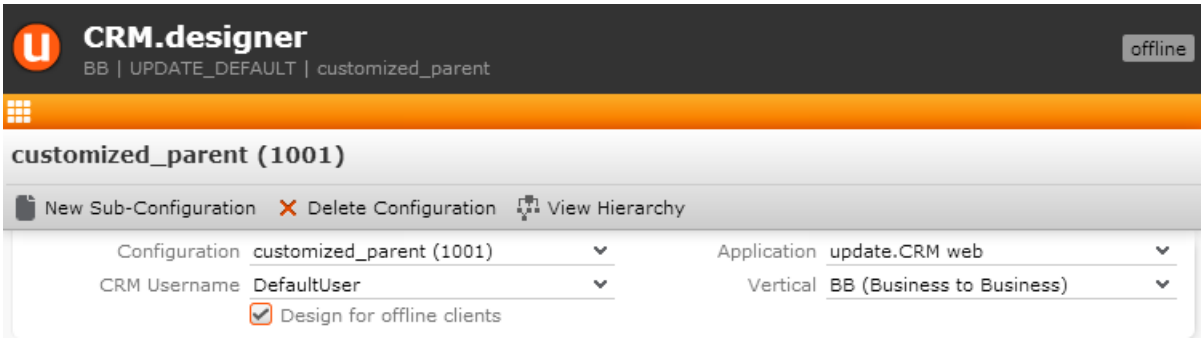
1. On the CRM.designer main page create a new configuration (**New Sub-Configuration** button), e.g. a "customized_offline" as common parent configuration used on all offline clients.
2. Click on **Users | Configurations**.
3. Switch to the **Configurations** view (radio button)
4. Select the online configuration to which you want to assign the newly created offline configuration (e.g. "customized_parent") by clicking (**Edit**).



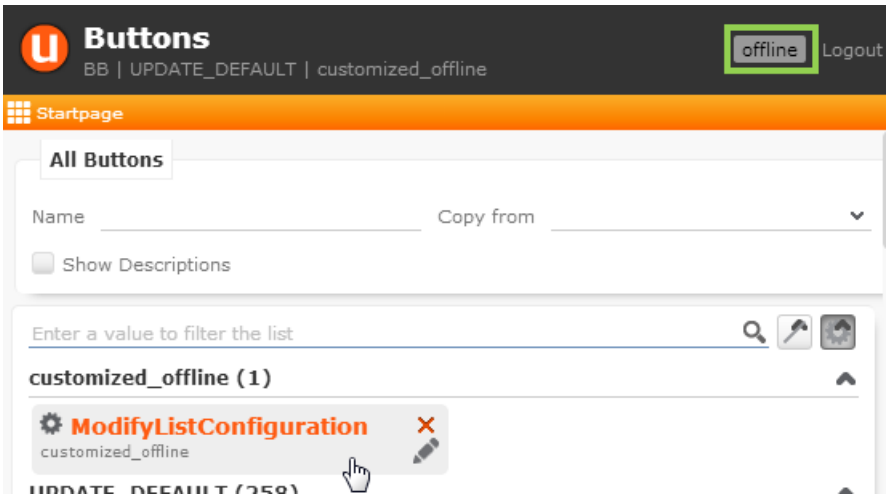
5. Select the offline configuration from the **Offline Configuration** drop-down list.
6. Save.

To design an offline configuration:

1. On the CRM.designer main page select the online configuration for which you want to implement offline customizations ("customized_parent" from the above example).
2. Enable the **Design for offline clients** check box.



All configuration changes are saved in the assigned offline configuration ("customized_offline").



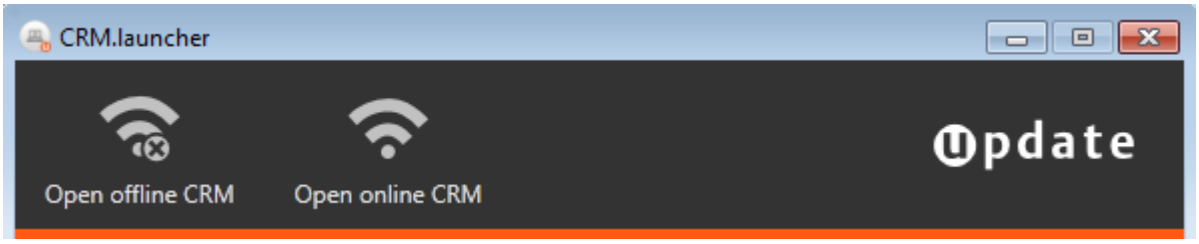
Note: Make sure you only implement configuration changes that really apply to Aurea CRM web offline when designing such a configuration.

Switching Between Online and Offline Mode

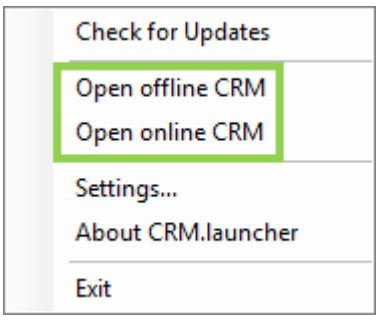
Learn how to switch between Online and Offline Mode.

Users have the following possibilities to start Aurea CRM web offline:

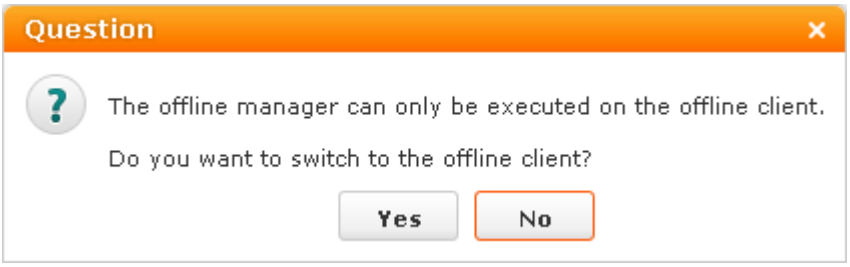
- CRM.launcher allows users to start Aurea CRM web alternatively in online or offline mode, see CRM.launcher on page 52.
  - At startup:



  - From the context menu:



- The **Offline Manager** menu item in Aurea CRM web allows users to switch to offline mode while working with Aurea CRM web, see Offline Manager on page 61.



  Offline Manager allows users to synchronize their local configuration and Aurea CRM database with the server, see Synchronization.

Note: Online mode is only available if the client is connected to the server. Offline mode is only available if the offline client is in a stable condition, i.e. no patches are currently being installed.

IIS Express 8.0

The IIS Express 8.0 web server is a simple web server handling all HTTP requests on the offline client.

IIS Express 8.0 can be installed as part of the Aurea CRM web offline setup, see Installing 3rd-Party Products. Setup writes all necessary settings in the `applicationhost.config` file located in `..\web.data`.

IIS Express 8.0 is started by CRM.launcher:

CRM.launcher reads all relevant settings from the `applicationhost.config` file and starts the IIS as the current user with those settings. CRM.launcher uses the port specified in the `Offline.xml` (`< CommonClientUrl >`).

Starting IIS Express 8.0 without CRM.launcher:

By default, the IIS Express 8.0 web server listens to HTTP requests on TCP/IP port 8080. You can specify another port via command line parameter. For a list of supported command line parameters. For further details, see Running IIS Express from the Command Line.

Note: Make sure that the IIS Express 8.0 web server is always used with exactly the same host name; it is not supported to e.g specify "localhost" as `< CommonClientUrl >` in the `Offline.xml` but start the web server with "127.0.0.1" via the command line.

2

Setup and Installation

Learn to set up and install Aurea CRM web offline.

The Aurea CRM web offline setup is executed in silent mode. After you completed all preparative steps, you can deploy the setups to your users who can then execute the setup on the client.

Following are the tasks involved in set up and installing Aurea CRM web offline:

Preparations on the Aurea CRM Server

This topic explains the prerequisits for set up and installation.

You need to complete the following preparation steps on the Aurea CRM web server before you can set up offline clients:

- A server installation of Aurea CRM web and Aurea CRM win must be available. For details, refer to the respective installation guides.

  Note: Aurea recommends that all necessary customizations for Aurea CRM web are implemented before setting up offline clients. For details, refer to the *Aurea CRM web Administrator Guide*.

- Create the required users, logins, default values and communications formats in the Aurea CRM database, see Preparations in the Aurea CRM database on page 15.
- Make sure the required CRM.designer configurations are available and up-to-date, see CRM.designer Configurations for Offline Clients on page 19.
- Specify Web Configuration parameters for Aurea CRM web offline, see Web Configuration Parameters on page 21.
- Specify all required (server-side) settings in the `Offline.xml`, see Offline.xml on page 22. The settings for the client are created by the setup.
- Prepare the `SilentClient.ini` file, see SilentClient.ini on page 27.
- Create the required directory structure and copy all necessary files to their location, see Folder and File Preparation on page 30.

Preparations in the Aurea CRM database

Learn about the records used to create the initial data stock download files for the offline clients.

For detailed information, see Configuring Stations Automatically in the *Aurea CRM win Administrator Guide*.

Note: When upgrading from update.seven to Aurea CRM, make sure each offline user has a station number assigned in his rep record before migration. All necessary steps are described in the following topics. Otherwise you need to manually enter the client station number (`<ClientStationNumber>` in the `Offline.xml` file) on each offline client after installation, before initialization.

The following records and formats must be available in the Aurea CRM database before you can set up offline clients:

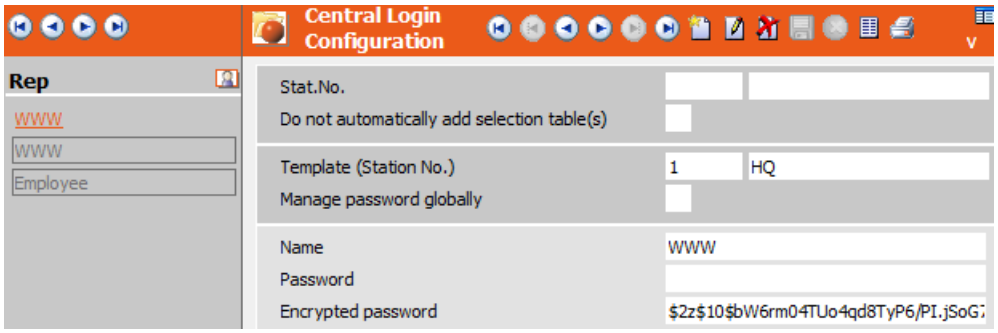
Central Login Configuration for Administrative Users

For administrative users that need to be able to access all offline clients you need to create **Central Login Configuration** records (ES) in the Aurea CRM win **Rights** module. Common administrative users are the SU, WWW and COM users.

Create central logins that are valid for all stations:

1. For each of these users, create a **Central Login Configuration** record without entering a station number.

   The login is then distributed to **all** stations.



   By setting the **Template (Station No.)** field to the main office's station number, you can update the login record on the main office, and the changes are automatically distributed to the offline stations.

   If you check **Manage password globally** as well, password changes are also distributed to the offline stations.

Note: If you change the password of the SU or WWW user centrally, you need to distribute a new `users.xml` file to your offline stations. For further details, see *CRM.users tool* in the Aurea CRM web Installation & Technical Guide.

For general information on the Central Login Configuration info area, see Central Login Configuration in the *CRM.core Administrator Guide*.

Communication Formats

The data transfer between the Aurea CRM database on the Aurea CRM web server and the local Aurea CRM databases on the offline clients is performed by the Aurea CRM win **Communication** module (`mmco.exe`). For further details, see *Communication* in the *Aurea CRM win Administrator Guide*.

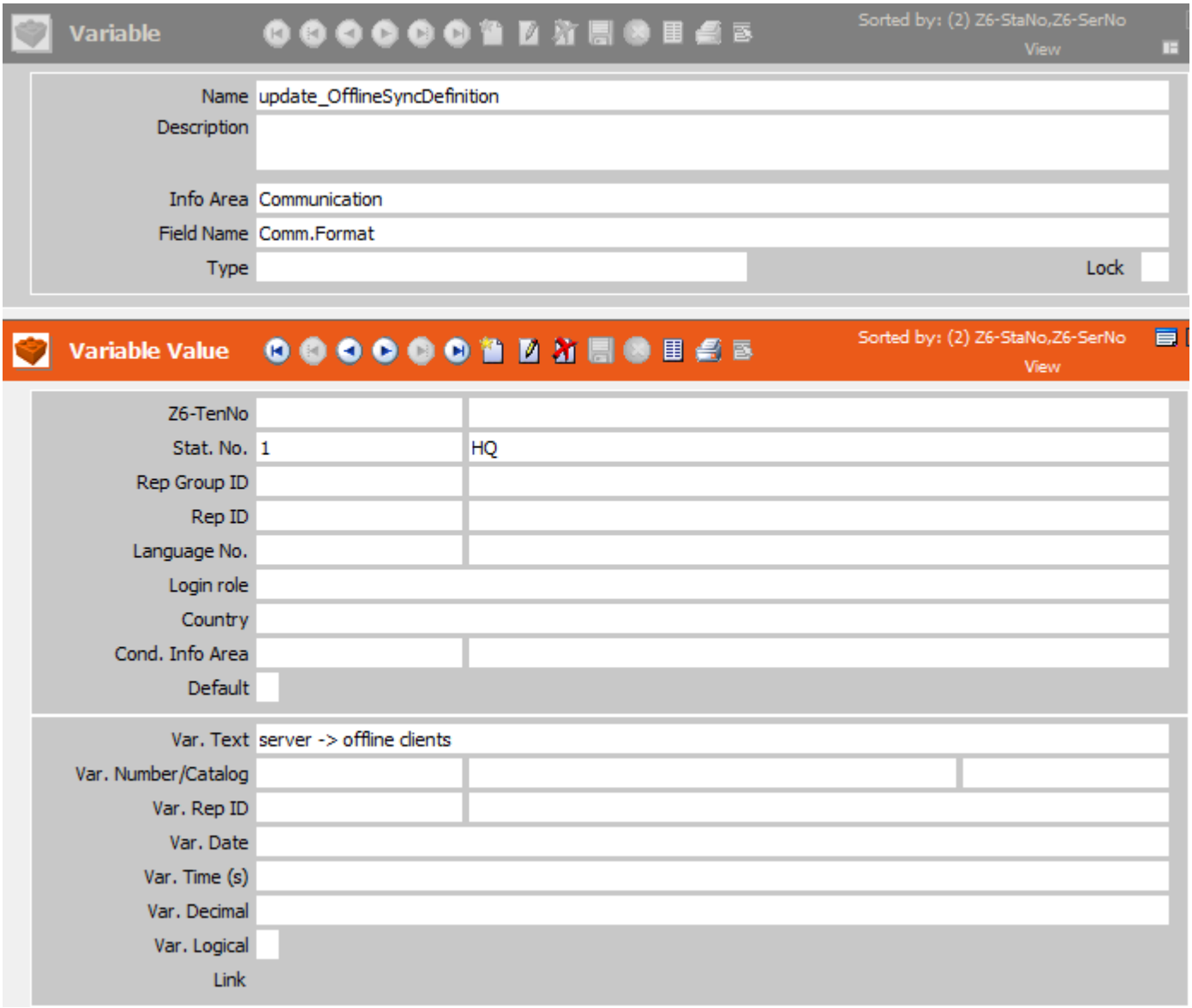
To define which data is transferred, you need to define a communication format for each direction:

- **server -> offline client:** To optimize performance, limit the data that is output to the offline clients using variable conditions, e.g. output only the rep's own customers, activities within a certain time frame etc.
- **offline client -> server:** Define a communication format that transfers all records from the offline clients to the server, i.e. check the output column for all info areas.

