CQ5 WCM System Administrator Guide

Day

CQ5 WCM System Administrator Guide

Contents

1 Introduction

1.1 Introduction

CQ5 is the next generation product from Day, moving forward with functionality while continuing our commitment to standards.

The product has been completely redesigned, and rewritten, introducing new concepts and technologies. This has a major impact on system administration.

1.2 Purpose of this Document

To provide the information necessary for all system administration tasks, including information on CQ WCM technical requirements, and installation, upgrade, and configuration instructions for CQ WCM.

1.3 Target Audience

- System Administrators

2 CQ WCM Technical Requirements

2.1 Introduction

This chapter covers the technical issues of:

- Supported Platforms
- Minimum Sizing Requirements
- Prerequisites
- Support Levels

2.2 Prerequisites

To install CQ WCM you need:

- JRE or JDK 1.5 (or higher). See Additional Platform Notes for further information.

Note

The JRE is no longer bundled with CQ WCM as it was in versions prior to CQ 5. You can obtain JRE 1.5 or higher from the Sun web site

- the CQ WCM quickstart jar file which also contains the required applications:
  - CRX 1.4.2
  - CQSE (CQ Servlet Engine) supporting 2.4 (or higher)

2.3 Minimum Sizing Requirements

The following are the *minimum* requirements:

- 400 MB free disk space in the deployment directory
- 512 MB memory

Note

If you are using CQ DAM, you must increase the heap size when starting CQ WCM.

You will have to increase these in accordance with your operational expectations.

2.4 Support Levels

This document lists the supported client and server platforms for CQ 5.2 WCM. Day provides several levels of support, both for our recommended configurations and for other configurations.

2.4.1 Recommended Configurations

Day recommends these configurations and provides full support as part of the standard software maintenance agreement. Here we share the same support level overview as in CRX.

The configuration is tested and known bugs are fixed. Issues that are submitted by customers in the form of a fully qualified request are addressed in future releases.

Table 2.1. Recommended Configurations

Support Level	Description
A: Supported	Day provides full support and maintenance for this configuration. This configuration is covered by Day's quality assurance process.

Note

In previous releases, Day used three different support levels for the fully supported configurations (A-C). Currently the supported platforms are all on level A.

2.4.2 Configurations with Optional Support Agreement

Day cannot guarantee that the configuration works as expected, and support for these configurations is not included in the standard software maintenance agreement.

Table 2.2. Configuration with Optional Support Agreement

Support Level	Description
D: Validated, optional support available	This configuration has been validated by Day or a third party, or it is or has been in productive use. This platform is only sporadically tested by Day or third parties. Day offers optional platform support service agreements for this configuration.
E: Expected to work	The configuration is expected to work, and there are no reports to the contrary. Day offers optional platform support service agreements for this configuration.

2.4.3 Unsupported Configurations

Table 2.3. Unsupported Configurations

Support Level	Description
Z: Not supported	The configuration is not supported. Day does not make any statements about whether the configuration works, and does not support it.

2.5 Supported Platforms

2.5.1 Server Operating Systems

CQ5 works with the following server platforms:

Table 2.4. Supported Server Operating Systems

Platform	Support Level
Windows Server 2008	**A**: Supported
Windows 2003	**A**: Supported
Windows 2000	**D**: Optional support available
Windows XP	**A**: Supported
Solaris 10	**A**: Supported
Solaris 9	**A**: Supported
Solaris 8	**D**: Optional support available
RedHat Linux Kernel 2.6	**A**: Supported
RedHat Linux Kernel 2.4	**D**: Optional support available

Platform	Support Level
HP-UX 11i v1-3	**A**: Supported
IBM AIX 6.1	**D**: Optional support available
IBM AIX 5.3	**A**: Supported
IBM AIX 5.2	**D**: Optional support available
SUSE Linux Enterprise Server 10	**D**: Optional support available
SUSE Linux Enterprise Server 9	**D**: Optional support available
Mac OS X	**D**: Optional support available
SGI IRIX 6.5	**D**: Optional support available

Important

If you are installing CQ 5.x on a SUSE Linux Enterprise Server, be sure to have the latest Java SDK version installed. Also, if you are installing all components, including the examples, you must either increase the max heap size, or install using a two-step process:

1. Install (only) the CQ Servlet Engine using custom installation.
2. Install the Authoring instance with customized installation.

Warning

Windows 2003 with SP2 is not able to connect to WebDAV. Please see Microsoft - Software Update for Web Folders (KB907306) for full details of the manual updates which need to be made.

2.5.2 Cloud Computing Environments

CQ5 works with the following virtual machines in cloud computing environments:

Table 2.5. Supported Cloud Computing Environments

Platform	Support Level
Amazon EC2 (running non-clustered CQ5)	**D**: Optional support available

Important

EC2 optional support also needs compliance to Day's support policy for all other system components.

2.5.3 Servlet Engines / Application Servers

CQ5 runs on the following application servers.

Note

You can run CQ WCM without an application server, but you need a Servlet Engine.

Both CRX, and therefore CQ WCM, ship with Day's CQSE servlet engine, which you can use freely and which is fully supported.

Table 2.6. Supported Servlet Engines / Application Servers

Platform	Support Level
CQSE 4.1.5	**A**: Supported

Platform	Support Level
BEA WebLogic Server 10.3	**A**: Supported
BEA WebLogic Server 9	**D**: Optional support available
BEA WebLogic Server 8.1	**D**: Optional support available
IBM Websphere 6	**A**: Supported
IBM Websphere 5	**D**: Optional support available
Apache Tomcat 6	**D**: Optional support available
Apache Tomcat 5	**D**: Optional support available
Apache Geronimo 2	**D**: Optional support available
JBoss Application Server 4	**D**: Optional support available
Sun ONE Application Server 8	**D**: Optional support available
Oracle Application Server 9i	**D**: Optional support available
Jetty 4-6	**D**: Optional support available
Sun GlassFish	**D**: Optional support available

2.5.4 Java Virtual Machine

CQ5 operates with the following versions of the Java Virtual Machine (Runtime Environment).

Table 2.7. Supported Java Virtual Machines

Platform	Support Level
Sun JRE 1.6.x	**A**: Supported
Sun JRE 1.5.x	**A**: Supported
Sun JRE 1.4.2	**Z**: Not supported
IBM JRE 1.5	**A**: Supported
HP JRE 1.5	**A**: Supported

Important

The IBM and HP environments are only supported on IBM (AIX) and HP (HP-UX) server platforms respectively. For all non-IBM, non-HP platforms, the Sun JRE is supported.

Warning

If you are using Sun JDK 1.5.0_6, your Java VM may crash. Upgrading to Sun JDK 1.5.0_7 solves this problem.

2.5.5 Dispatcher Platforms (Web Servers)

The following platforms are supported for use with the Dispatcher, Day's caching and load balancing solution:

Table 2.8. Dispatcher - Dispatcher Platforms (Web Servers)

Platform	Support Level
Apache httpd 2.2	**A**: Supported
Apache httpd 2.0	**A**: Supported
Apache httpd 1.3.x	**D**: Optional support available
MS IIS 7 (Microsoft Internet Information Server)	**A**: Supported

Platform	Support Level
MS IIS 6 (Microsoft Internet Information Server)	**A**: Supported
MS IIS 5 (Microsoft Internet Information Server)	**A**: Supported
Sun Java System Web Server 7	**D**: Optional support available
Sun Java System Web Server 6	**D**: Optional support available

Note

Other HTTP servers, built on the basis of Apache sources, only have the same support level as that of the Apache httpd server when official Apache source distributions were used, or they were delivered with the operating system.

If in doubt, please ask Day for confirmation of the support level related to the respective server product.

2.5.6 Client Platforms

CQ WCM works with the following client platforms.

Important

All browsers are tested without plug-ins and add-ons. Some plug-ins and add-ons may interfere with CQ.

