

Kofax Customer Communications Manager

5.0

Installation Guide

CCM ComposerUI J2EE 5.0



Contents

Introduction.....	3
Copyrights and trademarks	3
The ITP/OnLine Server installation.....	3
Requirements.....	5
Client	5
Installing ITP/OnLine.....	6
Creating the Configuration Folder	6
Installing the Sample Applications	6
Installing CCM ComposerUI on the Web Server	7
WebSphere	7
JBoss	8
Apache Tomcat.....	10
SAP NetWeaver CE	10
Sun Application Server	11
Upgrading ITP/OnLine.....	13
Deploying the CCM ComposerUI Web Application	13
Updating an existing CCM ComposerUI Web Application on WebSphere.....	14
Upgrading the Sample Application	14
Upgrading CCM ComposerUI Server 3.x applications	15
Upgrading to CCM ComposerUI Server 4.2	15
Configuration	16
Configuring ITP/OnLine.....	16
General Configuration.....	16
Configuring Applications	16
ITP/OnLine configuration	16
ITP/Server environment.....	17
Text Blocks, Forms, Field Sets, Views, and Content Wizards.....	17
File Upload control.....	17
Client browser configuration	18
Internet Explorer security settings.....	18
OpenOffice.org support.....	18
To associate MIME-type application/vnd.sun.xml.writer with the sxw extension	19
Tips and tricks	19
Directory listing accessible	19
License	20

Introduction

CCM ComposerUI is an extension to CCM Core. CCM Core combined with CCM ComposerUI is called CCM ComposerUI Server.

All documentation can also be found on the Kofax Customer Communications Manager Knowledge Center (<http://ccmkc.kofax.com>).

Copyrights and trademarks

© 1993–2016 Lexmark. All rights reserved.

THIS SOFTWARE CONTAINS CONFIDENTIAL INFORMATION AND TRADE SECRETS OF KOFAX. USE, DISCLOSURE OR REPRODUCTION IS PROHIBITED WITHOUT THE PRIOR EXPRESS WRITTEN PERMISSION OF KOFAX.

Kofax, the Kofax logo, and the Kofax product names stated herein are trademarks or registered trademarks of Kofax in the U.S. and other countries. All other trademarks are the trademarks or registered trademarks of their respective owners.

U.S. Government Rights Commercial software. Government users are subject to the Kofax standard license agreement and applicable provisions of the FAR and its supplements.

You agree that you do not intend to and will not, directly or indirectly, export or transmit the Software or related documentation and technical data to any country to which such export or transmission is restricted by any applicable U.S. regulation or statute, without the prior written consent, if required, of the Bureau of Export Administration of the U.S. Department of Commerce, or such other governmental entity as may have jurisdiction over such export or transmission. You represent and warrant that you are not located in, under the control of, or a national or resident of any such country.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

The ITP/OnLine Server installation

CCM ComposerUI Server consists of two components:

1. CCM Core has functionality designed to make it function in an CCM ComposerUI Server setup. This functionality is an integral part of CCM Core and as such it is installed when CCM Core is installed. For more details refer to the CCM Core Installation Guide.
2. The web application CCM ComposerUI. The web application can be called from within a browser.

The web application CCM ComposerUI must be installed on a web server. This installation and the configuration of the web server are described in this installation guide.

An CCM ComposerUI installation can be used to run multiple 'custom applications', CCM ComposerUI web applications with customized looks and functionality. Several sample applications are delivered with CCM ComposerUI. This installation guide contains instructions on how to install and activate these sample applications.

Requirements

The following web servers are supported:

- WebSphere 6
- JBoss 5 or JBoss 6.1 in combination with Java 1.6
- Apache Tomcat 7.0 or Apache Tomcat 8.0 with Java 7 or higher
- SAP NetWeaver CE 7.1
- Sun Java™ System Application Server Platform Edition

Note

There is a known limitation for Sun Java™ System Application Server Platform Edition. On this platform CCM ComposerUI is incapable of deriving the browsing user from the HTTP context. Please refer to Sun Application Server (see "Sun Application Server" page 11) for more details.

Note

JBoss 4.0 will be supported until the end of 2011, **for existing installations only**. We strongly advise to upgrade to JBoss 5 in combination with Java 1.6.

Client

Browsers

CCM ComposerUI is supported on:

- Microsoft Internet Explorer version 11,
- Microsoft Edge
- Mozilla Firefox version 38 and up,
- Google Chrome version 43 and up,

Please refer to the Known Issues section of the CCM ComposerUI Server Manual for any issues related to specific browsers or browser versions.