For detailed information on defining communication formats, see Communication Format in the *Aurea CRM win Administrator Guide.*

update_offlineSyncDefinition Variable

To globally set the communication format to be used for the communication connection server -> offline client, you need to define a variable in the Z5 info area, see Variables in the *Aurea CRM.core Administrator Guide*.



Create a variable "update_OfflineSyncDefinition" for the **Comm.Format** field in the **Communication** info area. Define a value that contains the name of the communication format used to communicate between the server and offline station (**Var. Text** field).

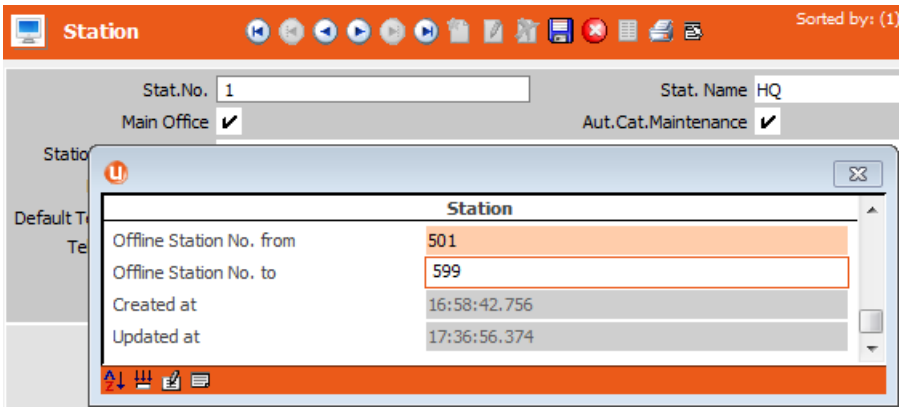
The variable must be defined so as to be applicable by all users that can add offline stations (e.g. a condition applied to the offline server's station number).

When creating offline users, the communication format for the connection server -> offline station is read from this variable. For further details, see Defining Users as "Offline Users".

Station Number Range for Offline Clients

Define a station number range for your offline clients:

1. In the Aurea CRM win **Rights** module, switch to the **Station** record (AS) of the Aurea CRM web server.
2. Enter the number range in the **Offline Station No. from/to** fields.



When creating offline users, the station number is automatically entered in the ID record and the respective station record (AS) is created. For further details, see Defining Users as "Offline Users".

Alternatively, you can manually enter the offline client's station number in the ID record of the respective offline user.

Default Values

To automatically set values for the offline clients, define default values in the server's station access rights definition. For further details, see *Default Values when Adding Records* in the *CRM.core Administrator Guide*.

You should at least define the following defaults:

- **Central Login Configuration** (ES): Module access settings for Aurea CRM win and Aurea CRM web.
- **Station** (AS): Select the access rights definition to be applied to the offline clients under **Station Access Rights**.
- **Communication** (C1): Select the communication format to be used for the communication connection offline client -> server under **Comm.Format**.

  You can define additional default values, triggers and workflows used to add records in these info areas.

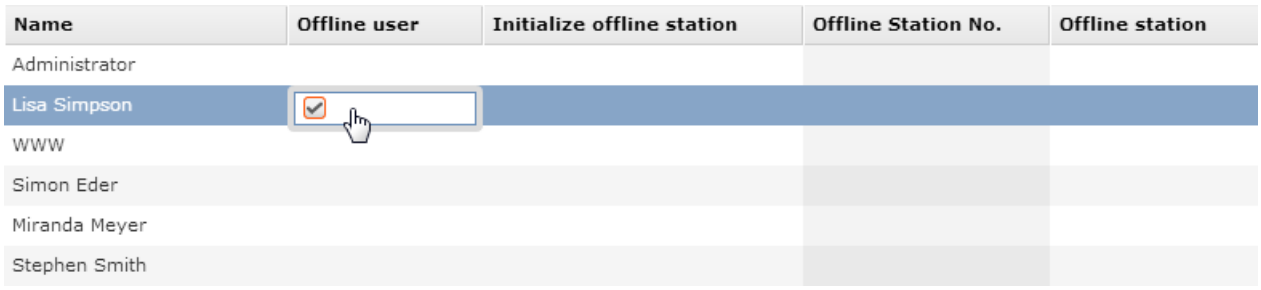
When creating offline users, the corresponding records are created using the defined default values, see Defining Users as "Offline Users".

Defining Users as "Offline Users"

Enabling the **Offline user** field in the **Rep** info area (ID) triggers the automatic creation of all necessary records required for the offline client (station, login, communication connections).

To define a user as offline user:

1. Switch to the desired **Rep** record (ID). The rep must be of type "Employee".



2. Check the **Offline user** field.

Upon saving the record Aurea CRM automatically adds/updates the following records:

- A new **Station** record (AS) is created. The station number is the next free station number from the station number range defined in the server's station record, see Station Number Range for Offline Clients. The rep is entered in the new station record. If no free station number is available, no new station record is created.
- A communication connection (C1) server -> offline client is created.
- A communication connection offline client -> server is created.
- A **Login** record (US) and a **Central Login Configuration** record (ES) is created.
- The **Initialize offline station** field in the ID record is set to true. (Once the initial data stock download file has been created for the offline client, the **Initialize offline station** field is reset to false.)

For more details, see Adding Offline Users in the *Aurea CRM win Administrator Guide*.

For a complete list of default and fallback values, see Default Values and Fallback Values for Offline Users and Stations in the *Aurea CRM win Administrator Guide*.

The next time you output communication files for the offline station, an initial data stock download is automatically created according to the defined communication format. You do not need to run `mmco.exe` via command line using the `-u` parameter in this case. for further details, see Configuring Offline Stations in the *Aurea CRM win Administrator Guide.*

CRM.designer Configurations for Offline Clients

Learn about CRM.designer configurations for offline clients.

In offline mode, the CRM.designer configuration is loaded from a local set of gzipped XML files (`*.gz`) located in the folder `..\data\offline\configuration` (default) of the Aurea CRM web offline installation.

The location is configurable via the `<XMLStoragePath>` tag in the `settings.xml` file, see settings.xml on page 26.

Note: All XML files are read-only. Therefore users cannot change their CRM.designer configurations while in offline mode, see Aurea CRM web offline Functionality on page 9.

For each configuration contained in the user's configuration hierarchy there is one `*.gz` file.

For the offline client to work, at least the file containing the UPDATE_DEFAULT configuration (`Config_0.gz` or `Config_0.xml`) must be available.

The default configurations UPDATE_DEFAULT (`Config_0.gz`) and UPDATE_DEFAULT Offline (`Config_6.gz`) are automatically deployed by the Aurea CRM web offline setup.

The client-specific configurations are automatically downloaded from the server and copied to `< XMLStoragePath >` when initializing the offline client. For further details, see Initializing Offline Clients on page 36.

Note: Aurea recommends creating and deploying parent configurations used by all clients as global patches. For further details, see Manually Providing Configurations for Offline Clients. Otherwise these `*.gz` files would be created multiple times on the server, thus occupying a lot of disk space.

Each time a user synchronizes his offline client, his CRM.designer configurations are automatically updated. For further details, see Synchronization.

Use the Web Configuration parameters `Offline.IncludeRootConfigurationInSync`, `Offline.RootConfigurationToSync` and `Offline.ShipDataModel` to determine which CRM.designer configurations are actually transferred to the offline clients during initialization/synchronization, see Web Configuration Parameters on page 21.

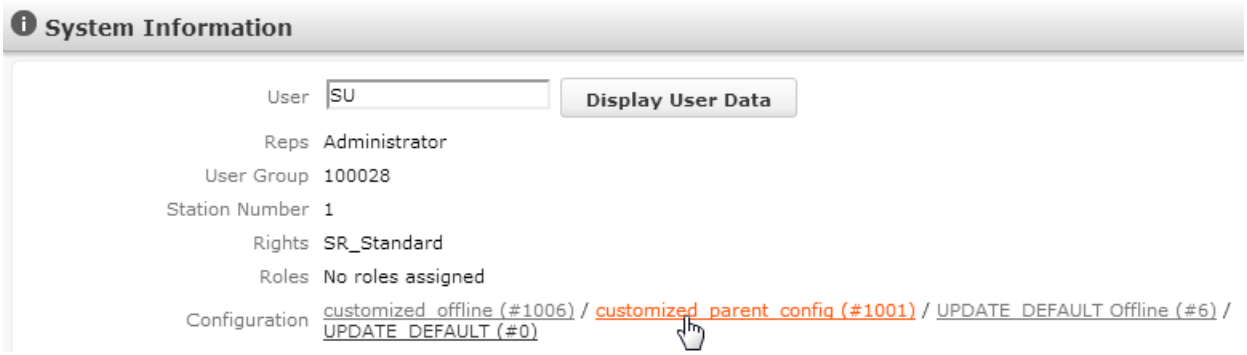
Note: Aurea CRM web offline does not check whether the downloaded configuration files have the same version number as the offline client. Therefore, make sure you do not transfer configurations from/to an outdated CRM.designer database.

Note: If you added a language on the server, you need to redistribute the new UPDATE_DEFAULT configuration (`Config_0.gz`) to the offline clients.

Manually Providing Configurations for Offline Clients

To manually create configuration files (`*.gz`), e.g. a custom parent configuration:

1. On the server navigate to Aurea CRM web's System Information page (**Settings** > **System Information**).
2. Enter the name of a **User** who has the desired configuration assigned.
3. Click **Display User Data**. The selected user's configuration hierarchy is displayed.



4. Click on a link to download a configuration. The configuration is downloaded as a *.gz file.
5. Copy the file to `< GlobalPatchDirectory >`, see Folder and File Preparation on page 30.

   The configuration is downloaded from the server and copied to `< XMLStoragePath >` when initializing the offline client, see Initializing Offline Clients on page 36.

Web Configuration Parameters

Web Configuration parameters available for Aurea CRM web offline.

The following Web Configuration parameters are available for Aurea CRM web offline. In Aurea CRM web these settings are available in the **System Configuration** (`ID_NET_SYSTEMCONFIGURATION`) on the **Offline** tab.

For information on setting Web Configuration parameters, see Web Configuration in the *Aurea CRM web Administrator Guide*.

Offline.IncludeRootConfigurationInSync

If checked, the root configuration specified in `Offline.RootConfigurationToSync` is downloaded as well, see Offline.RootConfigurationToSync.

Type: Checkbox

Default: Checked

Offline.MaxTimeLimitForSync

Defines the timespan (in hours) after which the user is prompted to synchronize again. The prompt is displayed on login and logout.

Type: Number

Default: 24

Offline.RootConfigurationToSync

Determines the root configuration for the initialization and synchronization of offline clients.

Every configuration in the configuration hierarchy between `Offline.RootConfigurationToSync` and the user's configuration is downloaded from the server during initialization/synchronization.

The root configuration itself is not included unless `Offline.IncludeRootConfigurationinSync` is checked. For further details, see Offline.RootConfigurationToSync.

Type: `Combo :Config`

Default: UPDATE_DEFAULT

Offline.ShipDataModel

If checked, the Aurea CRM data model saved in the CRM.designer database is included in the configuration file that is transferred to the offline client during synchronization.

Type: Checkbox

Offline.xml

You need to manually enter all required settings in the `Offline.xml` used by the server.

The `Offline.xml` file located in the `..\system\settings` directory contains configuration entries for the server, the offline client and CRM.launcher. It must be present on both the server and the offline client.

The `Offline.xml` used by the offline client is created during the setup process based on the settings in the `SilentClient.ini` file, see SilentClient.ini on page 27. For special customizations (e.g. `< WebRequestReadWriteTimeout >`) the file can also be prepared, see Folder and File Preparation on page 30.

Client and Server

Learn about Client and Server settings.

This section contains settings that apply to both an offline client and server:

<ThisMachinesRole>Server</ThisMachinesRole>

Defines the role of the machine where the `Offline.xml` is located. Available values: Client, Server.

For the clients, the value is automatically changed to "Client" during setup.

Default: Server

<RunCommunication>true</RunCommunication>

If set to `true`, the Aurea CRM win **Communication** module (`mmco.exe`) can be executed during synchronization.

Set this to `false` (on the server) if you want to deploy only CRM.designer configurations during synchronization.

Default: True

<CommunicationPath>c:\win\bb</CommunicationPath>

The path to the Aurea CRM win installation directory. Communication stores its output files here (in a sub-directory `..\<station number>`). For further details, see Directory Structure in the *Aurea CRM win Administrator Guide*.

For the client this path is automatically set during setup based on the value of `InstallDir` specified in the `SilentClient.ini`.

Default: `c:\Program Files\update.CRM\Aurea CRM web offline BTB\mm`

<CommunicationExecutablePath>c:\win\bb\system\exe\mmco.exe</CommunicationExecutablePath>

The path to the **Communication** module (`mmco.exe`).

For the client this path is automatically set during setup based on the value of `InstallDir` specified in the `SilentClient.ini`.

Default: `c:\Program Files\update.CRM\Aurea CRM web offline BTB\mm\system\exe\mmco.exe`

<DeleteCommunicationFilesAfterTransfer>true</DeleteCommunicationFilesAfterTransfer>

If set to `true`, the communication files are deleted after they have been transfered.

Default: true

<CommonClientUrl>http://localhost:7575</CommonClientUrl>

The URL of Aurea CRM web offline installed on the local machine (e.g. notbook). The URL must consist of the host name (`http://localhost`), a colon (`:`) and a port number (`7575`).

This value is automatically set during setup based on the value of `ClientURL` specified in the `SilentClient.ini`.

This value is used by CRM.launcher to set up the offline client and by the server to try and redirect the user to his offline station.

Server Only

Learn about Server Only settings.

The settings specified in the `CRM.server` tag apply to the offline server only:

<CommunicationImportArguments>-u: "{0}" -p: "{1}" -k "ein,{2},l,q"</CommunicationImportArguments>

The command line arguments passed to `mmco.exe` on the server when inputting communication files from offline clients.

The following placeholders can be used:

`{0}` ... user name

`{1}` ... password

`{2}` ... station number

The values from the current session are used.

For information on the command line parameters, see *mmco.exe Parameters* in the Aurea CRM win Administrator Guide.

<CommunicationExportArguments>-u: "{0}" -p: "{1}" -k "aus,{2},l,q"</CommunicationExportArguments>

The command line arguments passed to `mmco.exe` on the server when outputting communication files to offline clients.

<CommunicationFileDownloadExcludePattern/>

Specify a regular expression used to exclude communication files from being transferred to the offline client. Use this tag to prevent the transfer of temporary communication files (`*.part`).

<CompressConfigurationFiles>true</CompressConfigurationFiles>

If set to `true`, the configuration files (XML) are compressed (`*. gz`) before being transferred to the offline client.

Default: true

<GlobalPatchDirectory>..\web.data\GlobalPatchDirectory</GlobalPatchDirectory>

The directory where global files intended for all clients are stored on the server, see Folder and File Preparation on page 30.

If empty, no global patches can be automatically distributed.

Client Only

Learn about Client Only settings.

The settings specified in the `<Client>` tag apply to the offline client only:

<ServerStationNumber>100</ServerStationNumber>

The station number of the Aurea CRM web server.

This value is automatically set during setup based on the value of `StationNumberServer` specified in the `SilentClient.ini`.

<ServerUrl>http://mywebserver.com/updateCRM_web</ServerUrl>

The URL of the Aurea CRM web server. This URL is used by CRM.launcher to download files from the server.

This value is automatically set during setup based on the value of `ServerURL` specified in the `SilentClient.ini`.

<HideOnlineLink>false</HideOnlineLink>

If set to `true`, the **Navigate to the online server** button on the Offline Manager page is hidden after synchronization.