2.5.6.1 Microsoft Windows XP

Table 2.9. Supported Client Platforms : Microsoft Windows XP

Platform	Browser	Support Level
Windows XP	Microsoft Internet Explorer 7.0	**A**: Supported
WIndows XP	Microsoft Internet Explorer 6.0	**A**: Supported
WIndows XP	Mozilla Firefox 3.x	**A**: Supported
WIndows XP	Mozilla Firefox 2.x	**D**: Optional support available
WIndows XP	Mozilla Firefox 1.5	**D**: Optional support available

2.5.6.2 Microsoft Windows Vista

Table 2.10. Supported Client Platforms : Microsoft Windows Vista

Platform	Browser	Support Level
Windows Vista	Microsoft Internet Explorer 7.0	**A**: Supported
Windows Vista	Mozilla Firefox 3.x	**A**: Supported
Windows Vista	Mozilla Firefox 2.x	**D**: Optional support available

2.5.6.3 Microsoft Windows 2000

Table 2.11. Supported Client Platforms : Microsoft Windows 2000

Platform	Browser	Support Level
Windows 2000	Microsoft Internet Explorer 7.0	**D**: Optional support available
Windows 2000	Microsoft Internet Explorer 6.0	**D**: Optional support available
Windows 2000	Microsoft Internet Explorer 5.5	**D**: Optional support available

Platform	Browser	Support Level
Windows 2000	Mozilla Firefox 3.x	**D**: Optional support available
Windows 2000	Mozilla Firefox 2.x	**D**: Optional support available

2.5.6.4 Apple Macintosh OS X

Table 2.12. Supported Client Platforms : Apple Macintosh OS X

Platform	Browser	Support Level
Mac OS X	Mozilla Firefox 3.x	**A**: Supported
Mac OS X	Mozilla Firefox 2.x	**D**: Optional support available
Mac OS X	Microsoft Internet Explorer 5.2	**Z**: Not supported

Note

CQ5 is tested with the latest Mozilla Firefox version available. All browsers are tested without plug-ins and add-ons. Some plug-ins and add-ons may interfere with CQ5.

2.5.7 Java Content Repository

CQ5 operates with any JSR-170 compliant repository.

Table 2.13. Supported Java Content Repositories

Platform	Support Level
Day CRX 1.4	**A**: Supported
Apache Jackrabbit 1.4	**D**: Optional support available

2.5.8 Additional Platform Notes

This section provides special notes and more detailed information about running CQ WCM on certain platforms.

2.5.8.1 Server Platforms

Note

WebDAV Clients. Windows 2003 with SP2 cannot connect to WebDAV. See Microsoft - Software Update for Web Folders (KB907306) for full details of the manual updates that need to be made.

2.5.8.2 Java Runtime Environments

Note

Java Versions. Generally, at least JRE 1.5 is recommended, as the following CQ WCM components or features require at least JRE 1.5:

- CRX Launchpad
- CRX Package Manager
- CQ WCM Quickstart

If JRE 1.4.2 is required due to other project/deployment dependencies, the CRX core repository (`crx-explorer_crx.war`) can still run with this Java version. In this case, the Package Manager is not supported.

Note

Java Distributions. The CQ WCM application requires only a Java Virtual Machine to run, which is provided by the Java Runtime Environment (JRE) distribution. Generally Day recommends using the Java Development Kit (JDK) distribution, which is a typical requirement of the Java application servers in which CQ WCM runs. Both JRE and JDK distributions are a part of the Java Standard Edition (Java SE).

Note

Java Environment Suppliers. The IBM and HP Java environments are only supported on IBM and HP server platforms, respectively. For all non-IBM, non-HP platforms, the Sun JRE is supported.

2.5.8.3 Application Servers

Note

Servlet API Versions. CRX Launchpad web development environment, based on Apache Sling, requires a servlet engine supporting Servlet API version 2.4. All other components of CRX, including the JCR repository and CRX web application, also work with servlet engines supporting Servlet API version 2.3.

Note

Application Server Support. CQ WCM is expected to work with any modern application servers/servlet engines (that support at least Servlet API 2.3), including popular open source products like Jetty, Apache Tomcat, Apache Geronimo or Sun GlassFish. Day recommends stable, non-deprecated versions of these application servers (information available on supplier websites). For production use, Day offers optional support for these platforms.

Note

QA Note. CQ WCM on IBM WebSphere 6 application server was tested with the IBM Java Development Kit (JDK) 1.4.2 under Linux.

3 Installing CQ WCM

3.1 Introduction to Quickstart and Installing CQ WCM

Unlike many other programs, you install CQ WCM by using a "Quickstart" self-extracting jar file. When you double-click the jar file for the first time, everything you need is automatically installed. The CQ quickstart jar file includes all files required for:

- the CRX 1.4.1 repository (a JCR-170 compliant repository and Apache Sling), virtual repository services, index and search services, workflow services, security and a Web server.
- the CQ Servlet Engine (CQSE). You can run CQ WCM without an application server, but you need a Servlet Engine. Both CRX, and therefore CQ WCM, ship with Day's CQSE servlet engine, which you can use freely and which is fully supported.

The first time you start the jar file, it creates an entire JCR-compliant repository in the background, which may take several minutes. After this startup is much quicker as the applications have been installed and the repository already created.

This section contains detailed instructions on how to install CQ WCM using the Quickstart jar file.

Note

These instructions apply only to CQ version 5 and higher. Versions prior to 5 used an installer rather than a quickstart jar file. For information about installing previous versions, see the appropriate documentation at docs.day.com.

This section includes information on the following:

- Installation requirements
- Installing, starting, and stopping CQ WCM
- Custom installation procedures
- Installing CQ WCM on an application server
- Troubleshooting
- Uninstalling CQ WCM

3.2 Installation Requirements

3.2.1 Software and Hardware Requirements

Before you start to install CQ WCM please read CQ WCM Technical Requirements, in particular, the following sections:

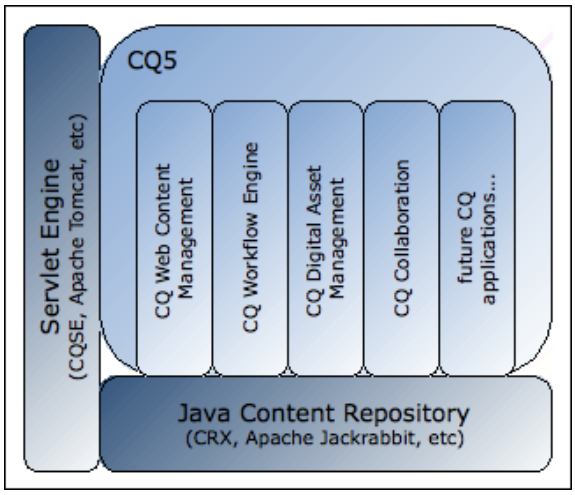
- Prerequisites
- Minimum Sizing Requirements
- Supported Platforms (together with the section called "Support Levels")

3.2.2 CQ 5 Architecture

CQ system architecture is described in detail in the CQ5 Architect Guide, which includes the underlying technologies such as Sling.

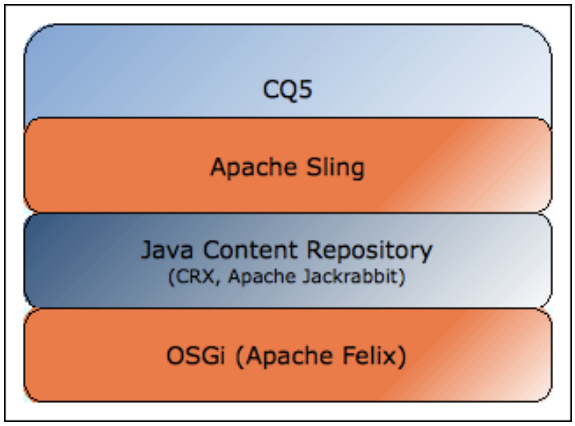
The following figure shows a high-level overview of how the various Day products and modules interact:

Figure 3.1. CQ5 Overview



The following figure shows the underlying technologies:

Figure 3.2. CQ5 Technology Stack



For detailed CQ5 architecture information, see the CQ5 Architect Guide.

3.2.3 Persistence Managers and Databases

Content is stored using a Persistence Manager. The persistence manager saves the repository content to a permanent storage solution, such as the file system or a database. By default, CQ5 (CRX 1.4) saves repository content to the Tar persistence manager.

If you use a database persistence manager to store the CQ data in an external database (that is, a database on a server that runs independently from CQ), you need to set up the database before installing CQ. By default, CQ 5 WCM uses the tar persistence manager. If you need to configure a database persistence manager, see Custom Installation - Configuring a Persistence Manager other than TarPM.

3.3 Installing, Starting, and Stopping CQ5 WCM

3.3.1 How to Install CQ WCM Author and Publish Instances using Quickstart

This section describes how to install CQ WCM.

Generally, when you set up CQ WCM, you need to set up an Author and a Publish instance - see the section called “Author and Publish Environments” for further details on the two types of environment.

Installation procedures for these are described in Installing an Author instance and Installing a Publish instance. If, for testing or other purposes, you need to install CQ WCM out of the box (an author instance with default settings and file names), you can use the generic procedure.