Other software

CCM ComposerUI produces preview documents in PDF format. To view these documents properly a PDF plug-in is required for the client browser. Adobe Reader is one of the PDF viewers which includes such a plug-in.

CCM ComposerUI offers functionality to edit Rich Text Blocks. To be able to edit Rich Text Blocks, you need to have Microsoft Word 2010 or higher and a browser with ActiveX support enabled. Currently, this feature is only available on CCM ComposerUI (ASP.NET). Editing of Rich Text Blocks in the Microsoft Word DOC format is only supported with Microsoft Word 2010.

Virus scanners can negatively influence performance.

Installing ITP/OnLine

This chapter provides information on the procedure for creating a new installation of the CCM ComposerUI web application. The procedure for upgrading existing installations is described in chapter Upgrading ITP/OnLine (see "Upgrading ITP/OnLine" page 13).

Creating the Configuration Folder

The configuration and custom applications of an CCM ComposerUI installation are stored in a single configuration folder. To prepare for installation of CCM ComposerUI, start by creating a configuration folder, for instance in C:\ITPOnLine.

Note:

For technical reasons, the path to the configuration folder should not contain spaces.

Installing the Sample Applications

The functionality of the CCM ComposerUI web application is exposed through one or more custom CCM ComposerUI applications, which are installed in the CCM ComposerUI configuration folder.

CCM ComposerUI comes with a sample custom application: newsample. This sample application is delivered in the installation package in the form of a ZIP archive called newsample.zip.

Note

The ZIP archives custom1.zip and custom2.zip do not contain complete samples. These zip files are used in examples in the ITP/OnLine Customization Manual.

Note that installation of the sample application is optional. Usually, the sample application should be installed for development installations. However, for production installations it is advisable to install only the custom CCM ComposerUI application that will be used in this particular production situation.

In order to make these sample applications available, follow the steps below:

1. **Create** the folder newsample in the CCM ComposerUI configuration folder.
2. **Unzip** the file newsample.zip into the folder newsample.
3. **For SAP only:** Change the redirection URL in the modelbegin.jsp file. Replace "runmodel?" with "../.. /app/*ApplicationName*/runmodel?", where *ApplicationName* is the name of the application like newsample. Also change the redirection URL in the modelselect.jsp file: replace "listmodels?" with "../.. /app/*ApplicationName*/listmodels?".

This procedure is the same for all supported web servers.

Note

If the sample application is installed after the CCM ComposerUI web application has already been deployed and the web server has been started, the application is not immediately available. To make it available, the web server must be restarted.

Installing CCM ComposerUI on the Web Server

The installation of CCM ComposerUI on the web server consists of several steps. The procedures for these steps are different for each of the supported web servers. They are described for each of the supported web servers in the sections that follow.

In general, the steps are as follows:

1. First, configure the location of the CCM ComposerUI configuration folder in the web server configuration. This ensures that CCM ComposerUI knows where to find its configuration files.
2. For some web servers, the web server configuration must be altered so that CCM ComposerUI can work properly.
3. Finally, the CCM ComposerUI web application must be deployed to the web server.

The procedures for these steps are different for each of the supported web servers. They are described for each of the supported web servers in the following sections.

WebSphere

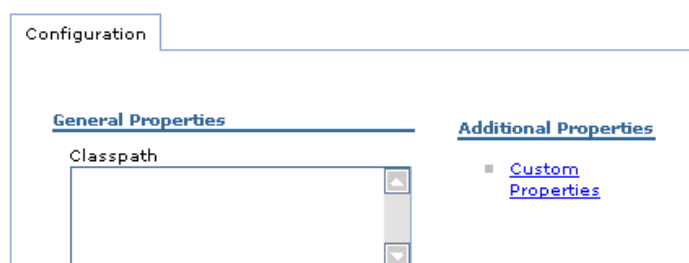
Setting the ITP/OnLine Configuration Folder

To set the configuration folder, you need to set the variable `com.aia_itp.itpols.config` to the full path to the configuration folder that was created in section [Creating the Configuration Folder](#) (see "[Creating the Configuration Folder](#)" page 6).

To do this, start the WebSphere web admin and go to *Servers > Application Servers* and click on the server name. Then click on **Process Definition** in the Java and Process Management section:



Then choose *Java Virtual Machine > Custom Properties* (under additional properties):



Click **New** and create a property with the variable `com.aia_itp.itpols.config`. As the value, enter the full path to the configuration folder:

General Properties

* Name
com.aia_itp.itpols.config

* Value

Description

Apply OK Reset Cancel

Which results in:

Select	Name	Value
<input type="checkbox"/>	com.aia_itp.itpols.config	C:\ITPOLS

Total 1

Hereafter, you may have to save your master configuration and you will have to restart the server for the changes to take effect.