Default: false.

<CommunicationArguments>-u: "{0}" -p: "{1}" -k "ein,{2},l,q" -k "aus,{2}"</CommunicationArguments>

The command line arguments passed to `mmco.exe` on the client when communicating with the server.

Use the `COMUserName` and `COMUserPassword` settings in the `SilentClient.ini` to specify a dedicated COM user (e.g. if your offline users have no module rights for `mmco.exe`). If defined, setup overwrites `{0}` and `{1}` with the values of `COMUserName` and `COMUserPassword`.

<WebRequestReadWriteTimeout>100</WebRequestReadWriteTimeout>

The read/write timeout in seconds. This setting determines the number of seconds the client waits for a read/write operation to complete. **Default**: The default timeout of the .NET framework.

<WebRequestTimeout>100</WebRequestTimeout>

The timeout for web requests in seconds. This setting determines the number of seconds the client waits until the first byte of the response arrives from the server.

Default: The default timeout of the .NET framework.

<WebRequestRetryTimeout>1</WebRequestRetryTimeout>

The time in seconds the client waits before it retries a failed request.

Default: 1.

<MaxWebRequestRetries>10</MaxWebRequestRetries>

The maximum number of retries made for a single request before an exception is raised.

Default: 10.

<UploaderSettings>

The `< UploaderSettings >` tag contains settings used for the upload: Authentication mode, request timeout and read/write timeout. For details, see the comments in the default `Offline.xml` file.

Launcher Only

Learn about Launcher Only settings.

The settings specified in the `< update.launcher >` tag are used for initializing the offline client:

<IsInitialized>false</IsInitialized>

Indicates whether the offline client has been initialized:

After successful initialization, the value is automatically set to `true` by CRM.launcher.

<ClientStationNumber></ClientStationNumber>

The station number of the offline client.

During initialization, this value is automatically set to the station number entered in the rep record (ID), see Defining Users as "Offline Users".

Alternatively you can explicitly set the client's station number during setup by specifying a value for `StationNumberClient` in the `SilentClient.ini`.

<CommunicationArguments>-u: "{0}" -p: "{1}" -k "ein,+,{2},q"</CommunicationArguments>

The command line arguments passed to `mmco.exe` on the client when inputting the initial data stock download during initialization.

settings.xml

Learn about configuration of settings.xml for preparing Aurea CRM Server.

Aurea CRM web's `settings.xml` file contains the connection string to the CRM.designer database:

```
<update.configuration>
    <OleDbDsn>&designerDsn;</OleDbDsn>
    <Vertical>&VERTICAL;</Vertical>
</update.configuration>
```

Since offline clients do not read their CRM.designer configurations from a CRM.designer database but from (read-only) local XML files (`*.gz`), the `<OleDbDsn>` tag in the client's `settings.xml` must be disabled.

The location of the local configuration files is defined by the `<XmlStoragePath>` tag instead:

```
<update.configuration>
    <!-- <OleDbDsn>&designerDsn;</OleDbDsn> -->
    <Vertical>&VERTICAL;</Vertical>
    <XmlStoragePath>c:\Program Files\update.CRM\Aurea CRM web offline
BTB\web\data\offline\configuration</XmlStoragePath>
</update.configuration>
```

Note: The path must be absolute in order to be found by CRM.launcher.

These changes are performed during setup. `<XmlStoragePath>` is set based on the value of `InstallDir` specified in the `SilentClient.ini`.

For information on how to deploy a custom `settings.xml`, see Folder and File Preparation on page 30.

SilentClient.ini

Edit the `SilentClient.ini` file to provide all necessary settings for the installation of Aurea CRM web offline.

You can use the following placeholders for path-related variables:

• `%PROGRAM_FILES%`	the default `Program Files` directory
• `%APPDATA%`	the `Application Data` folder in the user directory
• `%LOCAL_APPDATA%`	the `Local Settings\Application` folder in the user directory
• `%COMMON_APPDATA%`	the `All Users\Application` Data directory
• `%USER_DOCUMENTS%`	the user's `My Documents` directory
• `%USERNAME%`	the (Windows) user name

Use `%PROGRAM_FILES%` (e.g. for `InstallDir`) to specify the default folder in the client's system language (`c:\Program Files` for English, `c:\Programme` for German etc.)

Contents of the default SilentClient.ini file

Mandatory options are highlighted. Use `;` to comment out a line.

```
-----------------------------------------------------------------
; Aurea CRM web offline client silent setup configuration file
;
; IMPORTANT NOTES:
;
; Internationalization:
; Use %PROGRAM_FILES% for the parameters InstallDir and DatabasePath if you
; want to use the default folder corresponding to different languages
; (e.g. "c:\Program Files" for English and "c:\Programme" for German")

[INSTALLATION]
InstallDir="%PROGRAM_FILES%\update.CRM"
SingleSignIn="N"

InstallLocalDB="Y"
InstallIISExpress="Y"

[DATABASE]
; This section defines all customizable database parameters.

InstanceName="MSSQLSERVER"
DatabasePath="%COMMON_APPDATA%\update.crm_webofflinedb"

TablePrefix="OFFLINE_WEB"

COMUserName=
COMUserPassword=
```

```
[STATIONCONFIGURATION]
StationNumberClient=100
StationNumberServer=1
ServerURL=http://mywebserver.com/updateCRM_web
ClientURL=http://localhost:7575

[OPTIONS]
; CreateShortcut="Y" creates a Menu Entry for all users allowing them to start the
 launcher
CreateShortcut="Y"

OmitFinishDialog="N"
PrepareSilentDeinstallation="N"

; LogPath=Creates the offlineSetup.log in the specified folder. The default value
 is C:\LogPath=C:\
```

Notes:

- `InstallDir`: Target directory for the Aurea CRM web offline installation.
- `DatabasePath`: Path to the local database file.

Note: The user must have full access to these directories.

Aurea recommends specifying a dedicated target directory for `InstallDir` and `DatabasePath` (e.g. `InstallDir =C :\ update.CRM`) and defining sufficient access rights for this directory: This ensures that users working with Aurea CRM web offline are able to create and modify files and folders (Write/Modify rights).

- `StationNumberClient`: Used as fallback if the user has no offline station defined in the database.
- `COMUserName /Password`: Specify a dedicated communication user for the client (e.g. because the offline user has no module rights for `mmco.exe`). Enter user name and password as plain text.
- `PrepareSilentDeinstallation`: Set this option to `"Y"` if you want to be able to silently uninstall Aurea CRM web offline. For further details, see Uninstalling on page 75.

update.launcher.exe.config

The settings specified in the `update.launcher.exe.config` configure execution of CRM.launcher on the client.

To use other than the default values you need to deploy the file. For further details, see Folder and File Preparation on page 30.

The following settings are available:

<LogPath>%PRODUCTFOLDER%\log</LogPath>

The log file directory for Aurea CRM web offline, CRM.launcher and IIS Express, see Log Files on page 65.

<MaxRetries>2</MaxRetries>

The maximum number of retries for requests, e.g. check connection to server.

If set to `0`, no retry requests are sent to the server.

If undefined, the default value `2` is used.

<RetryDelay>5</RetryDelay>

The time in seconds, CRM.launcher waits between requests.

If undefined, the default value `15` is used.

<CheckForUpdatesInterval>7</CheckForUpdatesInterval>

The time in days, after which CRM.launcher checks for new updates.

If set to `0`, CRM.launcher never checks for updates.

If undefined, the default value `7` is used.

<UserNameRequired>true</UserNameRequired>

If set to `true`, a user name is required for the communication between CRM.launcher and the server (e.g. requesting the file list from the server during the patching process).

If single sign-in is configured, this setting is ignored and the Windows credentials are used instead, see Configuring Single Sign-On on page 35.

<PasswordRequired>true</PasswordRequired>

If set to `true`, a password is required for the communication between CRM.launcher and the server.

If single sign-in is configured, this setting is ignored and the Windows credentials are used instead.

<EnableManualUpdates>false</EnableManualUpdates>

If set to `true` (default), the **Check for updates...** menu item is available in CRM.launcher's context menu enabling users to manually check for updates.

If set to `false`, only automatic update checking is available and performed in the interval specified in the `< CheckForUpdatesInterval >` tag.

<PatchingSettings>

This section configures the execution of patches on the client. For details, see the article “How to configure patching for CRM.launcher” at https://support.aurea.com.

<Logging>

For information on configuring the logging for CRM.launcher, see the article “How to configure logging for CRM.launcher” at https://support.aurea.com.

Folder and File Preparation

Folder and File Preparation on Aurea CRM Server.

Directory Structure - Example

```
<GlobalPatchDirectory>\8.1.770.60227_bb_release_20130624.zip
<GlobalPatchDirectory>\Config_1001.gz
<GlobalPatchDirectory>\global.ps1
<GlobalPatchDirectory>\my_global_patch.exe
<GlobalPatchDirectory>\images\new_image.png
<GlobalPatchDirectory>\styles\custom.theme\constants.css
<CommunicationPath>\<offlineStationNumber>\out\20130714.0
<CommunicationPath>\<offlineStationNumber>\patches\my_special_patch.exe
<CommunicationPath>\<offlineStationNumber>\patches\special.ps1
```

Prepare the following directories and files on the server to be used by the Aurea CRM web offline setup and the initialization process:

- Create the folder structure as specified by the `< CommunicationPath >` and `< GlobalPatchDirectory >` tags in the `Offline.xml`, see Offline.xml on page 22.
- Create initial data stock downloads for the offline clients, see Defining Users as "Offline Users". The communication files are output to `< CommunicationPath > \{ offlineStationNumber }\out`.

  You can compress large communication files using gzip (`*. gz )`.
- If you want to deploy custom `Offline.xml` and `settings.xml` files, you need to copy them to `< GlobalPatchDirectory >\system\settings`.

  If you only use the settings that are provided by the `SilentClient.ini` file, the `Offline.xml` and `settings.xml` files are automatically created/updated during setup.
- If your SU and WWW users have passwords, you need copy the `users.xml` file to `< GlobalPatchDirectory >\system\settings` as well.
- If needed (e.g. to deploy custom scripts, images etc. used by Aurea CRM web), copy additional files (uncompressed or `*.zip`) to the appropriate directories:
  - global files -> `< GlobalPatchDirectory >`
  - client-specific files -> `< CommunicationPath >\{ offlineStationNumber }\patches`

  Inside these directories you need to use the same directory structure as your (future) Aurea CRM web offline installation, i.e. to deploy custom images, copy them to `< GlobalPatchDirectory >\images`.

Note: When creating `*.zip` files make sure that all files are compressed using the DEFLATE algorithm. CRM.launcher can not handle files compressed with DEFLATE64.

Note: Windows automatically uses DEFLATE64 when zipping files larger than 2000 MB (uncompressed size).

Setting Up Offline Clients

The setup of Aurea CRM web offline is executed in silent mode enabling users to install Aurea CRM web offline on the client without any user interaction.

The `SilentClient.ini` file containing all necessary settings must be present in the installation package, see SilentClient.ini on page 27.

To execute the setup, the user must be local admin. The user must have full access to the installation path.

All steps executed by the setup are logged in the `OfflineSetup.log` file (located at the path specified by the value of `LogPath` in the `SilentClient.ini`.

The setup consists of the following:

- all required 3rd party products, see Installing 3rd-Party Products.
- Aurea CRM win (reduced installation)
- Aurea CRM web
- `update.Launcher.exe`
- `Encrypt.exe`
- default `Offline.xml`
- `Config_0.gz` (UPDATE_DEFAULT) and `Config_6.gz` (UPDATE_DEFAULT Offline)

The below sections in this chapter describe the process of setting up offline clients.

For more information, see “HOWTO Setup an offline client” on the Aurea support site: https://support.aurea.com.

System Requirements

You can find all system requirements (hardware, software, disk space, access rights, supported browsers etc.) at https://support.aurea.com (web offline section).

Setup Types

The setup can be executed in the following variations:

- silent setup with on-screen progress information, see Setup with Progress Information.
- silent setup without any on-screen information except error messages, see Setup without Progress Information.

Note: This installation setup may require administrator rights.

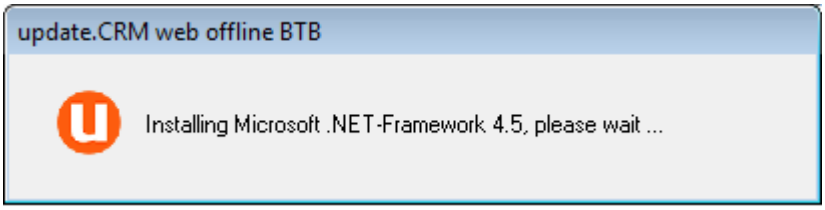
Setup with Progress Information

The user starts the setup by double-clicking `setup.exe`.

If setup is started on an unsupported operation system (see System Requirements), a message is displayed and the error is written to the `OfflineSetup.log` file.

Installing 3rd-Party Products

Setup checks if the necessary 3rd-party products are installed on the client.



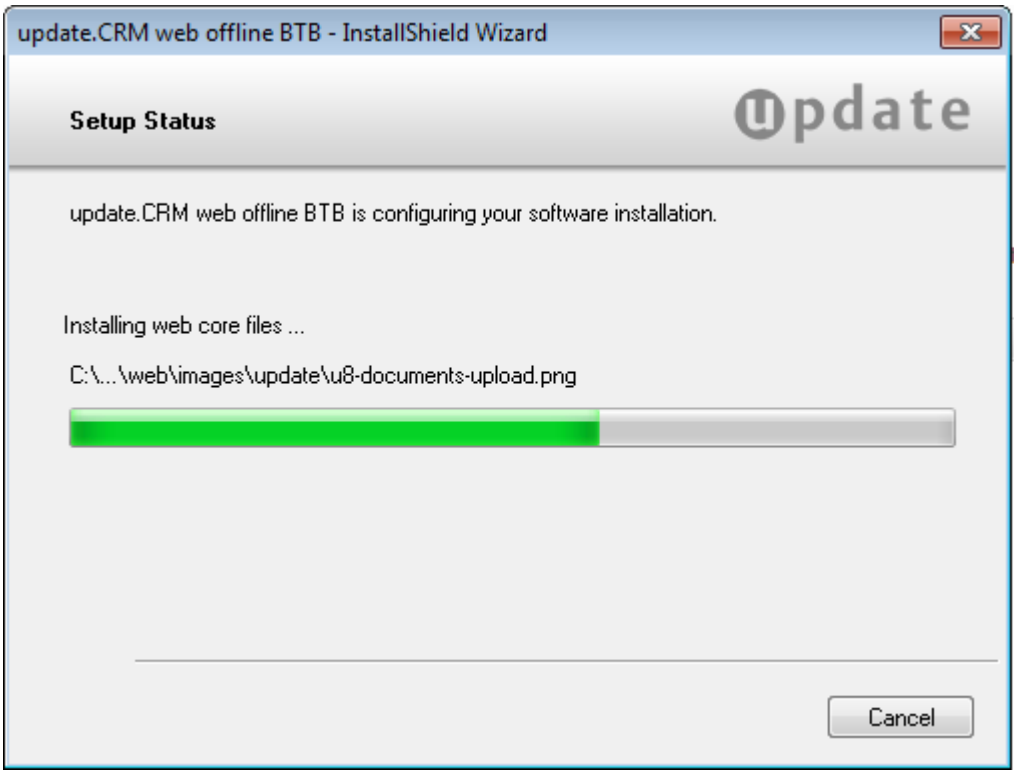
Setup checks for/installs the following 3rd-party products:

- .NET Framework 4.5
- SMO 2012
- MSI 4.5
- MSXML 6.0
- SQL Server 2014 LocalDB
- Web Server IIS Express 8.0
- SQL Server Native Client 11.0

If the installation of a 3rd-party product fails, a message is displayed and the error is written to the `OfflineSetup.log` file.