3.3.1.1 Installing an Author Instance

This procedure describes how to set up a default Author instance on port 4502 of the desired host.

To install an author instance:

1. On the host file system, create a directory and name it **`author`**.
2. Copy the CQ5 `cq-wcm-quickstart-<version>.jar` file into **`author/`**.
3. Rename `cq-wcm-quickstart-<version>.jar` to `cq5-author-4502.jar`. If you need a different port this can be set in the filename.
4. Copy a valid `license.properties` file into **`author/`** as well (the same directory as the `cq5-author-4502.jar` file).

   Note

   If when starting the application, you do not provide the `license.properties` file, CQ WCM redirects you to the **`Welcome`** screen where you enter a valid license key. You can also request a valid license key from Day at this time.

5. Start CQ WCM Quickstart:

   - If using a GUI file-system explorer, **`double-click`** the `cq5-author-4502.jar` file.
   - If using the command line, type:

     ```
     java -jar cq5-author-4502.jar
     ```

   - Use a custom script located in the `crx quickstart`folder, such as `server.bat` to start CQ. The Start and Stop scripts are for UNIX, Linux, and Macintosh. The `server.bat` script is for Windows.

     Important

     You cannot use a custom script when you install the `quickstart.jar` file unless you expand the file first. Use the `-unpack` option on the command line to unpack the contents before running the script as in `java -jar cq-wcm-quickstart-<version>.jar -unpack`.

6. When the installation is completed, you are automatically redirected to `http://localhost:4502/bin/login.html`.

   Note

   With the default settings, the syndication agent points toward the publish instance at `http://localhost:4503/`.

3.3.1.2 Installing a Publish Instance

To set up a publish instance on port 4503 of the desired host, you perform the same steps as in installing an author instance except that you create a directory named `publish` (instead of

author) and you rename the quickstart.jar file as cq5-publish-4503.jar. You can select any port number.

To install a publish instance:

1. On the host file system, create a directory and name it **publish**.
2. Copy the CQ WCM cq-wcm-quickstart-<version>.jar file into **publish/**.
3. Rename cq-wcm-quickstart-<version>.jar to cq5-publish-4503.jar. If you need a different port this can be set in the filename.
4. Copy a valid license.properties file into **publish/** as well (the same directory as the cq5-publish-4503.jar file).

   Note

   If when starting the application, you do not provide the license.properties file, CQ WCM redirects you to the **Welcome** screen where you enter a valid license key. You can also request a valid license key from Day at this time.

5. Start CQ WCM Quickstart:
   - If using a GUI file-system explorer, **double-click** the cq5-publish-4503.jar file.
   - If using the command line, type:

     ```
     java -jar cq5-publish-4503.jar
     ```

   - Use a custom script located in the crx quickstartfolder, such as server.bat to start CQ. The Start and Stop scripts are for UNIX, Linux, and Macintosh. The server.bat script is for Windows.

   Important

   You cannot use a custom script when you install the quickstart.jar file unless you expand the file first. Use the -unpack option on the command line to unpack the contents before running the script as in java -jar cq-wcm-quickstart-<version>.jar -unpack.

6. When the installation is completed, you can browse your site (for example, http://localhost:4503/content/geometrixx/en/company.html)

3.3.1.3 Installing a CQ WCM Instance - Generic Procedure

This procedure is a generic, quickstart procedure for installing CQ WCM; it installs an author instance using default settings and file names. It is often used when testing, or other scenarios when you need a quickly available instance.

Note

You do not need to perform this procedure before installing author or publish instances. To install an author or publish instance, see Installing an Author instance and Installing the Publish instance.

To install a generic CQ WCM instance (author):

1. Copy the CQ WCM cq-wcm-quickstart-<version>.jar file (for example, cq-wcm-quickstart-5.2.0.20090319.jar) to the desired directory on the host file system.
2. Copy a valid license.properties file into the same directory as the cq-wcm-quickstart-<version>.jar file.

Note

If when starting the application, you do not provide the `license.properties` file, CQ WCM redirects you to the **`Welcome`** screen where you enter a valid license key. You can also request a valid license key from Day at this time.

3. Start CQ WCM Quickstart by doing one of the following:

   - If using a GUI file-system explorer, **`double-click`** the `cq-wcm-quickstart-<version>.jar` file. This installs and automatically starts the server.

Note

The repository data is stored in the subdirectory `crx-quickstart/repository`.

   - If using the command line, type the following:

```
java -jar cq-wcm-quickstart-<version>.jar
```

   - Use a custom script located in the `crx quickstart` folder, such as `server.bat` to start CQ. The Start and Stop scripts are for UNIX, Linux, and Macintosh. The `server.bat` script is for Windows.

Note

You cannot use a custom script when you install the `quickstart.jar` file unless you expand the file first. Use the `-unpack` option on the command line to unpack the contents before running the script as in `java -jar cq-wcm-quickstart-<version>.jar -unpack`.

4. When the installation is complete, you are automatically redirected to `http://localhost:4502/bin/login.html`.

Note

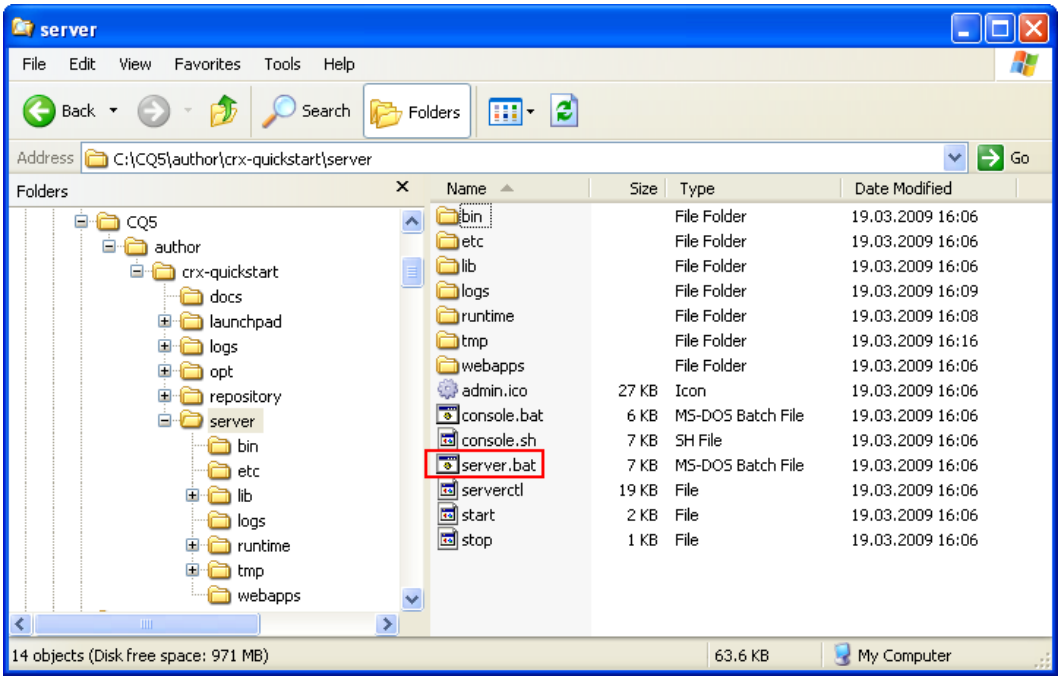
CQ WCM quickstart selects the first available port from the following list: `4502,8080,8081,8082,8083,8084,8085,8888,9362,<random>`. You can also set the port number. See installing an author instance for an example on how to set a port number.

3.3.2 Starting CQ WCM

Running CQ WCM is the same as starting it: you double-click CQ Quickstart or start CQ from the command line or a custom script.