Deploying the CCM ComposerUI Web Application

Open the web administrative console and select *Applications > Install New Application*. As the path to the new application, select the file `/itp.ear` that is included in the CCM ComposerUI installation package. Enter "itp" as the context root.

Specify the EAR, WAR or JAR module to upload and install.

Path to the new application.

Local file system

Specify path

Remote file system

Specify path

Context root
 Used only for standalone Web mc

Then complete the install wizard, which contains a number of follow-up screens. The default settings are normally all right. To view the new application and start/stop it, go to *Applications > Enterprise Applications*.

JBoss

Setting the ITP/OnLine Configuration Folder

To set the configuration folder, you need to set the variable `com.aia_itp.itpols.config` to the full path to the configuration folder that was created in section *Creating the Configuration Folder* (see "Creating the Configuration Folder" page 6).

To configure this in JBoss on Microsoft Windows, locate the following lines in the file `run.bat`,

which is part of the JBoss installation:

```

echo =====
echo.
echo   JBoss Bootstrap Environment
echo.
echo   JBOSS_HOME: %JBOSS_HOME%
echo.
echo   JAVA: %JAVA%
echo.
echo   JAVA_OPTS: %JAVA_OPTS%
echo.
echo   CLASSPATH: %JBOSS_CLASSPATH%
echo.
echo =====
echo.

```

Immediately *above* these lines, add the following lines:

```

rem Setup the ITP Server config location
set JAVA_OPTS=%JAVA_OPTS% -Dcom.aia_itp.itpols.config=<path>

```

Then, replace <path> by the full path to the CCM ComposerUI configuration folder. After adding these lines, the JBoss server must be restarted.

Note:

The run.bat code snippet above is from the JBoss 5.0.1 run.bat file. The contents of run.bat may differ slightly between JBoss 5 versions. It is important that the com.aia_itp.itpols.config variable is added to the JAVA_OPTS before the JAVA_OPTS are actually used.

To configure this in JBoss on Linux, it is similar to the example for Microsoft Windows above. Add the following lines at *the end of the run.conf* configuration file, which is part of the JBoss installation.

```

JAVA_OPTS="$JAVA_OPTS -Dcom.aia_itp.itpols.config=<path>"

```

Then, replace <path> by the full path to the CCM ComposerUI configuration folder. After adding these lines, the JBoss server must be restarted.

Note:

These instructions are only valid when JBoss is started using run.bat or run.sh. It is also possible that JBoss is configured to run as a service. In this case, add the option -Dcom.aia_itp.itpols.config=<path> to the parameters that are passed to the Java process that runs JBoss. The exact details of how this should be done depends on the method and operating system that is being used to run JBoss as a service.

Deploying the CCM ComposerUI Web Application

To deploy the CCM ComposerUI web application, copy the file itp.ear, which is included in the CCM ComposerUI installation package, into the 'deploy' folder of the JBoss server configuration that will be used to run CCM ComposerUI. For the default server configuration, this folder is located at [jboss install dir]\server\default\deploy.

Apache Tomcat

Deploying the CCM ComposerUI Web Application

Deploying CCM ComposerUI in Apache Tomcat 7.0 or 8.0 requires manual extraction of the packaged web applications, since .ear files are not supported. To deploy the application, perform the following steps:

1. Create a new folder in the Apache Tomcat webapps directory (usually C:\apache-tomcat-<version>\webapps), with the desired name of the CCM ComposerUI instance.
2. Extract the itp.ear file to a temporary location.
3. Extract the itp.war file from this temporary location into the newly created webapps sub-directory.

It may be required, after applying the Configuration Folder setting (See Setting the CCM ComposerUI Configuration Folder (see "Setting the CCM ComposerUI Configuration Folder" page 10)), to restart the Tomcat web server.

Setting the CCM ComposerUI Configuration Folder

To set the configuration folder, you need to add a context parameter to the deployed web application. To do this, navigate to the Apache Tomcat webapps folder, and open the newly deployed CCM ComposerUI instance folder. Then, open the WEB-INF folder, and edit the web.xml file.

The <web-app> XML tag needs to have an additional <context-param> tag created. This should be of the following format:

```
<context-param>
  <param-name>ITP-OnLineApps-Dir</param-name>
  <param-value>C:\ITPOnLine</param-value>
</context-param>
```

Here, 'C:\ITPOnLine' is an example, which should be replaced with the path to the configuration folder that was created in the section Creating the Configuration Folder (see "Creating the Configuration Folder" page 6).