Copying Files and Creating the Local Database

After verifying/installing all required 3rd-part products, setup starts the file copy process.



The setup screen displays a progress indicator and information on the currently executed action.

The directory structure is created according to the value of `InstallDir` in the `SilentClient.ini` file.

The following defaults are used:

`c:\Program Files\ update.CRM \Aurea CRM web offline BTB\mm`

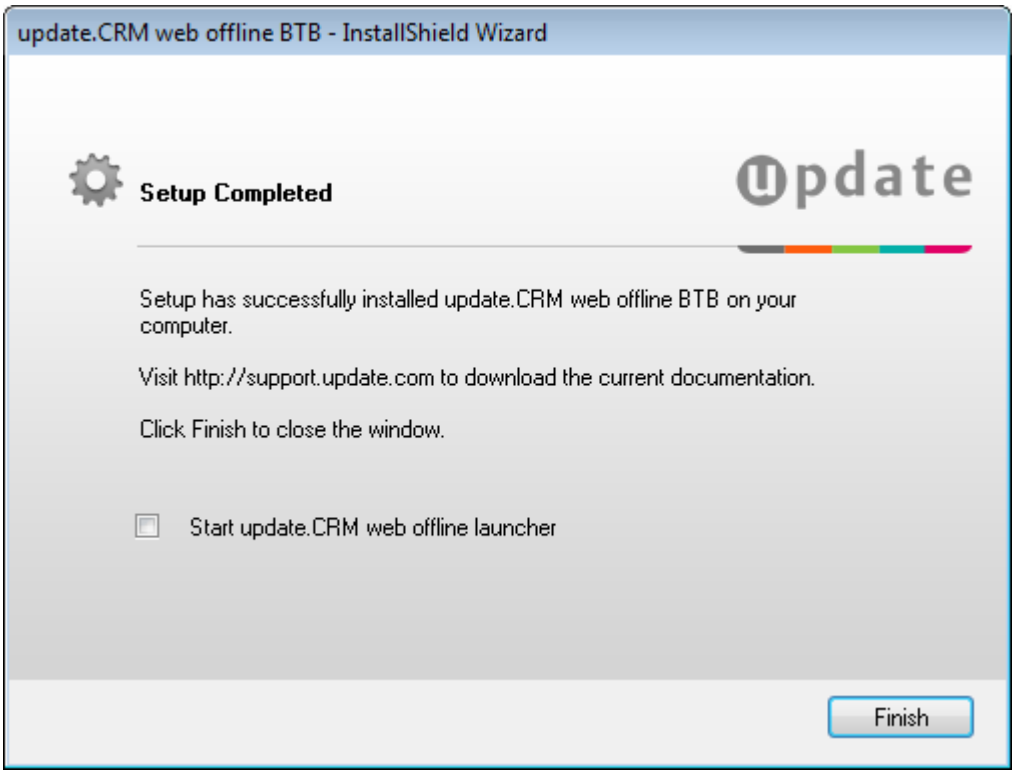
`c:\Program Files\ update.CRM \Aurea CRM web offline BTB\web`

`update.launcher.exe` and `update.launcher.exe.config` are copied to Aurea CRM web offline's root directory. The language files for CRM.launcher are copied to the `.. \Resources\Texts` folder.

`Offline.xml`, `settings.xml` and `users.xml` are updated in/copied to `.. \web\system\settings`.

Setup creates the Aurea CRM database and tables and enters the connection string in the `mmdb.ini` file according to the settings in the `SilentClient.ini` file (`[DATABASE]` section).

Setup Complete



Once the installation of Aurea CRM web offline is completed, a folder Aurea CRM web offline is added to the Start Menu. It contains CRM.launcher.

Users can choose to start CRM.launcher upon exiting the setup.

Setup without Progress Information

Run the setup.exe program from the command line shell with the parameter `-silent`, setup is executed without any on-screen information (except error messages).

The setup is executed as described under Setup with Progress Information.

For more command line options see Command line installation options on page 40.

Executing Setup in the User Context

If you want your users to execute the setup on their own, you need to complete the following preparative steps (on each client machine and in the `SilentClient.ini` file respectively):

1. Make sure all 3rd-party products are installed.
2. Create a destination folder to which the user has full access.
3. Both Aurea CRM web offline and the database must be installed to this folder (`InstallDir` and `DatabasePath` in the `SilentClient.ini` file).

   Installing to the default folder of Windows (e.g. `c:\Program Files`) is **not** supported.
4. Make sure `LogPath` points to a location where the user has write access.
5. Add the following option to the `[Installation]` section in the `SilentClient.ini` file:

```
UserInstallMode ="Y"
```

Note: Uninstalling Aurea CRM web offline requires administrator rights, therefore users can not uninstall Aurea CRM web offline themselves.

Configuring External User Authentication

Learn to configure External User Authentication to reuse CRM.Launcher Authentication Cookies.

CRM web offline frequently synchronizes it's database with CRM server. To ensure that the users connecting to CRM.Web from outside the intranet are properly authenticated and their data is updated and synchronized with web offline, you can configure CRM web offline to use the CRM.Launcher authentication cookies for the database synchronization. CRM web offline can use those same cookies to connect to the server without having to ask the user for credentials again. In order to accomplish this, the *"GetCookiesFromLauncher"* property has to be set on the **Offline.xml** configuration file.

See the sample below:

```
<OfflineManager>
    <Client>
        <GetCookiesFromLauncher>true</GetCookiesFromLauncher>
    </Client>
<OfflineManager>
```

Note: CRM.Launcher should also be configured to save the authentication cookies for them to be retrieved by CRM Web offline. For more information, see *Launcher Support for External Login via Firewalls* in the *CRM.Web Administrator Guide*.

Configuring Single Sign-On

Learn how to configure single sign-on for Aurea CRM web offline.

Aurea CRM web offline supports single sign-on. To enable single sign-on for Aurea CRM web offline:

- Set `SingleSignIn ="Y"` in the `SilentClient.ini` file.
- Enable Windows Authentication for IIS Express and disable anonymous authentication by configuring the `<anonymousAuthentication>` tag in the `web.data\applicationhost.config` file. See the sample configuration below:

```
<anonymousAuthentication enabled="false" userName="" />
<windowsAuthentication enabled="true">
```

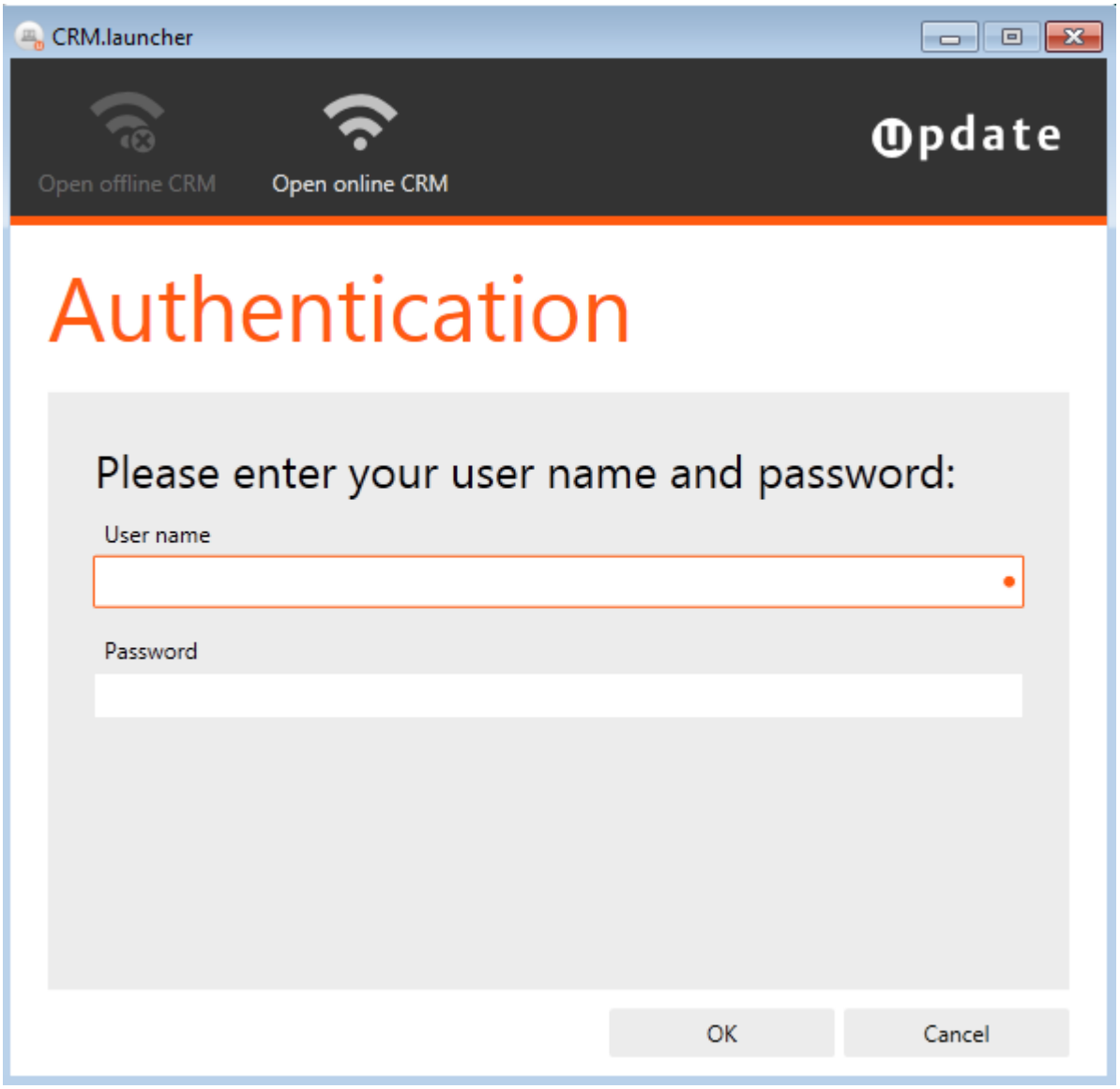
Note: Single sign-on must be configured for both Aurea CRM web and Aurea CRM web offline, see *Configuring Single Sign-On* in the *Aurea CRM web Installation & Technical Guide*. If setup is executed within the user context, single sign-on can not be enabled during setup. You need to edit the `applicationhost.config` file (`<Authentication>` tag) **before** setup is executed.

Initializing Offline Clients

Learn how to initializing Offline Clients for Aurea CRM web offline.

Once Aurea CRM web offline has been installed on the client, users can initialize Aurea CRM web offline using CRM.launcher (**Check for updates** context menu option). For further details, see CRM.launcher on page 52.

CRM.launcher checks the connection to the server and prompts the user to enter his credentials.

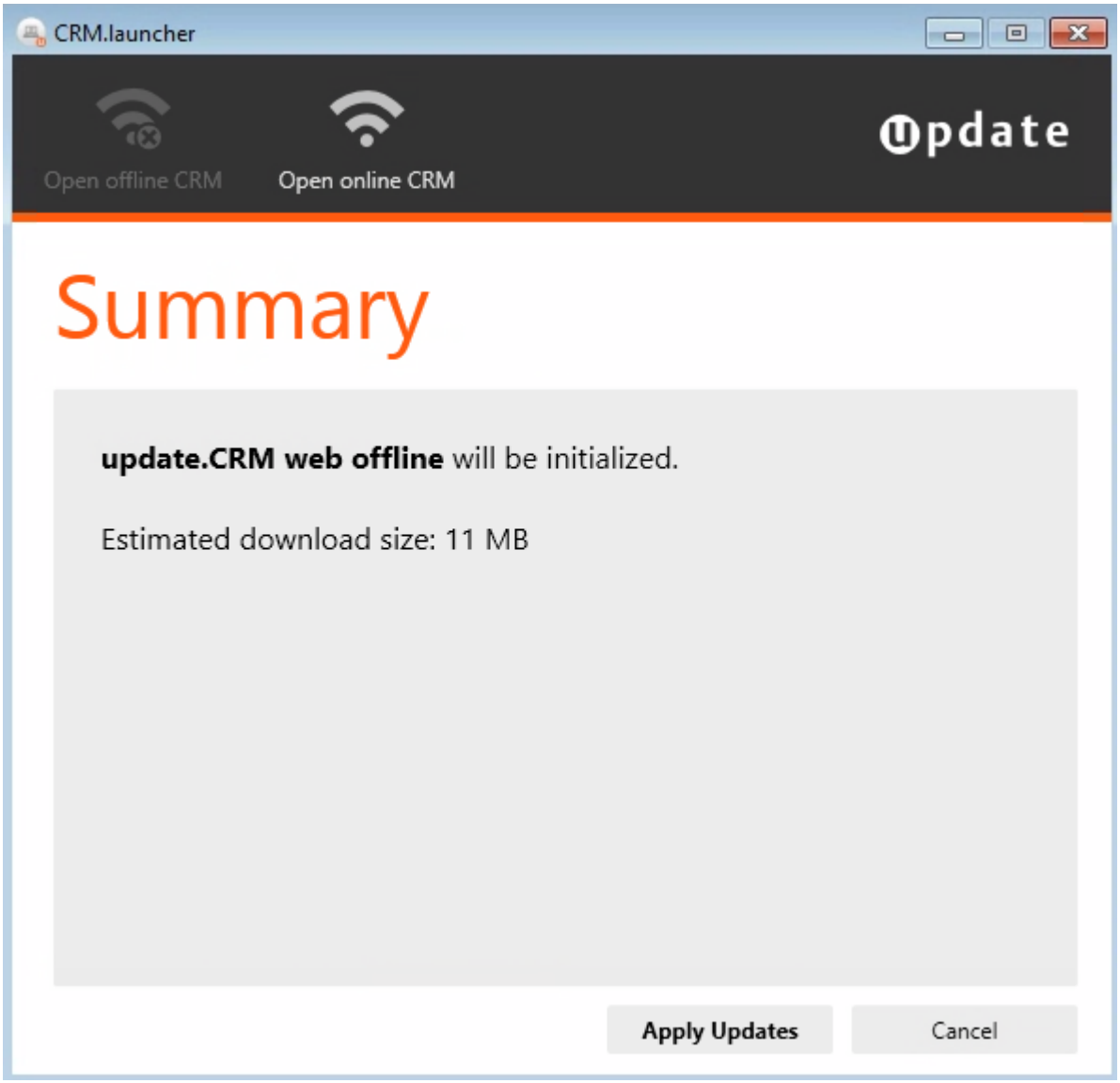


CRM.launcher checks if the user is a valid Aurea CRM web offline user, i.e. if a station number is entered in his rep record or in the `Offline.xml` file (`< ClientStationNumber >` tag).