To start CQ WCM:

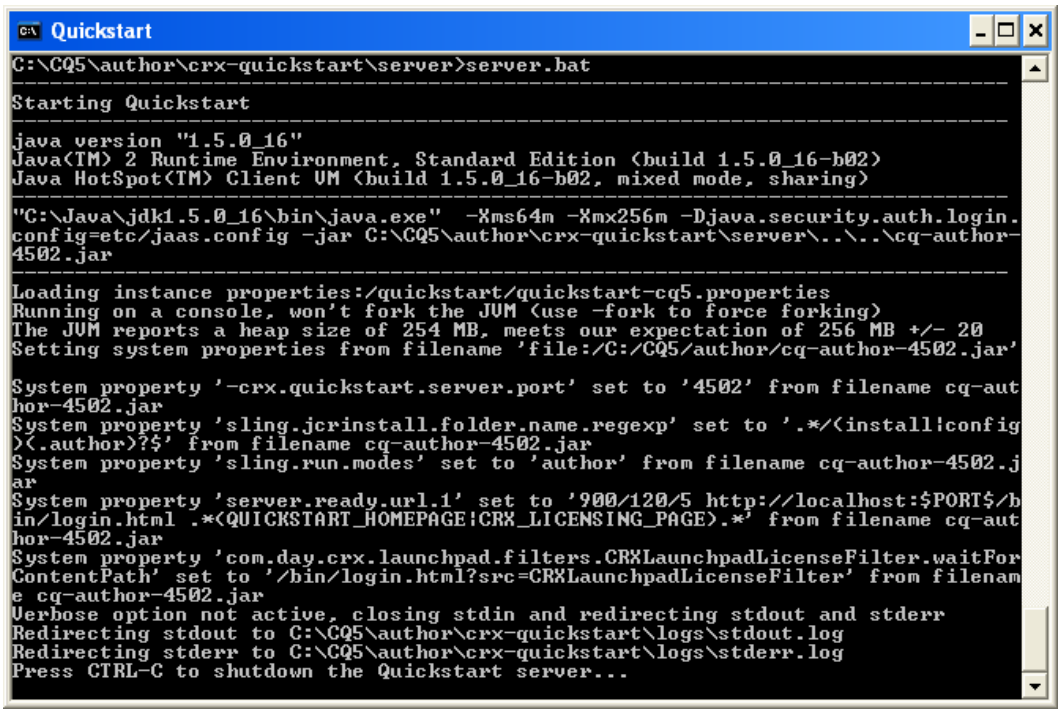
1. Navigate to your `quickstart.jar` file.
2. Do one of the following:
   a. If using a GUI file-system explorer, double-click the quickstart.jar file.
   b. If using the command line, type `java -jar cq-wcm-quickstart-<version>.jar`.
   c. Use one of the custom scripts located in the `crx-quickstart/server` folder to start CQ. The `start` and `stop` scripts are for UNIX, Linux, and Macintosh. The `server.bat` script is for Windows:



Note

You cannot use a custom script when you install the `quickstart.jar` file unless you expand the file first. Use the `-unpack` option on the command line to unpack the contents before running the script.

Running `server.bat` looks like the following:



3. CQ WCM starts and automatically redirects your web browser to the appropriate page, usually the login page (for example `http://localhost:4502/`):

Once logged in, you have access to CQ WCM. See the CQ WCM User Guide for details on using CQ WCM.

3.3.2.1 Options for the Windows server.bat script

The `server.bat` file can be used to start CQ under Windows. The file defines various default values and can be customized as required.

Caution

Day recommends that you make a backup of the file before editing.

The options for `server.bat` are:

```
Quickstart
usage: server.bat [options]

where options include:
    -debug [socket]           enable debug (shared mem is default)
    -suspended                start suspended (only if debug)
    -quiet                    don't show info message
    -jconsole                 start with -Dcom.sun.management.jmxremote
    -help                     this usage
    -jar                      explicitely specify the jar file to run

for additional tuning, edit the first section of the server.bat file
```

3.3.2.2 Options for the shell start and serverctl scripts

Both the `start` and `serverctl` scripts can be used to start CQ. Both files set various default values and can be customized as required.

Caution

Day recommends that you make a backup of the files before editing them.

To use the `start` script enter the following at the command prompt:

- **`start [options]`**

To use the `serverctl` script enter the following at the command prompt:

- **`serverctl [options] {start|stop|restart|status}`**

The options for both are:

```
Day Quickstart Server Control, version 5.0.0 ($Rev: 45191 $)
    usage: serverctl [options] {start|stop|restart|status}

Engine Options:
    --context          the communique context directory, defaults to <cq-installation-dir>/
crx-quickstart/server
    --port, -p         the default port to listen to if not overrriden in server.xml
    --loglevel, -l     the initial loglevel.
    --logdir           the log directory for startup.log. defaults to <cq-installation-
dir>/crx-quickstart/server/logs
    --log              the startup log file. defaults to <cq-installation-dir>/crx-
quickstart/server/logs/startup.log
    --interface, -a    the interfact to bind to (use 0.0.0.0 for any).
    --jar              explicitely specify the jar file to run, relative to --context

Java Options:
    --javahome         the java home directory. overrides JAVA_HOME env var.
    --heap-min         the minimum heap in megabytes. defaults to 128
    --heap-max         the maximum heap in megabytes. defaults to 256
    --debug, -d [socket|shmem]
                       starts jvm in debug mode. default 'socket'
    --debug-suspended [socket|shmem]
                       starts jvm in suspended debug mode. default 'socket'
    --debug-port       port for debug address. default "30303"
    --profile [yjp]    start jvm with YourKit Java Profiler
    --javaopts         additional java options
    --jaas             use jaas.config. default is disabled
    --jaas-config      config for jaas. default is etc/jaas.config
    --verbose-gc       turn on vebose gc

Other options
    --fg, -D           starts cq in foreground
    --bg               starts cq in background. this is the default.
    --verbose, -v      be more verbose
    --dry              prepare but do not start
    --help, -h         show this help
    --max-files        sets the ulimit for max open files before executing the jvm.
                       default is 1024
```

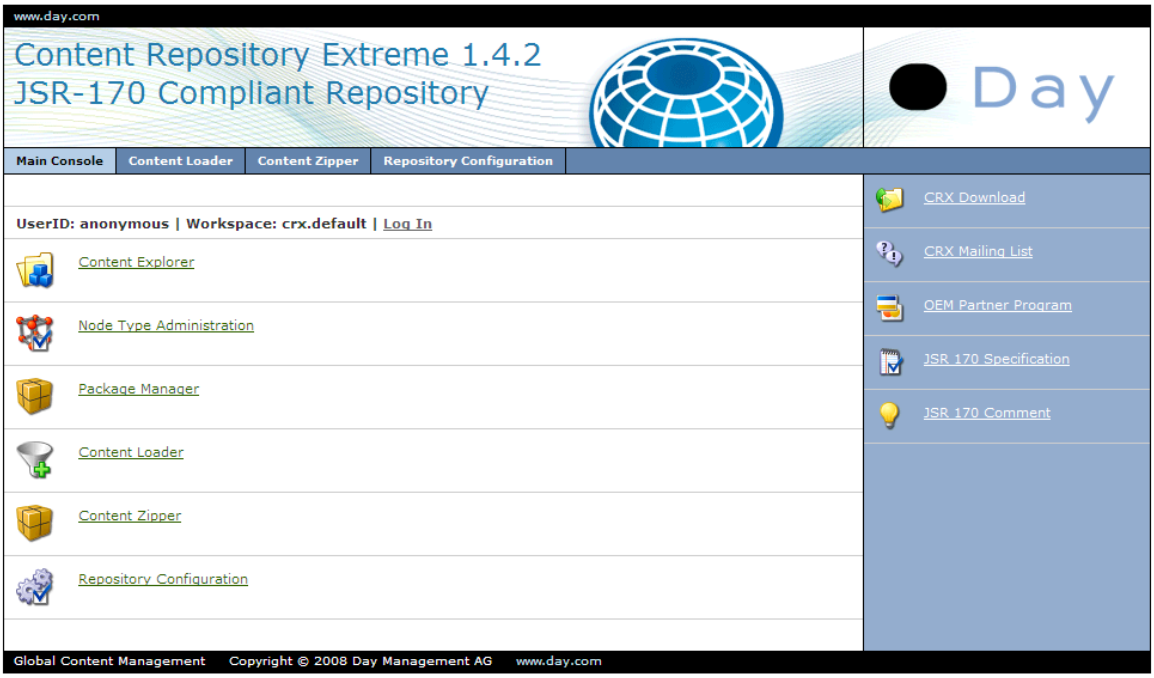
3.3.3 Accessing CRX, CQSE, and the Apache Felix Management Console

Once CQ WCM has been started, you can also access:

- CRX - used to access and manage the repository
- CQSE - used to manage the web applications
- Apache Felix Management Console - used to manage or configure the OSGi bundles