This step may require you to restart the Tomcat web server for the changes to be applied.

SAP NetWeaver CE

Deploying the CCM ComposerUI Web Application

Copy the itp.sca file to the SAP NetWeaver CE deployment folder (usually C:\usr\sap\trans\EPS\in).

Start the Java Support Package Manager (JSPM) on the NetWeaver deployment system, and log on.

1. Select New Software Components as the Package Type, and continue with Next.
2. Select the AiaSoftware/itp software component for deployment, and continue with Next.
3. Check that the AiaSoftware/itp software component has status OK, and continue with Next.
4. Deployment will start, which takes some time and includes a **restart of the SAP system**.
5. Check that the AiaSoftware/itp software component has status DEPLOYED, and finish with Exit.

Setting the CCM ComposerUI Configuration Folder

To set the configuration folder, you need to set the variable `onlineappsdir` to the full path to the configuration folder that was created in section [Creating the Configuration Folder](#) (see "Creating the Configuration Folder" page 6).

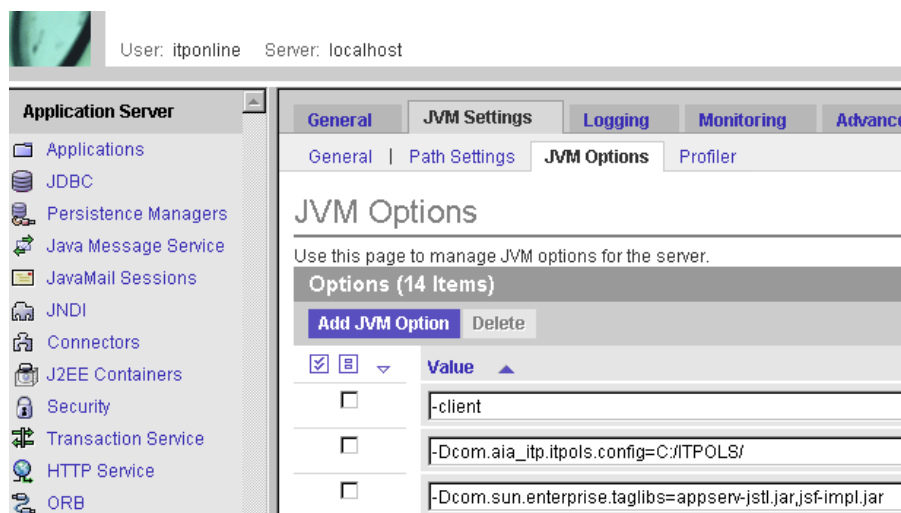
Start the AS Java Config tool and connect to the database.

1. Navigate to: `cluster-data > template - CE_Composition_Environment_production_full > applications > AiaSoftware > itp.`
2. Select the key: `onlineappsdir`.
3. Enter the location of the previously created Configuration Folder in the Custom value field.
4. Press the Set button and use Apply Changes from the File menu to apply the new setting.
5. **Restart the SAP system** as prompted by the AS Java Config tool, which can be done using the SAP Management Console.

Sun Application Server

Setting the CCM ComposerUI Configuration Folder

To set the configuration folder, start the server. Then, start the web admin console and log in. In the application server selection, choose *JVM Settings - JVM Options*. As shown below, add the option `"-Dcom.aia_itp.itpols.config=<path>"`, where `<path>` is the full path to the CCM ComposerUI configuration folder.



In order to circumvent a problem in Sun Application Server, it is important to add the option `"-Dcom.aia_itp.itpols.noprincipal=Y"`. This prevents CCM ComposerUI from trying to derive the browsing user from the HTTP context, which will fail on this platform.

As a consequence of this setting, CCM ComposerUI will pass an empty user name to CCM Core. For security reasons, this will only be accepted when using prepared model runs (see CCM ComposerUI Manual for more information).

In order to allow for classic model runs, a user name can be configured to be passed to CCM Core. This can be done on a per application basis, by setting the `"-Dcom.aia_itp.itpols.<application>.runas=<user>"`, where `<application>` is the name of the CCM ComposerUI application (e.g. *sample*) and `<user>` is the user name (e.g. *domain/user*). In this case it is important that, after completing the installation, the "Use NTLM authentication" checkbox on the main configuration page of CCM ComposerUI is checked and that the "Verify NTLM

authentication" checkbox is left unchecked.