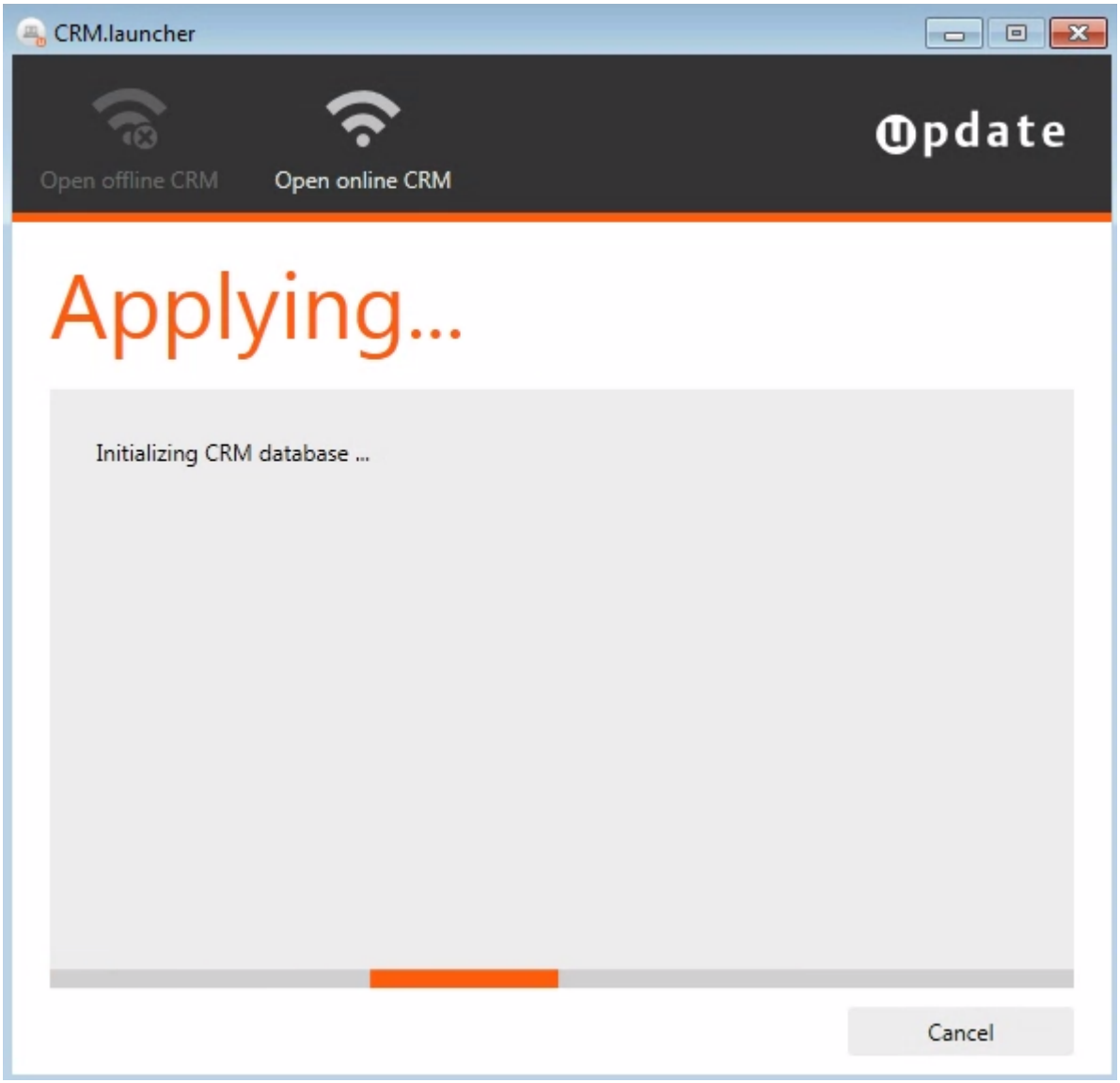
The initialization process consists of the following steps:

- creation of CRM.designer configuration files on the server
- download of files from the server:
  - the initial data stock download
  - the CRM.designer configuration files
  - global patches
  - client-specific patches

- file transfer and patch installation
- input of the initial data stock download



The user needs to start initialization by clicking on **Apply Updates**.



Once the initialization process is complete, CRM.launcher sets the `< IsInitialized >` tag in the `Offline.xml` to `true`. CRM.launcher automatically starts the IIS and loads the client URL of Aurea CRM web in the client's default browser.

If the initialization process fails, the error is documented in the `launcher.log` file, see Log Files on page 65.

Command line installation options

This topic explains the Command line installation options.

The web offline installer and the CRM.launcher accept the following command line options:

- --silent —This option is available for both CRM.launcher and web offline installers. It ensures that the installation is performed without presenting any User Interface. The CRM.launcher is installed in single user mode.
- --MigrateLauncher —This option is available only for web offline installer. Use this option to upgrade the CRM.launcher along with upgrading web offline.
- --Vertical —This option specifies the vertical for which the web offline is upgraded. It takes one of the following values: BTB; FS; OTC. For example, `–Vertical=OTC`.
- --i —This option specifies the path to the customized `SilentClient.ini` file. For example, `– i =<path to the SilentClient.ini file>`. If this option is not specified then the installer looks for the `SilentClient.ini` file in a temporary folder under `% appdata %` folder which the installer creates itself.

3

Upgrading

This topic explains how to upgrade Aurea CRM Web Offline to latest version.

The CRM.launcher can be configured to automatically apply patches to the web offline client or use the full installer to upgrade Web offline to the latest version.

CRM.launcher searches for new patches or installers in the `<GlobalPatchDirectory>` on the server (default path value: `..\web.data\GlobalPatchDirectory`). The `<GlobalPatchDirectory>` is defined in the `offline.xml` file located at `..\system\settings`.

The configuration parameter `FixedForceResume` is placed under the `<update.launcher>` section in the `crm.Launcher.exe.config` file. Setting this parameter value to `true` resets the status of the last patch to false, on starting CRM.launcher. This forces the CRM.launcher to start the upgrade installation of CRM weboffline as a part of the launcher upgrade process.

The setup installer shuts down the CRM.launcher and re-launches the installer after the web offline installation is complete. On restart, the CRM.launcher checks the value of the last patch and sets the value of configuration parameter `FixedForceResume` to `false`.

Note: The same procedure also upgrades the CRM.launcher if upgrades are available.

Upgrading using the Web Offline Installer

This topic explains how to upgrade Aurea CRM web offline to the latest version.

To upgrade your Aurea CRM web offline installation to the latest version, perform the following steps:

1. Unzip the installation package (`<vertical>_ weboffline _<version number>.zip`) to a local drive. `< vertical >` can be BTB, FS, or OTC.
2. In the unzipped folder, double click **setup.exe**. The **Upgrade setup** wizard displays.

   Note: This installation setup may require administrator rights.

   ---

3. Click **Next** to start the upgrade process. The **Setup Status** window displays upgrade installation.

4. When the upgrade installation is complete, the **Update Complete** window displays.
5. Click **Finish** to complete the upgrade installation.

Upgrading using the Silent Install Mode

Learn to upgrade using the silent install mode.

You can configure CRM web offline and CRM.launcher to perform an upgrade using the latest installer by configuring the patch settings in the `offline.xml` file for the client. The CRM Web Offline setup executable file should be placed in the global patch directory and it is correctly downloaded by CRM.launcher and executes the installer for the latest version in silent mode. For more information on how the patching process works, see Patching CRM.launcher and Aurea CRM web offline on page 54.

The server patching or upgrading settings for CRM.launcher are configured by adding `< PatchingSettings />` element under the `< update.launcher >` element in the `offline.xml` file for the client. The `< PatchingSettings />` element can contain any number of `<Setting/>` elements.

Each `<Setting/>` element can contain the following elements:

- `< LogProcessOutput />`: defines if the standard output and error output should be written to a file.
- `<Parameter/>`: defines the parameter(s) for the execution
- `< SuccessExitCodes />`: defines what are the success exit codes for the execution

---

Note: The order of the `<Setting/>`, `< LogProcessOutput />` and `<Parameter/>` element is important as CRM.launcher uses the first one which matches the filter and conditions. So the most specific `<Setting/>` element should be the first in the list.

To configure the server patch settings you have to place the following elements in the `offline.xml` file for the client:

- Setting element
- LogProcessOutput element
- Parameter element
- SuccessExitCodes element
- Arguments element
- Repair and Non-Repair mode
- Powershell Scripts

The remaining sections in this chapter provide details on configuring different setting elements in the `offline.xml` file. For more information, see the wiki article “HOWTO Configure PatchingSettings for CRM.launcher” on the Aurea support site: https://support.aurea.com.

Setting Element

This topic specifies the patch and/or installer executable files the settings should be applied.

The following table provides a description of the attributes:

XML Attribute	Required	Description
fileNameFilter	True	Only apply the settings to files whose name matches this filter. Value is specified as a regular expression.
certificateDnsNameFilter	False	Only apply the settings to files whose signer name in the certificate matches this filter. Value is specified as a regular expression.

The Setting element can have the following child elements:

XML Element	Default Value	Required	Max. Occurrence
LogProcessOutput	true	false	2
Parameter	empty string	false	2
SuccessExitCodes	null	false	1
Arguments	null	false	1

The following sample configuration shows the attribute values with regular expressions that specify the following:

- The file extension is `.exe` AND
- The file is signed AND
- The DnsName of the certificate used to sign the file is 'Aurea software'

```
...
<Setting fileNameFilter="\.exe$" certificateDnsNameFilter="^update software AG$">
```

```
    ...
</Setting>
...
```

The following sample configuration shows the attribute values that matches only files with the `exe` extension:

```
...
<Setting fileNameFilter="\.exe$">
    ...
</Setting>
...
```

LogProcessOutput Element

This element defines if the standard output and error output of the process should be written to a file.

This element is optional. If this element is not defined, the standard output and error output is written to a file by default.

It supports the following XML attributes:

XML Attribute	Required	Description
repair	false	If true, the setting only applies if the file is started in repair mode. If false, the setting only applies if the file is started in non-repair mode. If the attribute doesn't exit, the setting applies to both modes (repair and non-repair)

Examples

In the following sample configuration, the `repair` attribute is not used and the output of executing any file with an `msi` extension is not be written to a file:

```
...
<Setting fileNameFilter="\.msi$">
    <LogProcessOutput>false</LogProcessOutput>
    ...
</Setting>
...
```

In the following sample configuration, the repair attribute is set to `false` and the output of executing the first `msi` extension file is not be written to a file. If the first run fails, then for the next patch run the process output is written to files.

```
...
<Setting fileNameFilter="\.msi$">
 <LogProcessOutput>false</LogProcessOutput>
 ...
</Setting>
...
```

In the following sample configuration, the repair attribute is set to `true` and the output of executing the first `msi` extension file is written to a file. The process output of *.msi patch runs where the previous failed is NOT written to files.

```
...
<Setting fileNameFilter="\.msi$">
    <LogProcessOutput>false</LogProcessOutput>
    ...
</Setting>
...
```

Parameter Element

This element allows the specification of parameters to be passed to the executable patch when it is called.

There should be no more than one parameter instance for the repair and non-repair calls. These elements are both optional. Below in this section is a list of placeholders that can be used to have patch-specific information inserted into the parameter string on execution of the patch.

The following attributes are supported:

XML Attribute	Required	Description
repair	false	If true, the setting only applies if the file is started in repair mode. If false, the setting only applies if the file is started in non-repair mode. If the attribute doesn't exit, the setting applies to both modes (repair and non-repair)

The following placeholders are supported:

Placeholder	Description
{FileName}	Full path and file name for the patch to be executed.
{LogFile}	Full path and file name for the log file of the patch.
{LogPath}	Full path to CRM.launcher's general log directory for patches.
{InstallationPath} V8.2 Feature Pack +	Full path to CRM.launcher's installation directory.

Note: All values for the placeholders are wrapped in double quotes by default.

Examples

In the following sample configuration, all files with the `.bat` patch are executed with '`> { LogFile }`' as argument.

In this case, the `{ LogFile }` is replaced with the log file name and path.

For example, if the patch file name is `testFile.bat`, the argument is: `> c:\offline\web.data\logs\PathLogs\testFile.201307191004.log`.

```
...
<Setting fileNameFilter="\.bat$">
    ...
    <Parameter><![CDATA[ > {LogFile}]]></Parameter>
    ...
</Setting>
...
```

In the following sample configuration, the `repair` attribute is set to `false` and has a placeholder {FileName}.

In this case, the first `*. msi` patch run is executed with '`/ i { FileName } /QN / norestart REBOOT= ReallySuppress`' as argument and `{ FileName }` is replaced with the patch file name and path.

For example, if the patch file name is `testFile.msi`, the argument is: `/i "c:\Users\UserA\AppData\Local\update.launcher\Download\testFile.msi" /QN /norestart REBOOT=ReallySuppress`

```
 ...
<Setting fileNameFilter="\.msi$">
    ...
    <Parameter repair="false">/i {FileName} /QN /norestart
REBOOT=ReallySuppress</Parameter>
    ...
</Setting>
...
```

In the following sample configuration, the `repair` attribute is set to `false` and has a placeholder `{ LogFolder }`.

In this case, only if the previous run failed, the `*test.msi` patch is executed with `--Log ={ LogPath }` as argument and `{ LogPath }` is replaced with the path to the global log directory.

For example, if the patch file name is `test.msi`, the argument is: `--Log="c:\offline\web.data\logs\PathLogs"`.

```
...
<Setting fileNameFilter="test\.msi$">
    ...
    <Parameter repair="true">--Log={LogPath}</Parameter>
    ...
</Setting>
...
```

SuccessExitCodes section

This element contains a list of error codes / return codes for each executable patch.

These values are used to determine if that patch's execution was successful. If no error code(s) are specified, the return code is ignored and the patch is always considered successful.

```
...
<Setting fileNameFilter="\.exe$">
    ...
    <SuccessExitCodes>
        <Code>0</Code>
        ...
        <Code repair="false">9</Code>
        ...
        <Code repair="true">10</Code>
        ...
    </SuccessExitCodes>
    ...
</Setting>
...
```

It can contain the XML element `<Code/>` which supports the following XML attribute:

XML Attribute	Required	Description
repair	false	If true, the setting only applies if the file is started in repair mode. If false, the setting only applies if the file is started in non-repair mode. If the attribute doesn't exit, the setting applies to both modes (repair and non-repair)

Examples

```
...
<Setting fileNameFilter="\.msi$">
    ...
    <SuccessExitCodes>
        <Code>0</Code>
        <Code repair="false">1</Code>
        <Code repair="true">2</Code>
        <Code>3</Code>
        <Code repair="false">5</Code>
        <Code repair="true">8</Code>
        <Code>13</Code>
        <Code repair="false">21</Code>
        <Code repair="true">34</Code>
    </SuccessExitCodes>
    ...
</Setting>
...
```

In the first run for `*.msi` patches the success codes are:

- 0
- 1

- 3
- 5
- 13
- 21

In the re-run of a `*.msi` (which failed the previous run) the success codes are:

- 0
- 2
- 3
- 8
- 13
- 31

MSI / MSP Exit Codes Description

Value	Exit Code	Description
0	ERROR_SUCCESS	Action completed successfully.
1641	ERROR_SUCCESS_REBOOT_INITIATED	The installer has started a reboot. This error code not available on Windows Installer version 1.0.
3010	ERROR_SUCCESS_REBOOT_REQUIRED	A restart is required to complete the install. This does not include installs where the ForceReboot action is run. Note that this error is not available until future version of the installer.

Arguments Element

This element contains a list of arguments for each PowerShell script.

These values are added as arguments to the call. The arguments can be all named or all without name. It's not possible to mix this setting. If arguments without name are configured the order in the <Arguments> section is used to add them to the call.

```
...
<Setting fileNameFilter="\.ps1$">
...
<Arguments>
  <Arg>0</Arg>
  <Arg>9</Arg>
  <Arg>10</Arg>
</Arguments>
...
</Setting>
...
```

It should contain the XML element <Arg /> which supports the following XML attribute:

XML Attribute	Re-quired	Description
repair	false	The name of the parameter. If the attribute doesn't exists, the value is used without a name.