3.3.3.1 Accessing CRX

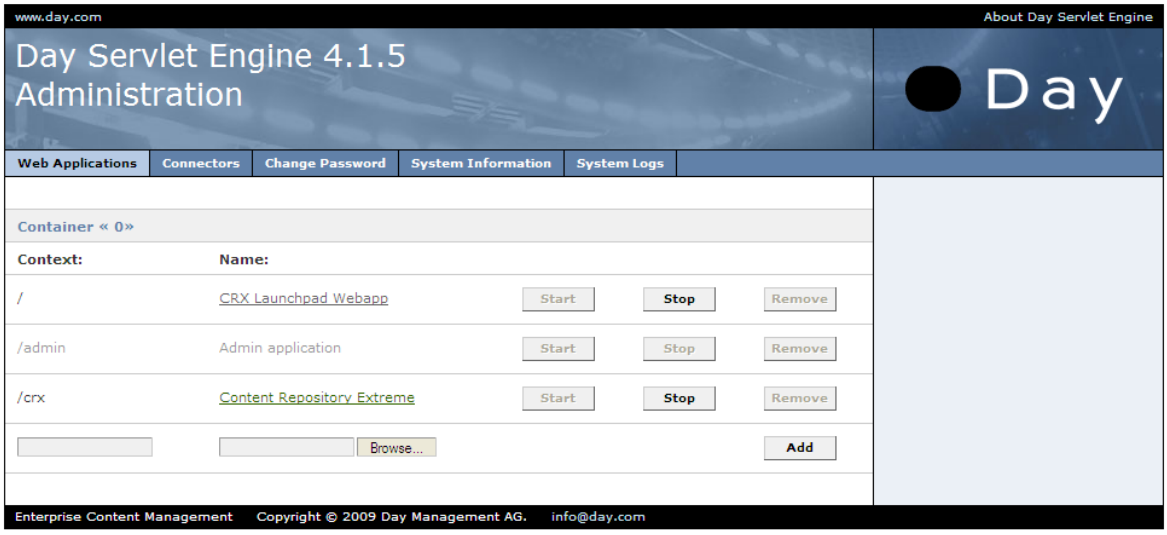
To access the CRX console use your browser to navigate to `http://<host>:<port>/crx/index.jsp`; for example `http://localhost:4502/crx/index.jsp`:



See the CRX documentation for full information about using CRX.

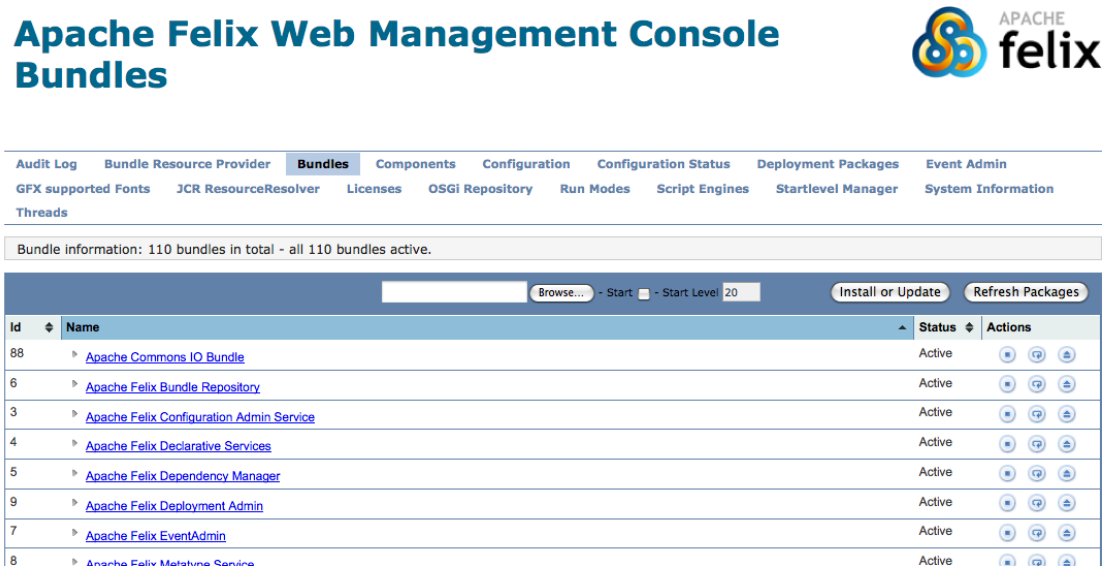
3.3.3.2 Accessing CQSE

To access the CQSE console use your browser to navigate to `http://<host>:<port>/admin/webapps`; for example `http://localhost:4502/admin/webapps`:



3.3.3.3 Accessing the Apache Felix Web Management Console

To access the Felix Web Management console use your browser to navigate to `http://<host>:<port>/system/console`; for example `http://localhost:4502/system/console` or for the Bundles page `http://localhost:4502/system/console/bundles`:

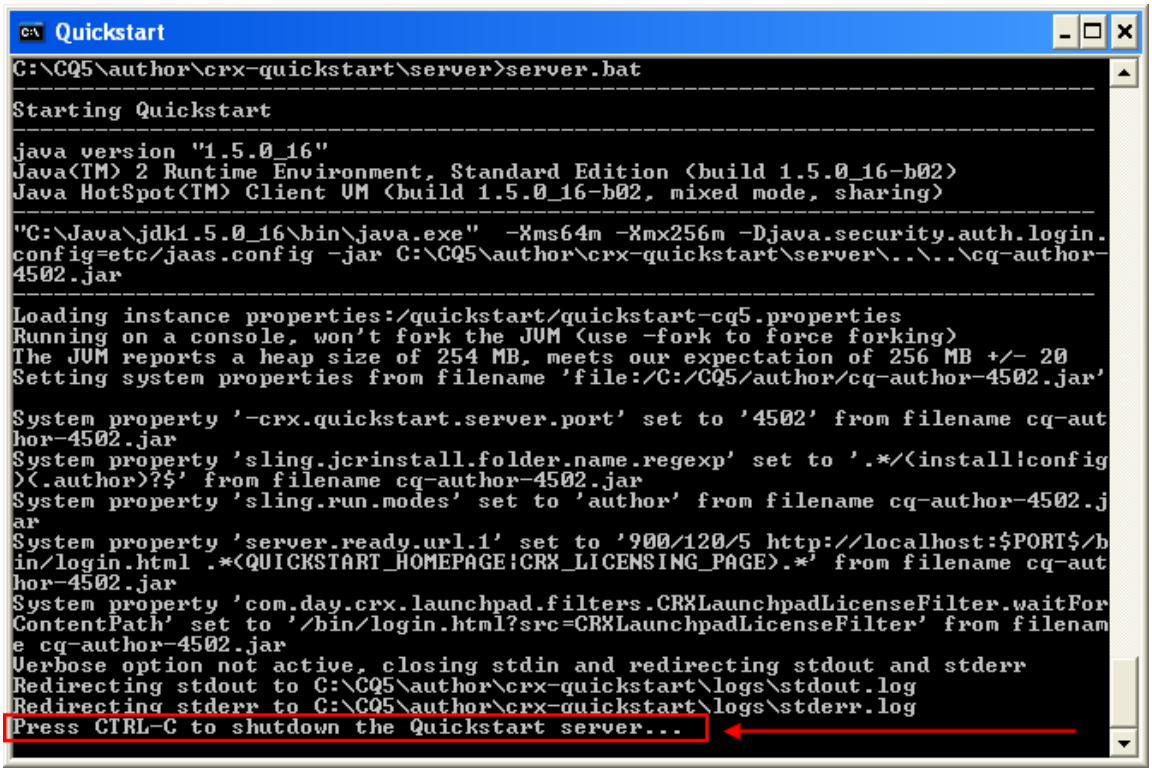


See the section called “Configuring CQ WCM with the Apache Felix Web Management Console” for further details.

3.3.4 Stopping CQ5 WCM

To stop CQ WCM, do one of the following:

- If you started CQ WCM from either a script or the command line, press **`Ctrl+C`** to shut down the server.



Note

If you are running UNIX with the start script, you must use the Stop script to stop CRX.

- If you started CQ WCM by double-clicking the jar file, click the **`On`** button (which then changes to **`Off`**) to shut down the server.

3.4 Custom Installation

For any customization you want to make to the CQ WCM installation, several options are available via the command line. You can either use the available options or rename the Quickstart file for custom uses.

3.4.1 Changing the Port Number by Renaming the File

The default port for CQ WCM is 4502. If that port is not available or already in use, Quickstart automatically configures itself to the first available port number as follows: 4502, 8080, 8081, 8082, 8083, 8084, 8085, 8888, 9362, <*random*>. You can also set the port number.

To change the port number of a CQ WCM quickstart file, you can rename the file by including a port number, for example, `cq-quickstart-4503.jar` or `cq-quickstart-7503.jar`. See Installing an Author instance and Installing a Publish instance for setting up the author and publish instances of CQ WCM. Renaming syntax is covered in detail in the quickstart help file (`-help` option). You can also change the port number by adding the `-port` option.