Setting the Web Server Security Configuration

The Sun J2EE Application server has a more strict default security configuration than other servers. This prevents CCM ComposerUI from saving configuration changes to the CCM ComposerUI configuration folder. For this reason, the default security configuration of the Sun J2EE Application Server must be altered.

Configure these settings by opening the file `<AppServer>\domains\domain1\config\server.policy`. In this file, locate the following section:

```
// Basic set of required permissions granted to all remaining code
grant {
  permission java.lang.RuntimePermission "loadLibrary.*";
  permission java.lang.RuntimePermission "queuePrintJob";
  permission java.net.SocketPermission "*", "connect";
  permission java.io.FilePermission "<<ALL FILES>>", "read,write";

  // work-around for pointbase bug 4864405
  permission java.io.FilePermission
    "${com.sun.aas.instanceRoot}${/}lib${/}databases${/}-", "delete";
  permission java.io.FilePermission "${java.io.tmpdir}${/}-", "delete";

  permission java.util.PropertyPermission "*", "read";

  permission java.lang.RuntimePermission "modifyThreadGroup";
};
```

Then change the following line:

```
permission java.io.FilePermission "<<ALL FILES>>", "read,write";
```

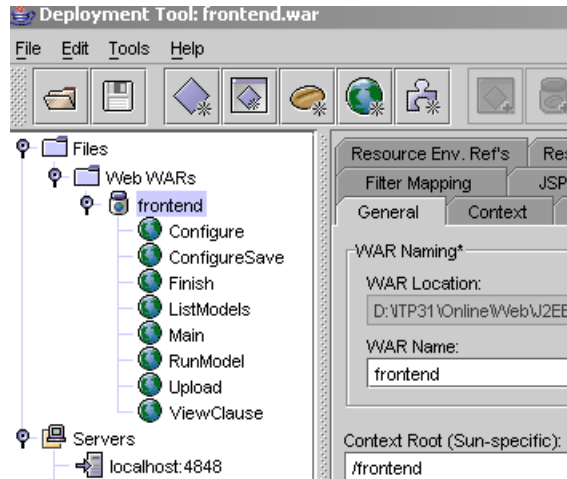
into:

```
permission java.io.FilePermission "<<ALL FILES>>", "read,write,delete";
```

Deploying the CCM ComposerUI Web Application

To deploy the CCM ComposerUI web application, start the deploy tool. Select *File > Open*, and open the file `itp.ear`, which is included in the CCM ComposerUI installation package. Select the file in the tree view.

Now select *Tools > Deploy*.



Upgrading ITP/OnLine

Upgrading an existing CCM ComposerUI installation consists of two steps. The first step consists of deploying the CCM ComposerUI web application, the second step consists of optionally upgrading the sample applications. These upgrade steps are described in the following sections.

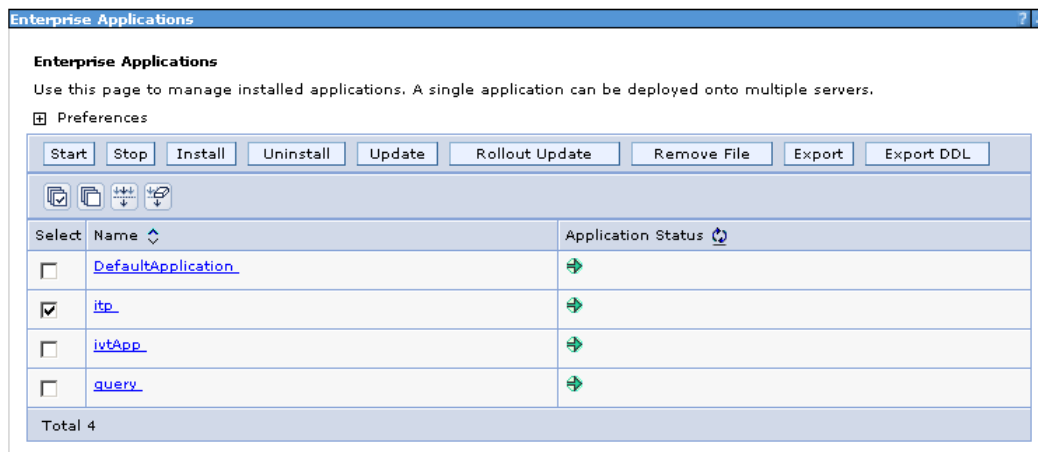
Deploying the CCM ComposerUI Web Application