XML attribute	Required	Description
key	false	The name of the parameter. If the attribute doesn't exists, the value is used without name

Examples

Example of a “not named parameter”

```
...
        <Setting fileNameFilter="\.ps1$">
        ...
        <Arguments>
        <Arg>0</Arg>
        <Arg>9</Arg>
        <Arg>10</Arg>
        </Arguments>
        ...
        </Setting>
        ...
```

This settings would result that all `*.ps1` files getting called as below:

```
> .\myfile.ps1 0 9 10
```

Example of a “named parameter”

```
...
        <Setting fileNameFilter="\.ps1$">
        ...
        <Arguments>
        <Arg key="arg1">0</Arg>
        <Arg key="arg3">9</Arg>
        <Arg key="arg2">10</Arg>
        </Arguments>
        ...
        </Setting>
        ...
```

This settings would result in all `*.ps1` files getting called as below:

```
> .\myfile.ps1 -arg1 0 -arg3 9 -arg2 10
```

Repair and Non-Repair mode

Learn about the Repair and Non-Repair mode.

The repair attribute is only valid for settings for *.msi or *.msp patches. For all other patches (*.exe, *.bat, *.cmd etc.), the launcher does not distinguish between repair and non-repair mode but always uses non-repair mode.

Repair mode

If the launcher tries to re-run an .msi or .msp patch that failed on a previous run, the launcher only applies the settings with repair attribute set to `true` or no repair attribute.

Non-repair mode

For the first run for *.msi or *.msp patches or all runs
for '*.exe, *.bat or *.cmd patches, the launcher only applies the settings with repair attribute set to `false` or no repair attribute.

Powershell Scripts

To enable the execution of Powershell (`*.ps1`) scripts, uncomment or add the PatchSetting for `*.ps1` files. A sample script without additional parameters.

```
<PatchingSettings>
  ...
  <Setting fileNameFilter="\.ps1$">
    <SuccessExitCodes>
      <Code>0</Code>
    </SuccessExitCodes>
  </Setting>
  ...
</PatchingSettings>
```

A sample script with additional parameters.

```
<PatchingSettings>
  ...
  <Setting fileNameFilter="MyFileNameThatNeedsParameter\.ps1$">
    <SuccessExitCodes>
      <Code>0</Code>
    </SuccessExitCodes>
    <Arguments>
      <Arg>My arg 1</Arg>
      <Arg>My arg 2</Arg>
      <Arg>My arg 3</Arg>
      <Arg>My arg 4</Arg>
    </Arguments>
  </Setting>
  ...
</PatchingSettings>
```

A sample script with NAMED parameters

```
<PatchingSettings>
  ...
  <Setting fileNameFilter="MyFileNameThatNeedsNamedParameter\.ps1$">
    <SuccessExitCodes>
      <Code>0</Code>
    </SuccessExitCodes>
    <Arguments>
      <Arg key="arg1">My arg 1</Arg>
      <Arg key="arg2">My arg 2</Arg>
```

```
      <Arg key="arg3">My arg 3</Arg>
      <Arg key="arg4">My arg 4</Arg>
    </Arguments>
  </Setting>
  ...
</PatchingSettings>
```

Note: The `<Setting />` with the most specific filter should be at the top.

4

Administration

Learn how to administer offline clients.

The following topics explains offline client administration:

CRM.launcher

CRM.launcher is used to control Aurea CRM web offline and the IIS on the client.

`update.Launcher.exe` is located in the root directory of the installation (as specified by `InstallDir`).

CRM.launcher allows users to:

- initialize the offline client when first starting to use Aurea CRM web offline. For further details, see Initializing Offline Clients on page 36.
- installing new versions of Aurea CRM web offline as well as other updates. For further details, see Patching CRM.launcher and Aurea CRM web offline on page 54.
- start Aurea CRM web in both online and offline mode. For further details, see Switching Between Online and Offline Mode on page 11.
- collect all files needed to submit a bug report (logs, configurations, database). For further details, see Creating a Bug Report on page 59.
- Authenticate users connecting to CRM Web offline in offline mode.

All steps executed by CRM.launcher are logged in the `launcher.log` file. For further details, see Log Files on page 65.

Starting CRM.launcher starts the client's local IIS. (Restarting CRM.launcher performs an iisreset.)

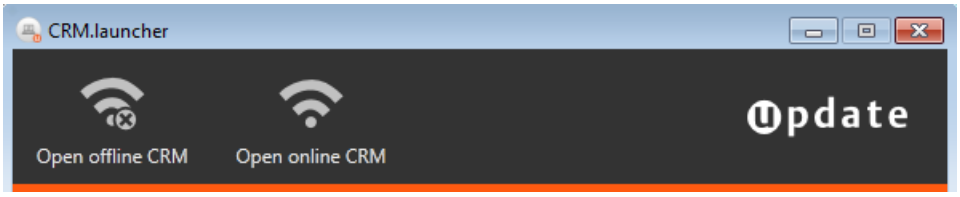
Note: When starting CRM.launcher for the first time a Windows security popup (UAC Firewall) may be displayed (depending on the security settings on the client).

At startup, CRM.launcher checks the server connection and performs a version check. If CRM.launcher itself and/or the Aurea CRM web offline version is not up-to-date, it prompts the user to upgrade. For further details, see Checking for Updates.

You can configure CRM.Web offline to authenticate client connections using the NT LAN Manager (NTLM) authentication protocol. To configure NTLM authentication, see Configuring NTLM Authentication for Offline clients on page 53.

Details on the current versions are saved in and read from the `launcher_telemetry.xml` file located in `%LocalAppData%\suite\CRM.launcher` (`<CurrentVersion>`, `<ServerVersion>` and `<LauncherVersion>` tags).

If the client's version is up-to-date, the user is prompted to choose between starting the online or offline version of Aurea CRM web:



For more information on CRM.launcher, see https://support.aurea.com/wiki/index.php?title=Launcher.

Configuring NTLM Authentication for Offline clients

This topic explains configuration for NTLM authentication.

When CRM.Launcher initiates the connection with CRM Web offline, it authenticates the user using the CRM.Web Offline login by default. If you want users to be authenticated with the NT LAN Manager (NTLM) authentication protocol then set the `<SendNtlmCredentials>` property to `true` in the `update.Launcher.exe.config` file. See the sample configuration below:

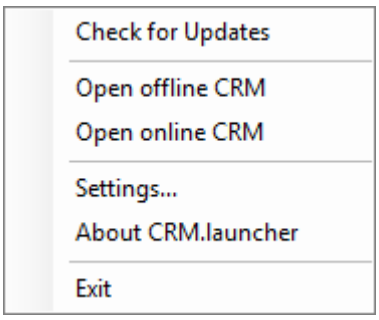
```
<update.net>
  <update.launcher>
    ...
    <SendNtlmCredentials>true</SendNtlmCredentials>
    ...
  </update.launcher>
</update.net>
```

Context Menu

Learn about the context menu in CRM.Launcher.

As long as CRM.launcher is running, an icon is displayed in the system tray:

Right-clicking on the icon opens the following context menu:



Available options:

- **Check for updates:** Checks if the version installed on the client is up-to-date, i.e. the same as on the server, and prompts the user to install patches, if available. For further details, see Patching CRM.launcher and Aurea CRM web offline on page 54.

  You can hide this menu item by setting `< EnableManualUpdates >` to `false` in the `update.launcher.exe.config` file.

  If the installation of a patch was aborted by the user, the first menu entry reads **Resume Patching** instead: This allows the user to resume the patching process.

- **Open online CRM:** Opens the client's default browser and navigates to Aurea CRM web on the server (`< ServerUrl >` in the `Offline.xml`).
- **Open offline CRM:** Opens the client's default browser and navigates to the client URL of Aurea CRM web (`< CommonClientUrl >` in the `Offline.xml`).
- **Settings:** The following options are available:
  - **General:** Allows users to select a language, see *CRM.launcher Languages* in the Aurea CRM web Administrator Guide.
  - **Troubleshooting:** Allows users to create a bug report, see Creating a Bug Report on page 59.
  - **Connectivity:** Allows users to set another port used for the connection between CRM.launcher and the browser (default: 18188).
- **Exit:** Exits CRM.launcher and stops the IIS.

Patching CRM.launcher and Aurea CRM web offline

CRM.launcher and Aurea CRM web offline installs the new patch automatically to upgrade.

CRM.launcher is used to automatically upgrade itself and Aurea CRM web offline on the client: It enables users to install new service packs, hotfixes and other patches, e.g. new template versions.

Note: Make sure to **test all patches before deploying them** to the offline clients; deploying faulty patches may lead to a lot of unstable offline clients.

CRM.launcher checks for new files in the following directories on the server:

- `< GlobalPatchDirectory >` for global files intended for all offline clients
- `< CommunicationPath >\{ offlineStationNumber }\patches` for client-specific files

For information on how to prepare the files, >> Folder and File Preparation on page 30.

Note: By default, only executables (`*.exe`) **signed by Aurea software** are executed. To enable the execution of other files (`*.exe`, `*. msi`, `*.ps1`), you need to change the `< PatchingSettings >` in the `update.launcher.exe.xml` file. For more information on parameters, see the article “How To Configure Patching Settings for CRM.launcher” at https://support.aurea.com.

Note: Running `*.bat` and `*. cmd` files is not supported since these files cannot be signed.

Note: Running executables and extracting zip archives is only done at **root level**, i.e. if you want CRM.launcher to run executables and unzip files, you must copy these files to the root level of `< GlobalPatchDirectory >` or `< CommunicationPath >\{ offlineStationNumber }\patches`.

Note: Executables are run in **silent mode**, therefore do not deploy any executables requiring user interaction.

Patches are **processed in alphabetic order** (executables first), global files are processed before specific files.

CRM.launcher patches are processed before all other patches (both from the global and client-specific directory). Use the following naming conventions:

```
launcher_any_to_8.2.1.2345.exe
```

```
launcher_from_8.2.7.1234_to_8.2.8.5678.exe
```

CRM.launcher patches are processed in **numeric order** (i.e. starting with the lowest `to_<version>` number).

If a CRM.launcher patch is found, CRM.launcher first patches and restarts itself and afterwards downloads and installs all other patches.

If the client has a CRM.launcher with a higher service pack than the server, initialization and patching are not possible.

Checking for Updates

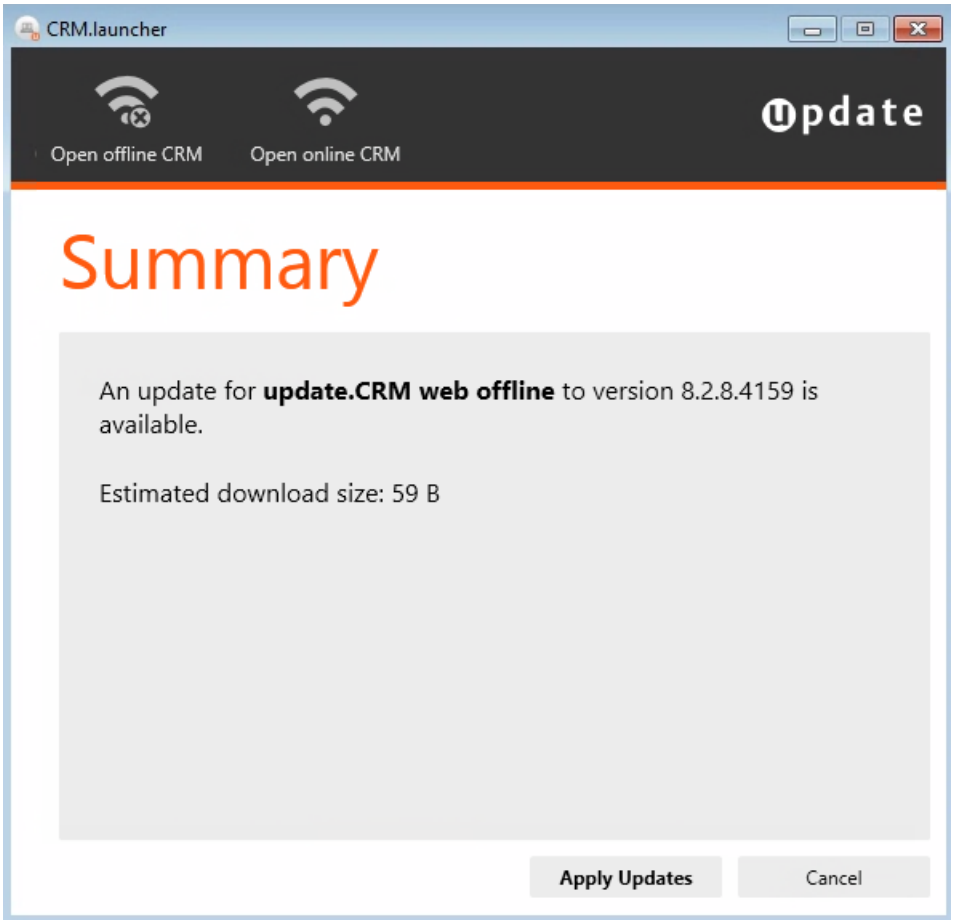
The check for new updates is performed as follows:

- Whenever CRM.launcher is started, it automatically performs a version check (service pack and hotfix numbers) and searches for available updates.
- In addition, CRM.launcher checks for new updates at regular intervals (default: 7 days) configurable via the `< CheckForUpdatesIntervall /> ` tag in the `update.launcher.exe.config` file. For further details, see update.launcher.exe.config on page 28. The interval is calculated based on the `< LastVersionCheck >` entry in the `launcher_telemetry.xml` file.
- Users can check for new updates manually by selecting **Check for updates** in the context menu.

If the Aurea CRM web offline version on the client is out-of-date and/or any updates are available, CRM.launcher informs the user as follows:

Scenario 1:

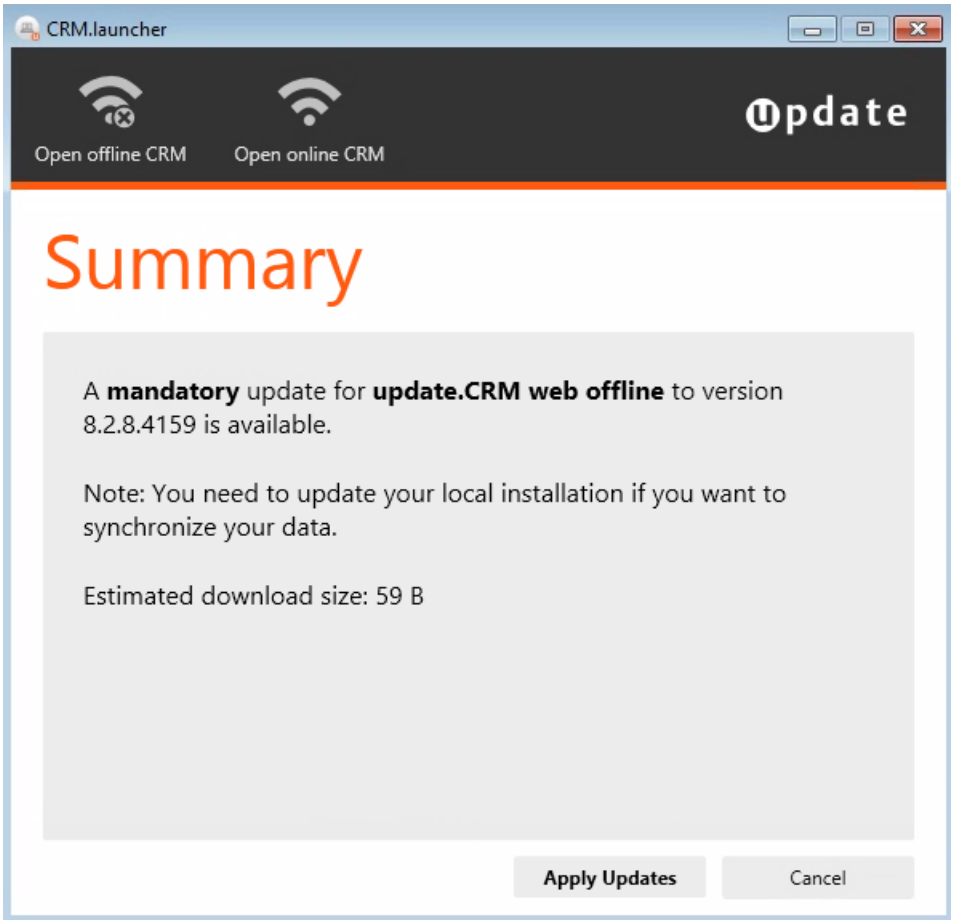
An optional update for Aurea CRM web offline (or CRM.launcher) is found on the server. The user can choose between installing the patch or starting Aurea CRM web offline.