Note

When you rename the file, it must start with `cq` or `crx` as in `cq5-publish-4503.jar`.

3.4.2 Configuring a Persistence Manager other than TarPM

To configure CQ WCM to use a persistence manager other than the default Tar PM, you must modify the CRX war file before starting installation. See Configuring a Persistence Manager other than TarPM in the CRX documentation for detailed instructions.

3.4.3 Installing and Starting CQ5 WCM as a Windows Service

To install and start CQ5 WCM as a Windows service:

1. In the command line, navigate to the `crx-quickstart/opt/helpers` folder of CQ5, and type `instsrv.bat cq5`. This installs the service on the system.

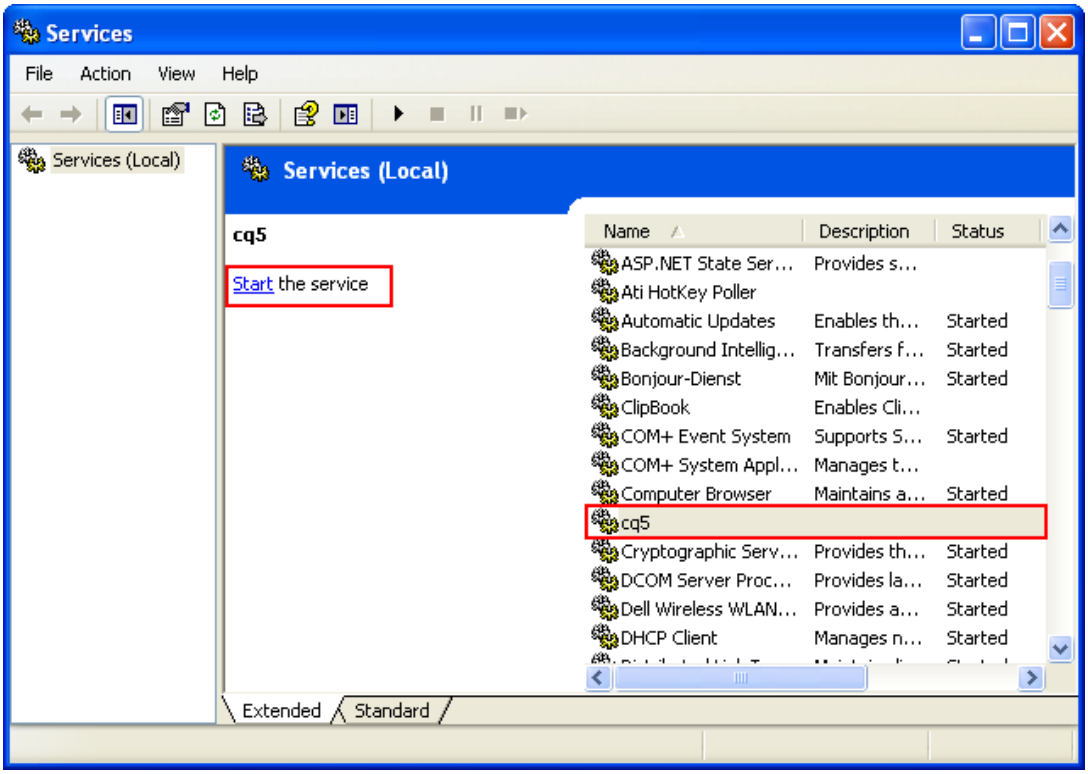
   

   Note

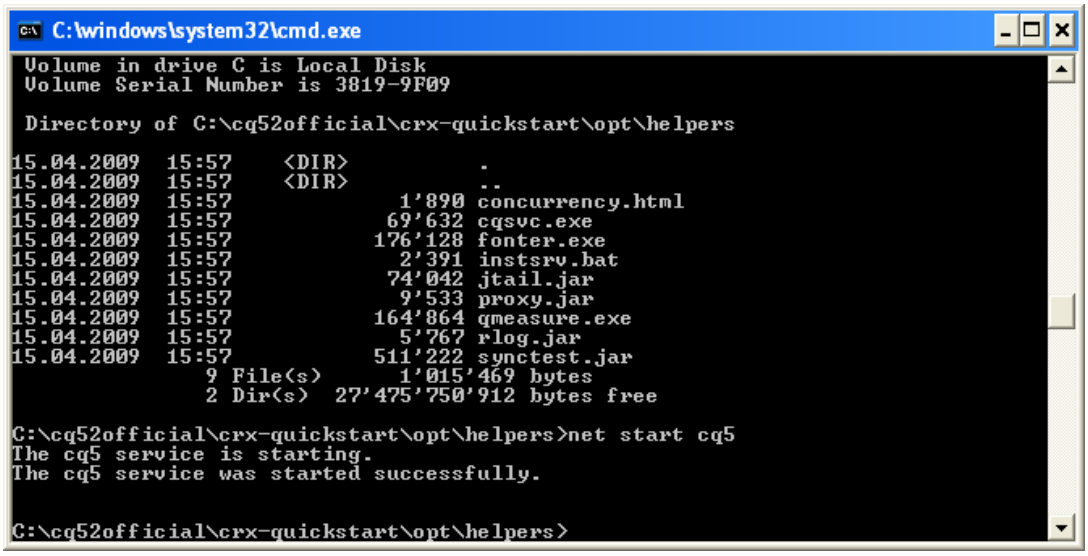
   To verify that the service has been added, open **`Services`** in the **`Administrative Tools`** control panel or type `net start` in the command line. You see **`cq5`** listed as a service.

2. Start the service by doing one of the following:

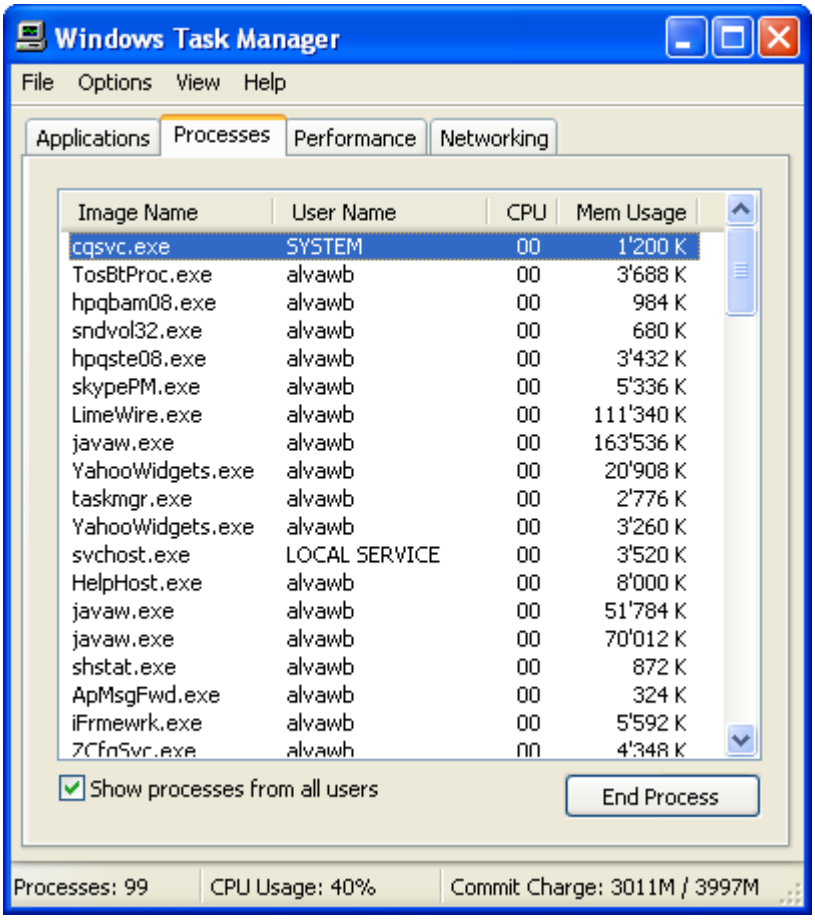
   - In the **`Services`** control panel, click **`cq5`** and click **`Start`**.



- In the command line, type `net start cq5`.



3. Windows indicates that the service is running. CQ WCM starts up and `cqsvc.exe` is visible in the Task Manager. In your web browser, navigate to CQ WCM, for example, `http://localhost:4502` to start using CQ WCM.



To uninstall the service, either click **`Stop`** in the **`Services`** control panel or in the command line, navigate to the folder and type `instsrv.bat -uninstall cq5`. The service gets removed from the list in the **`Services`** control panel or from the list in the command line when you type `net start`.