To upgrade an CCM ComposerUI installation, the CCM ComposerUI web application should be re-deployed to the web server. The procedure for deploying the CCM ComposerUI web application is different for each of the supported web servers:

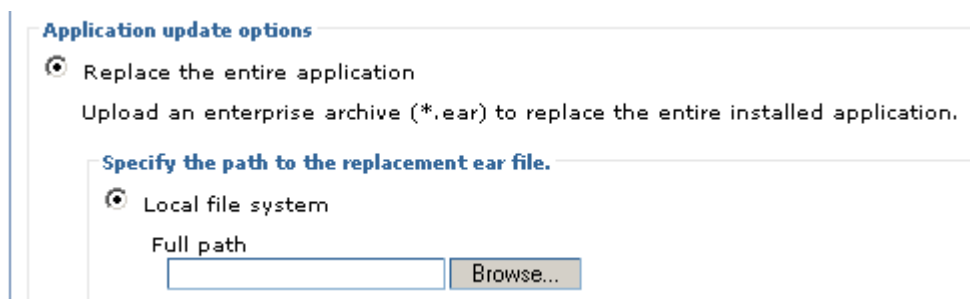
- For JBoss, the procedure is the same as for a new installation. The instructions can be found in JBoss: Deploying the ITP/OnLine Web Application (see "Deploying the CCM ComposerUI Web Application" page 9). After deploying the upgrade, JBoss does not need to be restarted.
- For Sun Application Server, the procedure is also the same as for a new installation. Refer to Sun Application Server: Deploying the ITP/OnLine Web Application (see "Deploying the CCM ComposerUI Web Application" page 12).
- For WebSphere, refer to the instructions in section Updating an existing ITP/OnLine Web Application on WebSphere (see "Updating an existing CCM ComposerUI Web Application on WebSphere" page 14).

Updating an existing CCM ComposerUI Web Application on WebSphere

Open the web administrative console and select *Applications > Enterprise Applications*. Select the application 'itp' and click the button Update.



In the dialog that follows choose to replace the entire application. Similar as to deploying, select the file `itp.ear` that is included in the CCM ComposerUI installation package. Again, enter "itp" as the context root.



Then complete the wizard, which contains a number of follow-up screens. The default settings on these follow-up screens are normally all right.

Upgrading the Sample Application

When upgrading the CCM ComposerUI web application, it is advisable to install the new version of the CCM ComposerUI sample custom application as well.

Note

Be careful to **not** install the new version of the CCM ComposerUI sample application in the following situations:

- When the installed sample application has been modified, and when these modifications should be retained. Installing a new version of the sample application will destroy any customizations that have been made.

- When the current installation does not have any sample application installed. Installing the sample application is an optional step in the installation of CCM ComposerUI. The sample application may have been skipped for a reason, for instance because the installation is intended for a production situation. Before installing the sample application, make sure that there is a reason for doing so.

To upgrade the CCM ComposerUI sample application, follow the steps below

1. In the CCM ComposerUI configuration folder, rename the folder newsample to newsample.old.
2. Create a new folder called newsample.
3. Install the new version of the application by unzipping the file newsample.zip, which is included in the CCM ComposerUI distribution, to the folder newsample.
4. Move the file config.xml from the folder newsample.old into the folder newsample.
5. Delete the folder newsample.old.

Note

From CCM ComposerUI 4.4 and onwards, the sample applications sample and sample2 are no longer distributed.

Upgrading CCM ComposerUI Server 3.x applications

From CCM ComposerUI 4.4 and onwards, support for table oriented output has been dropped. When upgrading to CCM ComposerUI Server 4.4, existing table oriented applications should be upgraded to div oriented applications.

Upgrading to CCM ComposerUI Server 4.2

Between CCM ComposerUI Server 3.5 and CCM ComposerUI Server 4.2 there is one relevant breaking change. Letterbooks are no longer retrieved from the file system; they are configured in and retrieved from ITP Base instead.

The new Letterbooks are expressed in XMLs with the namespace

`xmlns:itp='http://www.aia-ity.com/4.2/listmodels'` (rather than

`xmlns:itp='http://www.aia-ity.com/3.1/interact'`). This impacts all CCM ComposerUI

applications that contain customizations to the Letterbook functionality. The following files in your CCM ComposerUI applications are suspect to contain references to the old namespace and have to be modified to refer to the new namespace:

- `\xslt\custom\defaults_letterbook.xsl`
- `\xslt\custom\letterbook.xsl`

Note:

All other files in `\xslt\custom` folder do not apply to Letterbooks and should therefore not be subject to this change.