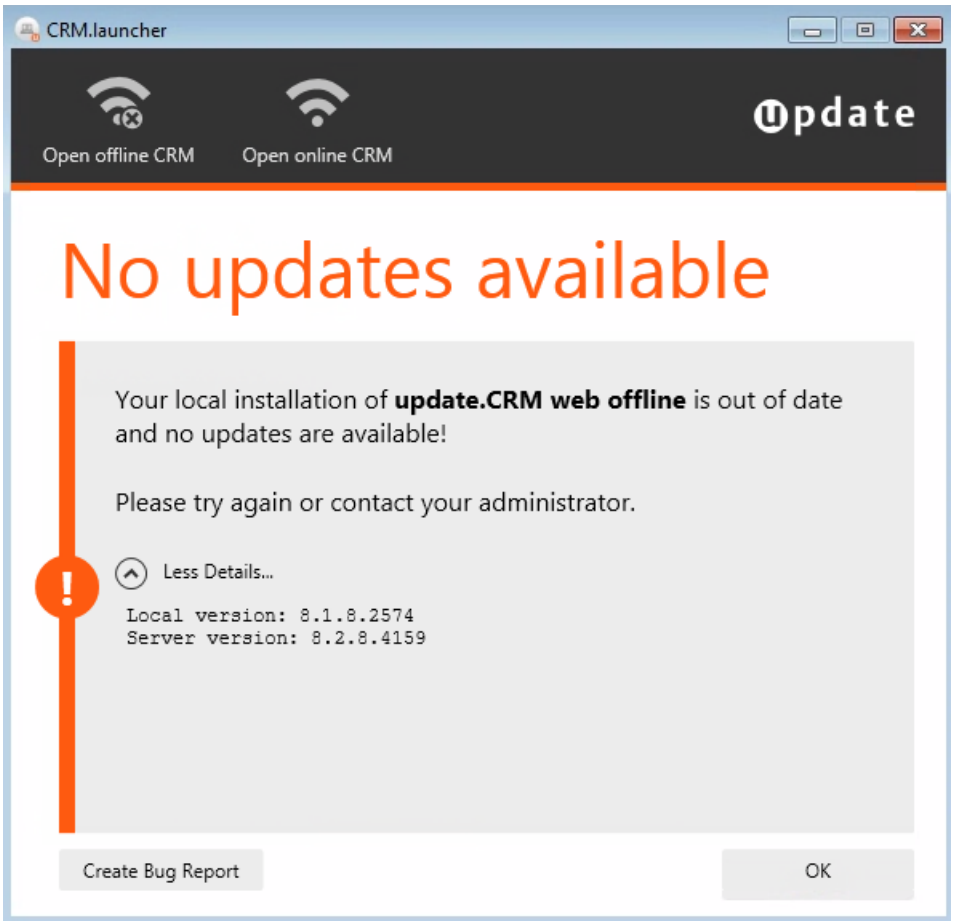
Scenario 2:

Server and client have different service pack (or version) numbers.

The user can choose between installing the patch or starting Aurea CRM web offline. The user is informed that he is not able to synchronize until the new patch is installed.



Note: If CRM.launcher detects an out-of-date version and does not find the appropriate files on the server, the user is informed accordingly:



Installing Patches

Users start the patching process by clicking **Apply Updates**.

Note: Since CRM.launcher stops the IIS while installing patches, patching is not possible during synchronization: If synchronization is currently executed, the user cannot install patches. If the user starts synchronization while a patch is being downloaded, patching is canceled and must be restarted after synchronization is complete.

CRM.launcher executes the following steps:

- download files from the server
- run executables
- extract zip archives

The files are downloaded to the user's `% LocalAppData %\ update.CRM \CRM.launcher\download` directory. If files in the global and the client-specific directory have the same name, only the client-specific file is downloaded.

The downloaded files are automatically deleted after patching.

Files that have been already downloaded to a client are logged in the station-specific `downloadAcknowledge.xml` (located in `<CommunicationPath>\{offlineStationNumber}`) on the server. CRM.launcher reads this file to identify which files to download.

Note: When acknowledging downloads, CRM.launcher checks for file name and modification date. If you distribute a new version of an already downloaded file, CRM.launcher downloads it.

The downloads are verified via MD5 hash. For troubleshooting information, see the article “Troubleshooting CRM.launcher 8.2 Feature Pack” at https://support.aurea.com.

Information on each processed file is written to the `patchmanifest.xml` file (located in `% LocalAppData %\ update.CRM \CRM.launcher\download` on the client).

If the user clicks **Cancel** while patches are being installed, the patching process is aborted. Users can resume patching by clicking **Resume Patching** (context menu option and button). Patching is resumed according to the information in the `patchmanifest.xml` file.

Note: Users can not start Aurea CRM web offline before the patching process is complete.

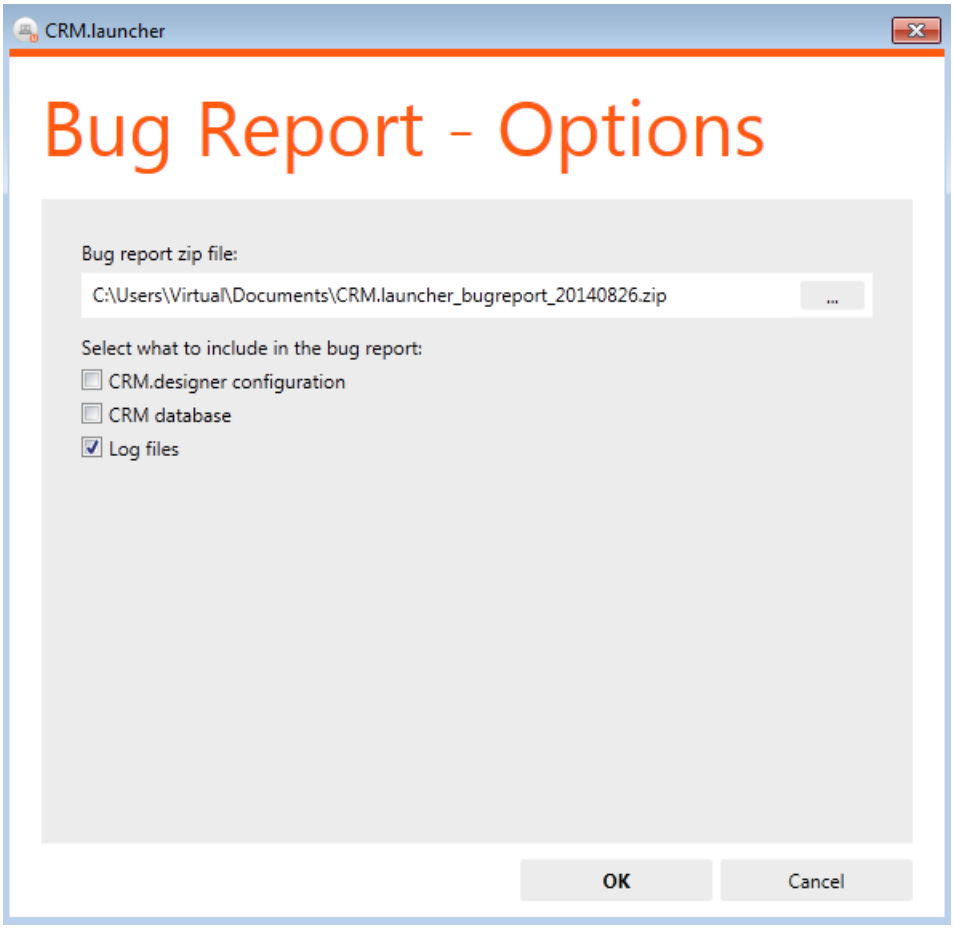
For further details on the patching process, see the article “CRM.launcher 8.2 Feature Pack at https://support.aurea.com.

For information on how to customize the patching process (logging, parameters, exit codes), see the article “How To Configure Patching Settings for CRM.launcher” at https://support.aurea.com.

Creating a Bug Report

Learn to create bug report from context menu.

By selecting **Settings** > **Troubleshooting** from the context menu users can create a `CRM.launcher_bugreport_<date>.zip` in the `..\Documents` folder (default) containing all files required to submit a bug report.



Available options:

- **Bug report zip file:** The name and location of the bug report.
- **CRM.designer configuration:** Downloads all CRM.designer configurations present on the client (`*.gz` in sub-folder `Config`). For further details, see CRM.designer Configurations for Offline Clients on page 19.
- **CRM Database:** Creates a backup of the offline client's local Aurea CRM database (`*.bak` in sub-folder `Database`).

- **Log files** (checked by default): Downloads the following log files:
  - All log files from the `< LogPath >` folder (in sub-folders `Logs`). For further details, see Log Files on page 65.
  - `u8_communication_<User>.log` files from `< CommunicationPath >` (in sub-folder `Logs\Communication`)
  - all patch logs (in sub-folder `Logs\ PatchLogs`)
- In addition to the user's selection from the above, the ZIP file contains a `Settings` sub-folder containing the following files (if available):
  - `applicationhost.config`
  - `history.xml`
  - `launcher_telemetry.xml`
  - `mmdb.ini` from the local Aurea CRM web's `..\system\sys` directory
  - `Offline.xml`
  - `patchmanifest _<date>.xml`
  - `settings.xml` from the local Aurea CRM web's `..\system\settings` directory
  - `update.launcher.exe.config`
  - `user.config`

  These settings files are only downloaded if found; if one or more of these files are not found, no message is displayed.

Offline Manager

Moving from online to offline mode.

The menu action `ID_NET_OFFLINEMANAGER` in `ID_NET_SERVICES` (**Settings** > **Offline Manager**) calls the Offline Manager page.

If this menu item is selected while in online mode, a message is displayed allowing the user to switch to offline mode.

On the Offline Manager page, the user can start synchronization by clicking **Offline Synchronization**.

Synchronization

Since offline clients use their local Aurea CRM database and configuration files, they need to synchronize their data with the Aurea CRM web server. During Synchronization:

- Changed CRM.designer configurations used on the client are transferred as (read-only) XML files (`*.gz`),
- Changes to the Aurea CRM database are exchanged between client and server via the Aurea CRM win **Communication** module.

When data is communicated between client and server, only delta information is exchanged to reduce the transfer volumes. A conflict resolution process ensures to communicate the latest changes, e.g. if two different reps modified the same field of the same record.

The user is prompted to start synchronization at regular intervals (default: 24 hours). The date and time of the last successful synchronization is determined based on the time stamp saved in the **Communication Log** info area (C3). For further details, see Communication Logs.

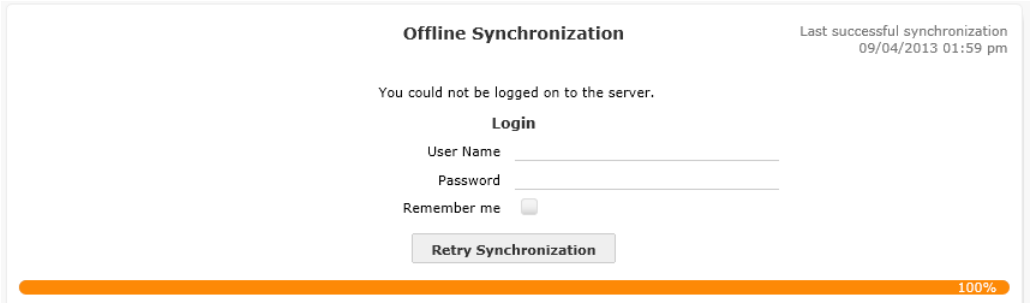
The synchronization interval is configurable via the `Offline.MaxTimeLimitForSync` Web Configuration parameter. For further details, see Offline.MaxTimeLimitForSync.

The synchronization prompt is displayed on login and logout when in offline mode.

The synchronization process can be started:

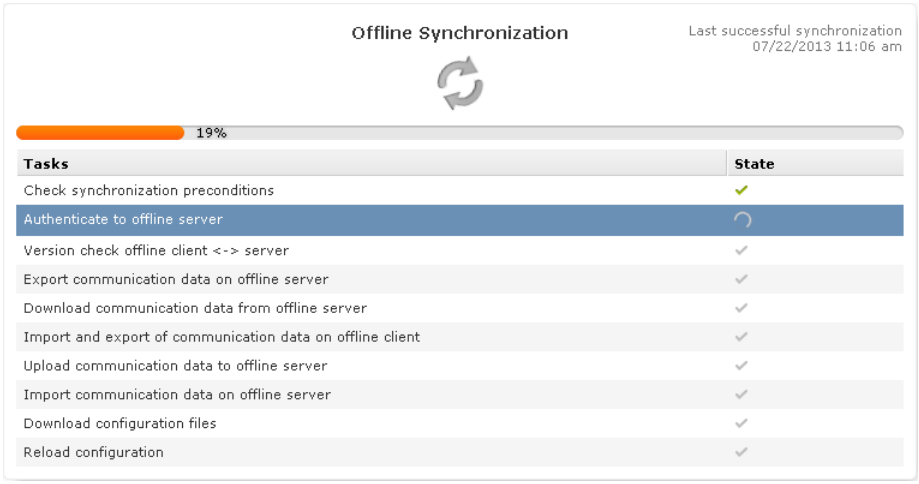
- by clicking **Yes** in the prompt,
- or by clicking on the **Offline Synchronization** button on the Offline Manager page.

When first starting the Offline Manager, users must provide their user name and password in order to be able to access the server.



The **Remember me** check box allows users to save their credentials in a browser cookie for all subsequent synchronization cycles.

See the below image for the task sequence of the synchronization process.



On the server you can display ongoing synchronization jobs in the Job Manager (**Administration** > **Job Manager**).

If the communication file received from the server contains data model changes, the user is prompted to restart CRM.launcher which then performs an iisreset.

After successfully completing the synchronization the user can choose between offline or online mode:

- **Navigate to online server:** Starts Aurea CRM web in online mode.
- **Apply changes and stay offline:** Invalidates the cache and stays in offline mode.

If an error occurs during synchronization (e.g. `mmco.exe` not found), Offline Manager displays a message and the error is written to the `web.log` file.

If a synchronization attempt failed (e.g. due to loss of the network connection), the communication files are not deleted but remain in the `..\out` directory. The files are downloaded and processed at the next synchronization attempt.

Before synchronizing, CRM.launcher compares the client version number with the server version number:

- If the client has a different service pack number, the user is prompted to upgrade his Aurea CRM web offline version using CRM.launcher. For further details, see CRM.launcher on page 52.
- If the client has a different hotfix number, the synchronization process is launched.

Communication Logs

In Aurea CRM web you can view the communication logs of your offline users by selecting the **Com. Protocols** menu item (`D_ComProtocols`) from the `M_Reps` context menu. The menu action is displayed only for reps that have a station number assigned (ID:117). (Alternatively you can view all communication protocols via **Administration** > **Jobs & Logs** > **Communication Log**).