3.4.4 Further options available from the Quickstart file

Further options and renaming conventions are described in the Quickstart help file, which is available through the `–help` option. To access the help, type:

- `java -jar cq-wcm-quickstart-<version>.jar –help`

```
--------------------------------------------------------------------------------
CQ5 WCM 5.2.0-SNAPSHOT (build #20090326-025803)
--------------------------------------------------------------------------------
Usage:
 Use these options on the Quickstart command line.
--------------------------------------------------------------------------------

-help (--help,-h)                                  Show this help message
-crx.quickstart.server.port (-p,-port) <port>      Set server port number
-gui                                               Show GUI if running on a
                                                   terminal
-crx.quickstart.nobrowser (-nobrowser)             Do not open browser at startup
-unpack                                            Unpack installation files
                                                   only, do not start the server
                                                   (implies -verbose)
-fe (-filename-regexp) <expr>                      Regular expression used to
                                                   select part of the jar
                                                   filename to  use for setting
                                                   system properties. Must
                                                   contain one (group) used to
```

```
                                                extract the part of the
                                                filename to use. Use the java
                                                Pattern class syntax.
-v (-verbose)                                   Do not redirect stdout/stderr
                                                to files and do not close
                                                stdin
-sp <prop> [<prop> ...]                         Set a system property - this
                                                has priority on all other ways
                                                of setting them at startup.
                                                Use like -sp foo=bar to set
                                                'foo' to value 'bar'.
-nofork                                         Do not fork the JVM, even if
                                                not running on a console
-fork                                           Force forking the JVM if
                                                running on a console
-forkargs <args> [<args> ...]                   Additional arguments for the
                                                forked JVM, defaults to
                                                '-Xmx384M '.  Use -- to
                                                specify values starting with
                                                -, example: '-forkargs --
                                                -server'
-rh (--required-heap) <prefix>                  Required heap size in
                                                megabytes, defaults to 256
--------------------------------------------------------------------------------
Quickstart filename options
--------------------------------------------------------------------------------
Usage:
 Rename the jar file, including one of the patterns shown below, to set the
corresponding option. Command-line options have priority on these filename
patterns.
--------------------------------------------------------------------------------

-NNNN            Include -NNNN.jar in the renamed jar filename to run on port
                 NNNN, example: quickstart-8085.jar
-nobrowser       Include -nobrowser in the renamed jar filename to avoid
                 opening the browser at startup, example:
                 quickstart-nobrowser-8085.jar
-publish         Include -publish in the renamed jar filename to run cq5 in
                 "publish" mode, example: cq5-publish-7503.jar
-author          Include -author in the renamed jar filename to run cq5 in
                 "author" mode, example: cq5-author-7502.jar
-crx (or cq)     Start the jar filename with "crx" or "cq" to wait for
                 'QUICKSTART_HOMEPAGE' at the / URL before opening browser
--------------------------------------------------------------------------------
The license.properties file
--------------------------------------------------------------------------------
  The license.properties file stores licensing information, created from the
  licensing form displayed on first startup and stored in the folder from where
  Quickstart is run.
--------------------------------------------------------------------------------
Log files
--------------------------------------------------------------------------------
  Once Quickstart has been unpacked and started, log files can be found under
  crx-quickstart/launchpad/logs, crx-quickstart/server/logs and
  crx-quickstart/logs.
--------------------------------------------------------------------------------
```

3.4.5 Uploading Packages

If you need to install further (often customized) packages for your installation see Uploading Packages in the CRX documentation for detailed instructions.

3.5 How to install CQ5 with an Application Server

The following sections detail how to install CQ5 in conjunction with various application servers:

- WebSphere v6.1
- WebLogic v10.1

- Tomcat v6
- JBoss v4

A generic overview is also given for general usage and information:

- Generic Procedures

3.5.1 WebSphere v6.1

After installing WebSphere v6.1 you:

3.5.1.1 Install CQ5

1. Unpack the installation files of the CQ5 Quickstart into a directory (without starting the server); the installation directory will be referred to as `<cq-installation-dir>`:
   - Start the CQ5 Quickstart jar with the option `-unpack`; for example:

     ```
     java -jar cq-wcm-quickstart-5.2.0.jar -unpack
     ```

     This will create a folder `<cq-installation-dir>crx-quickstart` containing the files and folders used for installation, without actually starting the installation.

     Important

     This *must* be done from the command line. If you open the jar file directly you will activate the Quickstart installation and start the server.
2. Copy the following jar files to the application server folder holding shared libraries:
   a. `CRX\server\lib\container\jcr-1.0.jar`
   b. `CRX\server\lib\container\crx-shared.jar`
3. Restart WebSphere.
4. Deploy the following web applications; they can be found in `<cq-installation-dir>\crx-quickstart\server\webapps`:
   a. CRX webapp; `crx-explorer_crx.war`.

      For example, deploy with the context path `/crx`.

   b. Launchpad webapp; `crx-launchpad.war`.

      For example, deploy with the context path `/launchpad`.
5. Start the two applications.
6. Register your CRX license:
   a. Access your CRX installation:

      `http://<server>:<port>/<context-path>/index.jsp`

      for example: `http://<server>:<port>/crx/index.jsp`

   b. Click the red warning message - “Click here...” (the message is a link).
   c. Enter your license key.

3.5.1.2 Configure the default JDK

WebSphere v6.1 uses JDK 1.5. By default the SAMLv2 JSP JDK source level uses JDK 1.3. As the SAMLv2 sample configuration uses the JDK 1.5 syntax, running it with the default source level will not work. The following steps should be used to configure the source level as 1.5:

1. Within the deployed `crx-explorer_crx.war`, edit `ibm-web-ext.xmi` and add the following configuration parameter to specify the JSP engine:

   ```
   <jspAttributes xmi:id="JSPAttribute_1225281520121"
   name="jdkSourceLevel" value="15"/>
   ```

   Note

   The integer (*n*) referenced in *JSPAttribute_<n>* must be unique within the file.

2. Repeat for `crx-launchpad.war`.

   Note

   The default configuration directory for the web module is:

   ```
   <WAS_ROOT>\profiles\profilename\config\cells\cellname
   \applications\enterpriseappname\ deployments\deployedname
   \webmodulename\WEB-INF\
   ```

   If you have already checked the option *Use Binary Configuration* the files are extracted to the following directory, where they can be edited:

   ```
   <WAS_ROOT>\profiles\profilename\installedApps\nodename
   \enterpriseappname\webmodulename\
   ```

   Where <*WAS_ROOT*> is the root directory of the web application server installation.

3. Restart Websphere.

3.5.1.3 Install your Content Packages

1. Stop CQ5.
2. Edit the workspace configuration file:
   a. Open the following file for edit:

      ```
      <cq-installation-dir>\crx-quickstart\repository\workspaces
      \workspace.xml
      ```

   b. Scroll down to the <*SearchIndex*> parameter section.

   c. Add <*param name="indexingConfiguration" value=""/*> to the <*SearchIndex*> parameter section.
3. Start CQ5.
4. Access the CRX main console.
5. Log in to the `crx.system` workspace as `admin`.
6. Navigate to the **`Package Manager`** in CRX.
7. Upload and install the following CQ5 package from <*cq-installation-dir*>`\crx-quickstart\repository\install\system`:

- WCM Security Content Package; `cq-security-content-<cq-version>.jar`.

8. Switch to the `crx.default` workspace, again as `admin`
9. Upload and install the following CQ5 packages from `<cq-installation-dir>\crx-quickstart\repository\install` in the following order:
   a. Sling Content Package; `0001-cq-wcm-sling-content-<cq-version>.jar`.
   b. WCM Content Package; `0002-cq-wcm-content-<cq-version>.jar`.

3.5.1.4 Enable Replication for Author instances of CQ5

For an author instance of CQ5 you must configure it to start in “author run mode” so that you can perform replications.

1. Open the file: `<cq-installation-dir>\crx-quickstart\launchpad\sling.properties` for edit.
2. Add the following two properties to the file:

```
sling.jcrinstall.folder.name.regexp = .*/(install|config)(.author)?$
sling.run.modes = author
```

3. Restart the `crx-launchpad` web application.

3.5.2 WebLogic v10.3

After installing WebLogic v10.3 and creating your domain you:

3.5.2.1 Configure the Server Locale

When you deploy CQ5 with WebLogic 10.3 you must have the server locale set to *en_US* to avoid errors such as:

```
java.lang.IllegalArgumentException: Bad date header: 'Wed, 12 Nov 2008
16:34:28 GMT'
```

These can occur when, for example, requesting a resource such as `/libs/widgets/0.gif`.