Configuration

This chapter describes the configuration of the CCM ComposerUI web interface, and various other configurations that may need to be modified for usage with CCM ComposerUI.

Configuring ITP/OnLine

General Configuration

Browse to `http://<server>:<port>/itp/configure`, where <server> is the name of your web server, and <port> is the port on which the web server is configured to listen.

In the general configuration page, you can specify the host name and port for the CCM Core that CCM ComposerUI will use.

When you click the option submit, these settings will be stored in the file `config.xml` in the CCM ComposerUI configuration folder.

Refer to the CCM ComposerUI Server Manual for an extended description on the general configuration settings.

Configuring Applications

A single CCM ComposerUI J2EE installation can expose multiple custom CCM ComposerUI applications on the same web site. Applications can be added, configured and removed through the general configuration page. Refer to the CCM ComposerUI Server Manual for an extended description of the application configuration settings.

ITP/OnLine configuration

It is possible to run models on CCM ComposerUI Server 3.1.14 or later from within the CCM Repository. Refer to the section Test Model tab for more information on configuring the CCM Repository project. The CCM ComposerUI Server itself also needs some configuration for this to work, which is described here.

1. Only perform this step when using an CCM Core version prior to 3.2.4. Version 3.2.4 and later already contain these scripts.
Copy the `ITPOLSStartModel.dss` script from the Resources folder in the CCM Repository Server installation to the `scripts/system` subfolder in the work folder of the CCM Core installation. Also copy the `CheckRepositoryModelAccess.dss` script from the Resources folder to the `scripts/user Library` folder of CCM Core.
2. In the CCM Core Administrator, select the services node and the Constants tab. Set the `AllowRepositoryModelRun` constant to the value `Y`, add it if it is not already present.
3. Recompile the scripts from within CCM Core Administrator, and click Save & Apply.
4. (Only perform this step when using CCM Core versions prior to 3.2.4, or when using an CCM Repository version 3.5.9 or earlier)
Copy the `GetModel.exe` from an CCM Repository Client installation to the `bin` subfolder of the CCM Core installation.

The AllowRepositoryModelRun constant determines if CCM ComposerUI Server will accept model run requests from the CCM Repository. If not set to 'Y', CCM Core will deny running models from within the CCM Repository. By explicitly having to configure an CCM ComposerUI Server installation to accept CCM Repository runs, it can be avoided that CCM Repository runs are run accidentally or unauthorized on a particular CCM ComposerUI Server installation.

The user exit script CheckRepositoryModelAccess.dss is run to check whether a specific user is allowed to run a specific model on CCM ComposerUI Server. The default implementation allows all. If you want more restrictive checking, you can implement it by changing this script.

ITP/Server environment

If a model is run using CCM ComposerUI Server, the CCM ComposerUI application decides which CCM Core environment should be used. If no environment is passed by the CCM ComposerUI application, the CCM Core default environment will be used. It is possible to override these configurations by passing an CCM Core environment on the CCM ComposerUI Server URI. In this way it is possible to configure an alternate environment for a project.

In the following example the environment 'live' is used to run the ITP Models.

```
http://itponlineserver:[port]/itp/app/sample/modelbegin.jsp?env=live
```

Text Blocks, Forms, Field Sets, Views, and Content Wizards

Dynamic objects -- a Data Backbone, Text Blocks, Forms, Field Sets, Views, and Content Wizards -- are retrieved from the CCM Content Publication Database or Editorial CCM Repository at the moment the ITP Model first references the object during execution. Which CCM Content Publication Database installation is used to retrieve these objects from is configured by a setting in the CCM Core environment. The dynamic objects are always retrieved from the same Project as the Model.

If a Model is tested from the CCM Repository Editorial the dynamic objects will be retrieved from the same Editorial CCM Repository from which the test run was initiated.

File Upload control

In the default configuration, the CCM ComposerUI web interface will use an ActiveX control to support file uploads from CCM ComposerUI forms. This means that by default the **File Upload mechanism will only work with Microsoft Internet Explorer/Windows clients**. Also, the security settings for the client's browser must allow the download and run of (signed) ActiveX controls. Refer to Microsoft Internet Explorer security settings (see "Internet Explorer security settings" page 18) for details.

The CCM ComposerUI web interface also supports the usage of a standard HTML file selection control. This can be controlled using a javascript variable. This variable can be found in the file `support\javascript\common.js`:

```
fileEditActiveX = true;
```

Change it to:

```
fileEditActiveX = false;
```

To use the HTML file selection control. Note that in order to modify `common.js`, it must be copied from the CCM ComposerUI EAR archive into the corresponding location in the CCM ComposerUI custom application.