For information on exporting a communication log as an HTML report, see *Communication Log* in the Aurea CRM web Administrator Guide.

For a detailed description of the communication log's contents, see *Communication Logs* in the Aurea CRM win Administrator Guide.

Reinitializing an Offline Client

Learn how to reinitialize an offline client.

If an offline user experiences problems with his local data, you can repair the offline client by reinitializing it; this deletes all data from the client's local Aurea CRM database and inputs a new initial data stock download.

To reinitialize an offline client:

1. On the Aurea CRM web server, switch to the rep record of the corresponding offline user.
2. Set the **Initialize offline station** field (ID:116) to true.

Upon the next synchronization request from the offline client, the server outputs a new initial data stock download for this client.

A message is displayed, advising the user not to work with Aurea CRM web offline until the process is completed.

Deactivating an Offline Client

Learn how to deactivate an offline client.

You can deactivate offline clients that are no longer used. A deactivated offline client can no longer receive data from the Aurea CRM web server.

To deactivate an offline client:

1. On the Aurea CRM web server, switch to the rep record of the corresponding offline user.
2. Reset the **Offline user** field (ID:115) to false.

This disables the communication connection server -> offline station. No communication data is output for the offline station. The connection offline station -> server remains active, in order to be able to input data from the offline station.

If a rep is defined as **Inactive** (ID:69), the **Offline user** field is automatically disabled.

Deleting the offline user's rep record also deactivates the offline client.

5

Troubleshooting

Learn how to troubleshoot the CRM Launcher.

Do the following to troubleshoot CRM launcher:

Log Files

Troubleshooting using log files.

Use the `< LogPath >` tag in the `update.launcher.exe.config` to specify a central directory for log files created by CRM.launcher, Aurea CRM web offline and IIS Express (default: `web.data \log`). This path is used by CRM.launcher when creating a bug report and must therefore be absolute or relative to `update.Launcher.exe`, to ensure that all relevant log files can be found. for further details, see Creating a Bug Report on page 59.

Use the `<Logging>` section in the `update.launcher.exe.config` file to configure the logging for CRM.launcher, see the article “How to configure logging on the server” at https://support.aurea.com.

Note: Make sure the `filePath` attributes point to the above described `LogPath` directory, otherwise the log files is not found when creating a bug report.

The following log files are available:

- `OfflineSetup.log`: The log file written by the Aurea CRM web offline setup (location defined by `LogPath` in the `SilentClient.ini`).
- `launcher.log`
- `iisExpress.log`
- `web.log`

The remaining sections in this chapter provide details on configuring logs for web offline, CRM.launcher, and CRM.launcher plugins. For more information, see the wiki article “HOWTO Configure Logging for CRM.launcher, CRM.launcher Plug-Ins and update.CRM web offline” on the Aurea support site: https://support.aurea.com.

Configuring Log files location

Learn to configure log file for troubleshooting.

CRM.launcher, CRM.launcher plug-ins and CRM web offline should be configured to use the same "central" log folder. Therefore you need to configure log files path to use the same folder as described below:

- Set the central log files folder path for CRM web offline in the `settings.xml` at the XPath `update/ update.Lib /Logging` in the file.
- Set the central log files folder path for CRM.launcher and CRM.launcher plug-ins in the `app.config` file at the node `update.net/update.lib/Logging`.
- Set the folder path for upgrades logs (of CRM web offline and CRM.launcher) to the `<central log folder>/ PatchLogs` folder in the `app.config` file at the node `update.net/ update.launcher / LogPath`.

Setting up the log configurations

Learn to set up the log configuration for troubleshooting.

Logging is configured by adding element `<Logging/>` to the element `<update.lib>` in the configuration XML file (for example, `settings.xml` or `app.config`).

Logging consists of 3 components:

- Formatter — specifies how the lines in the log is formatted.
- Writer — specifies where the log messages is written (a file, the event log, a database, ...).
- Channel — combines a formatter with a writer.

Note: There can be any number of log channels, writers and formatters.

Log Writers – specifies where the log messages is written (a file, the event log, a database, ...)

Log writers are of two types:

1. **FileLogWriter— writes log messages into files in the file system**

   The XML attributes supported by FileLogWriter are listed in the table below:

XML Attribute	Description
Name	The name of the log writer. This name has to be specified as XML attribute `writerName` on the `<Channel />` element. This is a mandatory attribute.
type	The class name and assembly implementing the file log writer. Default value is `update.Lib.Logging.Writers.FileLogWriter,update.Lib`. It is a mandatory attribute.
filePath	The path to the log file. You can use environment variables. Relative paths relatives to the applications base directory. It is a mandatory attribute.
maxSizeKB	The maximum size of the log file in KB. What happens if this limit is exceed can be defined by XML attribute `rollingMode`.
closeOnIdleTimeout	The number of seconds after which the log file is closed when being idle. Use the `rollingMode` attribute to define when backup log files is created. • "None" - No backup log files is created • "Startup" - A new backup log file is created every time the process starts. • "Size" - A new backup log files is created when the size specified in maxSizeKB is exceeded. • "Date" - A new backup log file is created for every day. Multiple values can be combined using "," (for example, "Startup,Size")
encoding	Specify the encoding (default is the operating system's current ANSI codepage). Common values are "utf-8"and"utf-16".Seehttp://msdn.microsoft.com/de-de/library/system.text.encoding.aspx for a complete list.
compressBackup	If "True" the backup log files is compressed using gzip (default "False").
maxBackups	The maximum number of backup log files that is created. If this limit is exceeded the oldest log files is deleted.
backupFolder	The folder where backup log files is stored. You can use environment variables. Relative paths relates to the applications base directory.

XML Attribute	Description
autoFlush	If "True" (default) the log file is flushed after every write. This ensures that in case the logging process is terminated all the log messages are in the file.
includeDateInBackupFile-Name	If "True" a date is included in the backup file name. If "False" (default) the backup files is numbered consecutively.

A sample usage of the FileLogWriter element is shown below:

```
...
<Logging>
    ...
    <Writer name="fileLogWriter"
            type="update.Lib.Logging.Writers.FileLogWriter,update.Lib"
            filePath="myapplication.log"
/>
    ...
</Logging>
...
```

2. **SysLogWriter— sends log messages over a network using TCP or UDP packets.**

   Though it does not contain any SysLog-specific logic, it is designed to be used in combination with some flavor of the SysLogFormatter.

   The XML attributes supported by SysLogWriter are listed in the table below:

XML Attribute	Description
Name	The name of the log writer. This name has to be specified as XML attribute `writerName` on the `<Channel />` element.
type	The class name and assembly implementing the Sys log writer. Default value is `update.Lib.Logging.Writers.SysLogWriter,update.Lib`.
filePath	The path to the log file. You can use environment variables. Relative paths relates to the applications base directory.
host	The hostname or IP address of the target for the communication.
port	The port number of the target for the communication.
protocol	The communication protocol to be used (tcp or udp are supported).

A sample usage of the FileLogWriter element is shown below.

```
...
<Logging>
    ...
    <Writer name="mySysLogWriter"
            type="update.Lib.Logging.Writers.SysLogWriter,update.Lib"
            host="syslogserver.acme.com"
            port="514"
            protocol="udp"
/>
    ...
</Logging>
...
```

Log Formatters – specifies how the lines in the log is formatted.

You can specify the formats of the log messages using the log formatter elements. The following elements are available:

DefaultLogFormatter

The DefaultLogFormatter is a generic formatter. The following sample shows an example of its usage:

```
...
<Logging>
    ...
    <Formatter name="myFormatter"
            type="update.Lib.Logging.Formatters.DefaultLogFormatter,update.Lib"
/>
    ...
</Logging>
...
```

The XML attributes supported by DefaultLogFormatter element are listed in the table below:

XML Attribute	Description
Name	The name of the log writer. This name has to be specified as XML attribute `writerName` on the `<Channel />` element.
logDate	If "true" the timestamp is written to the log file.
logMillis	If "true" the timestamp including milliseconds is written to the log file. In this case the setting for logDate is ignored.
logTicks	If "true" the tick count is written to the log file.
logProcessId	If "true" the process ID of the current process is written to the log file.
logThreadId	If "true" the ID of the current thread is written to the log file.

XML Attribute	Description
logFacility	If "true" the facility is written to the log file.
logLevelFormat	Defines if and how the current log level is written to the log file. • "None" - The log level is not written to the log file. • "Value" - The numeric value of the log level is written to the log file. • "Name" - The name of the log level is written to the log file.
logException	If "true" exceptions is written to the log file.
logMemory	If "true" the memory consumed by the current process is written to the log file.
logCpu	If "true" the CPU utilization by the current process is written to the log file.
logUser	If "true" the name of the current user is written to the log file.

WebSysLogFormatter

The web SysLog formatter formats the messages in a web context so that they can be interpreted by a SysLog server. Apart from the standard set of XML attributes it supports the following:

XML Attribute	Description
type	The class name and assembly implementing the sys log formatter. This has attribute has to be set to `update.Web.Logging.WebSysLogFormatter,update.Web`.
maxMessageLength	The character limit for the message. Some SysLog servers (and actually the original SysLog standard as well) may limit the length of the message, in which case this setting can be used to make sure that our message won't be rejected. **Note:** If the message exceeds the specified length, it is truncated.

The following sample shows the usage of WebSysLogFormatter element:

```
...
<Logging>
```

```
    ...
    <Formatter name="myFormatter"
            type="update.Web.Logging.WebSysLogFormatter,update.Web"
/>
    ...
</Logging>
...
```

Channel

A channel element combines a formatter element and a writer element. A channel element defines how the messages are written (using the formatter elements) and where to write the log messages (using the writer elements).

The following code example, shows the usage of a Channel element:

```
...
<Logging>
     ...
     <Channel logLevel="Warning"
             writerName="myLogWriter"
             formatterName="myLogFormatter" />
     ...
</Logging>
...
```

The Channel element supports the following XML attributes:

XML Attribute	Description
logLevel	Defines how verbose this log channel is (default "Debug"). Possible values are: NoLogging; Emergency; Alert; Critical; Error; Warning; Notice; Info; and Debug.
writerName	The name of the writer used by this channel. A writer with this name has to be defined!
fallbackWriterName	The name of the fallback writer used by this channel. A writer with this name has to be defined! It is temporarily used in case of a failure of the default writer.
fallbackTimeout	Timeout period after which an attempt is made to reinitialize the writer after it has failed and has been replaced by the fallback writer (if any). The value has to be specified as a .NET-compatible TimeSpan, e.g. for 5 minutes the value would be 0:05, for 10 seconds 0:00:10. The default value is 5 minutes.
formatterName	The name of the formatted used by this channel. A formatted with this name has to be defined!
facilityFilter	Only log messages whose facility matches this filter is written to the channel. This filter has to be specified as a regular expression.

XML Attribute	Description
facilityFilterExclude	Log messages whose facility matches this filter is *NOT* written to the channel. This filter has to be specified as a regular expression.
messageFilter	Only log messages whose message text matches this filter is written to the channel. This filter has to be specified as a regular expression.
messageFilterExclude	Log messages whose message text matches this filter is *NOT* written to the channel. This filter has to be specified as a regular expression.

Note: The attributes writerName and formatterName are mandatory, all other attributes are optional.

Filtering log messages

Learn to filter the log message.

You can filter the log messages by specifying conditions, using one of the following attributes on the XML elements <Logging /> and <Channel />.

XML Attribute	Description
facilityFilter	Only log messages whose facility matches this filter is written to the channel.
facilityFilterExclude	Log messages whose facility matches this filter is *NOT* written to the channel.
messageFilter	Only log messages whose message text matches this filter is written to the channel.
messageFilterExclude	Log messages whose message text matches this filter is *NOT* written to the channel.

Note: Filters have to be specified as a regular expression. All filter attributes are optional.

Hints

Hints for troubleshooting.

Below are the troubleshooting hints:

Execution Timeout

If you experience problems concerning timeouts during the download of large delta communication files, you can increase this timeout in the `web.config` file of each client by editing/adding the line:

```
<httpRuntime executionTimeout = "350"/>
```

Installation Checkup

If an error occurs during synchronization, refer to the client's log files. For further information, see Creating a Bug Report on page 59.

Check the following items to identify problems caused by a faulty installation:

Aurea CRM win

- **Configuration module:** Start `mmcfg.exe` to verify that all logins have been correctly imported and that the correct station number is entered.
- **Directories:** Check whether all user directories as well as the `.. \< StationNumber >` directories for the central station and the user's station were created. After initialization, in `.. \<main office>\out` a notification file should exist that was created during the import of the initial data stock.
- **Files:** In the `.. \system\exe` directory only the following executables must be present: `mmcfg.exe`, `mmco.exe`, `mmim.exe`, `mmRegServers.exe`, `mmba.sup` and `mmri.exe`

  Furthermore, check the contents the `mmdb.ini` file in `.. \system\sys`.

Aurea CRM web

- **..\system\sys:** Aurea CRM web's `\sys` directory must be identical to Aurea CRM win's `\sys` directory.
- **web.config :** To see whether `web.config` file was correctly modified, verify that the tag `<clear/>` is not present, that there are only two `<Add Verb …>` tags.
- **CRM.designer configuration files:** After installing Aurea CRM web offline, the default configuration files (`Config_0.gz` and `Config_6.gz`) must be present in `< XMLStoragePath >`. After initialization, all child configurations for the offline user must be present as well. For further details, see CRM.designer Configurations for Offline Clients on page 19.
- **Offline.xml:** Check the `Offline.xml` file for all values provided during setup (station number, URL etc.).

Load Balancing on the Server

If you want to use load balancing on the server, make sure that the load balancer is able to work session-aware.

Explanation: For the synchronization process with the offline clients, `mmco.exe` is started on the server. Since the communication process can take a few minutes, it is started in background mode (in order not to block other users of the web server). The offline client polls the server's communication status at regular intervals. Since the communication process is only running on one node of the load balancing cluster, session-awareness is required to guarantee that an offline client always polls the same node; this status inquiry would lead to an error on all other cluster nodes and thus abort the synchronization process.

Troubleshooting CRM.launcher

For troubleshooting hints concerning CRM.launcher, see the article “Troubleshooting CRM.launcher 8.2 Feature Pack” at https://support.aurea.com.

Uploading Large Files

For security reasons the upload of files is restricted to 10 MB. When uploading larger files to the server the following properties need to be changed in the `web.config` file:

```
<system.web>
    <httpRuntime maxRequestLength="1048576" />
</system.web>
<system.webServer>
    <security>
        <requestFiltering>
            <requestLimits maxAllowedContentLength="1073741824" />
        </requestFiltering>
    </security>
</system.webServer>
```

Note: The values defined for `maxRequestLength` and `maxAllowedContentLength` must match. In the above example, the defined maximum upload size is 1024 MB. Therefore, `maxRequestLength` has to be set to 1048576 **KILOBYTES**, and `maxAllowed-ContentLength` to 1073741824 **BYTES**.

For details, see Microsoft Knowledge Base Article <httpRuntime>-Element.

6

Uninstalling

You can uninstall Aurea CRM web offline via **Add/Remove Programs**.

Note: Uninstalling Aurea CRM web offline requires administrator rights.

Note: Uninstalling Aurea CRM web offline deletes the database as well.

To uninstall Aurea CRM web offline in silent mode use one of the following methods:

- Command line: Start `Setup.exe` with parameters `--silent --uninstall`.
- To silently uninstall via **Add/Remove Programs**:

  Set `PrepareSilentDeinstallation` to "Y" (in the `SilentClient.ini` file).

This modifies the `UninstallString` entry in the registry to enable silent uninstallation.