Note that the use of an ActiveX control means that at first use this control is downloaded to the client and installed. When problems are encountered with this control, check the client's browser security settings, they should be as described in the next chapter. If the settings are correct it might be that the download/install of the ActiveX control failed. There is a method to check this; refer to the chapter Known Issues of the CCM ComposerUI Server Manual, section File upload fails, nothing happens.

Client browser configuration

There are some specific settings needed on client browsers. This paragraph lists them.

Internet Explorer security settings

By default ITP/OnLine Server uses an ActiveX control to load result documents in Word and to upload the edited documents to the server. Also by default ITP/OnLine Server also uses an ActiveX control to upload files to the server. (Refer to the OnLine manual the Customisation chapter for the settings that control the use of these controls.) Because of this, the security settings on Internet Explorer clients must allow the download a subsequent use of ActiveX controls.

Depending on whether the ITP/OnLine Server is part of the same intranet the client is part of, the security settings must be set for Internet or for the Local intranet. Choose in the client's browser: *Tools > Internet Options... > Security tab*. If the ITP/OnLine Server is part of the network the client, select the web content zone: Local Intranet, otherwise select Internet. Then, click the button Custom level and the window Security Settings will appear. Scroll to the section that is concerned with ActiveX controls and plug-ins. Recommended settings are:

- Download signed ActiveX controls: Enable.
(Prompt is also an option but this will show an extra window to the users that will allow them to deny the download. Use prompt on a test client and Enable on real clients)
- Download unsigned ActiveX controls: Disable.
- Initialize and script ActiveX controls not marked as safe: Disable.
- Run ActiveX controls and plug-ins: Enable.
- Script ActiveX controls marked safe for scripting: Enable.

These settings will prevent that the user is prompted while at the same time maintaining a degree of security by excluding unsigned ActiveX.

If ITP/OnLine Server is part of the same intranet the client is part of, we advise to check the option Automatic logon with current user name and password in the section User Authentication the window Security Settings.

OpenOffice.org support

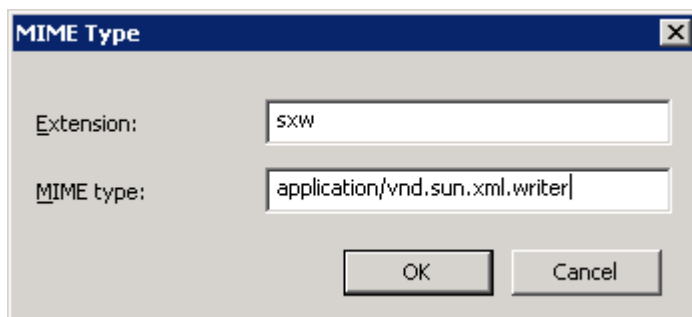
Support for OpenOffice.org documents has been added to ITP/(OnLine) Server 3.1.8. OpenOffice.org documents need the following configuration on the Web server:

- MIME-type application/vnd.sun.xml.writer needs to be associated with the sxw extension

For new installations, the ITP/OnLine installer will take care of this. If you update from an older version of ITP/OnLine and want to use the OpenOffice.org support, you must configure these settings manually:

To associate MIME-type application/vnd.sun.xml.writer with the sxw extension

- Go to the web site in which you created the virtual directories for your installation
- Right-click the virtual folder for the Front-end and select properties
- Go to the HTTP headers tab and click MIME Types. Click on the New button
- Associate to extension sxw with MIME type application/vnd.sun.xml.writer



Tips and tricks

Directory listing accessible

If you install the web server according the standard configuration, a directory listing is accessible in the browser for some of the CCM ComposerUI application folders. It is possible to change this behavior in the web server configuration.

For JBoss, you have to look for the file web.xml and change the parameter value to 'false'. You can find this web.xml file for example in:

`\jboss-4.0.0\server\default\deploy\jbossweb-tomcat50.sar\web.xml`.

The following setting determines that directory listings are not shown.

```
<servlet>
  <servlet-name>default</servlet-name>
  <servlet-class>
    org.apache.catalina.servlets.DefaultServlet
  </servlet-class>
  <init-param>
    <param-name>debug</param-name>
    <param-value>0</param-value>
  </init-param>
  <init-param>
    <param-name>listings</param-name>
    <param-value>>false</param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
</servlet>
```

License

The ITP/OnLine license needs to be added with the ITP/Server license manager on the ITP/OnLine Server machine. Refer to the ITP/Server installation guide for more details.