To configure the server locale on Microsoft Windows:

1. Open the **`Control Panel`**.
2. Open **`Regional and Languages Options`**.
3. In the **`Regional Options`** tab, for **`Standards and formats`** select English(United States).

To configure the server locale on Linux or Unix:

- set the environment variable *LANG* to en_US.

3.5.2.2 Enable Basic Authentication Headers

To enable out-of-the-box authentication of users in CQ5, authentication by the application server must be switched off:

1. Open `<WebLogic-installation-dir>/user_projects/domains/<your-domain>/config/config.xml`.
2. Locate the element `<security-configuration>`.

3. Add the following child element to it:

```
<enforce-valid-basic-auth-credentials>
    false
</enforce-valid-basic-auth-credentials>
```

4. If you had already started WebLogic then you will need to restart it.

3.5.2.3 Install CQ5

1. Unpack the installation files of the CQ5 Quickstart into a directory (without starting the server); the installation directory will be referred to as `<cq-installation-dir>`:
   - Start the CQ5 Quickstart jar with the option `-unpack`; for example:

     `java -jar cq-wcm-quickstart-5.2.0.jar -unpack`

     This will create a folder `<cq-installation-dir>crx-quickstart` containing the files and folders used for installation, without actually starting the installation.

     Important

     This *must* be done from the command line. If you open the jar file directly you will activate the Quickstart installation and start the server.

2. Copy the following jar files to the application server folder holding shared libraries:
   a. `CRX\server\lib\container\jcr-1.0.jar`
   b. `CRX\server\lib\container\crx-shared.jar`
3. Restart WebLogic.
4. Deploy the following web applications; they can be found in `<cq-installation-dir>\crx-quickstart\server\webapps`:
   a. CRX webapp; `crx-explorer_crx.war`.

      For example, deploy with the context path `/crx`.

   b. Launchpad webapp; `crx-launchpad.war`.

      For example, deploy with the context path `/launchpad`.
5. Start the two applications.
6. Register your CRX license:
   a. Access your CRX installation:

      `http://<server>:<port>/<context-path>/index.jsp`

      for example: `http://<server>:<port>/crx/index.jsp`

   b. Click the red warning message - “Click here...” (the message is a link).
   c. Enter your license key.

3.5.2.4 Install your Content Packages

1. Stop CQ5.
2. Edit the workspace configuration file:

a. Open the following file for edit:

   `<cq-installation-dir>\crx-quickstart\repository\workspaces\workspace.xml`

b. Scroll down to the <*SearchIndex*> parameter section.

c. Add <*param name="indexingConfiguration" value=""/*> to the <*SearchIndex*> parameter section.

3. Start CQ5.
4. Access the CRX main console.
5. Log in to the `crx.system` workspace as `admin`.
6. Navigate to the **`Package Manager`** in CRX.
7. Upload and install the following CQ5 package from `<cq-installation-dir>\crx-quickstart\repository\install\system`:
   - WCM Security Content Package; `cq-security-content-<cq-version>.jar`.
8. Switch to the `crx.default` workspace, again as `admin`
9. Upload and install the following CQ5 packages from `<cq-installation-dir>\crx-quickstart\repository\install` in the following order:
   a. Sling Content Package; `0001-cq-wcm-sling-content-<cq-version>.jar`.
   b. WCM Content Package; `0002-cq-wcm-content-<cq-version>.jar`.

3.5.2.5 Enable Replication for Author instances of CQ5

For an author instance of CQ5 you must configure it to start in “author run mode” so that you can perform replications.

1. Open the file: `<cq-installation-dir>\crx-quickstart\launchpad\sling.properties` for edit.
2. Add the following two properties to the file:

```
sling.jcrinstall.folder.name.regexp = .*/(install|config)(.author)?$
sling.run.modes = author
```

3. Restart the `crx-launchpad` web application.

3.5.3 Tomcat v6

After installing Tomcat v6 you:

3.5.3.1 Configure Tomcat access accounts

Tomcat enables neither `admin` nor `manager` access at installation.

Therefore you have to manually edit `tomcat-users.xml` to allow access for these accounts:

1. Navigate to the Tomcat configuration folder.
2. Edit `tomcat-users.xml` to include access for `admin` and `manager`. The configuration should look similar to the following example:

```
<?xml version='1.0' encoding='utf-8'?>
<tomcat-users>
    <role rolename="manager"/>
    <role rolename="tomcat"/>
    <role rolename="admin"/>
    <role rolename="role1"/>
    <user username="both" password="tomcat" roles="tomcat,role1"/>
    <user username="tomcat" password="tomcat" roles="tomcat"/>
    <user username="admin" password="admin" roles="admin,manager"/>
    <user username="role1" password="tomcat" roles="role1"/>
</tomcat-users>
```

3.5.3.2 Install CQ5

1. Unpack the installation files of the CQ5 Quickstart into a directory (without starting the server); the installation directory will be referred to as `<cq-installation-dir>`:

   - Start the CQ5 Quickstart jar with the option `-unpack`; for example:

     `java -jar cq-wcm-quickstart-5.2.0.jar -unpack`

     This will create a folder `<cq-installation-dir>crx-quickstart` containing the files and folders used for installation, without actually starting the installation.

     Important

     This *must* be done from the command line. If you open the jar file directly you will activate the Quickstart installation and start the server.

2. Copy the following jar files to the application server folder holding shared libraries:

   a. `CRX\server\lib\container\jcr-1.0.jar`

   b. `CRX\server\lib\container\crx-shared.jar`

3. Restart Tomcat.

4. Deploy the following web applications; they can be found in `<cq-installation-dir>\crx-quickstart\server\webapps`:

   a. CRX webapp; `crx-explorer_crx.war`.

      For example, deploy with the context path `/crx`.

   b. Launchpad webapp; `crx-launchpad.war`.

      For example, deploy with the context path `/launchpad`.

5. Start the two applications.

6. Register your CRX license:

   a. Access your CRX installation:

      `http://<server>:<port>/<context-path>/index.jsp`

      for example: `http://<server>:<port>/crx/index.jsp`

   b. Click the red warning message - “Click here...” (the message is a link).

   c. Enter your license key.

3.5.3.3 Install your Content Packages

1. Stop CQ5.
2. Edit the workspace configuration file:
   a. Open the following file for edit:

      `<cq-installation-dir>\crx-quickstart\repository\workspaces\workspace.xml`

   b. Scroll down to the *<SearchIndex>* parameter section.

   c. Add *<param name="indexingConfiguration" value=""/>* to the *<SearchIndex>* parameter section.
3. Start CQ5.
4. Access the CRX main console.
5. Log in to the `crx.system` workspace as `admin`.
6. Navigate to the **`Package Manager`** in CRX.
7. Upload and install the following CQ5 package from `<cq-installation-dir>\crx-quickstart\repository\install\system`:
   - WCM Security Content Package; `cq-security-content-<cq-version>.jar`.
8. Switch to the `crx.default` workspace, again as `admin`
9. Upload and install the following CQ5 packages from `<cq-installation-dir>\crx-quickstart\repository\install` in the following order:
   a. Sling Content Package; `0001-cq-wcm-sling-content-<cq-version>.jar`.

   b. WCM Content Package; `0002-cq-wcm-content-<cq-version>.jar`.

3.5.3.4 Enable Replication for Author instances of CQ5

For an author instance of CQ5 you must configure it to start in “author run mode” so that you can perform replications.

1. Open the file: `<cq-installation-dir>\crx-quickstart\launchpad\sling.properties` for edit.
2. Add the following two properties to the file:

```
sling.jcrinstall.folder.name.regexp = .*/(install|config)(.author)?$
sling.run.modes = author
```

3. Restart the `crx-launchpad` web application.

3.5.4 JBoss v4

After installing JBoss v4 you:

3.5.4.1 Install CQ5

1. Unpack the installation files of the CQ5 Quickstart into a directory (without starting the server); the installation directory is referred to as *<cq-installation-dir>*:

- Start the CQ5 Quickstart jar with the option `-unpack`; for example:

  ```
  java -jar cq-wcm-quickstart-5.2.0.jar -unpack
  ```

  This creates a folder <*cq-installation-dir*>`crx-quickstart` containing the files and folders used for installation, without actually starting the installation.

Important

This *must* be done from the command line. If you open the jar file directly you will activate the Quickstart installation and start the server.

2. Copy the following jar files to the application server folder holding shared libraries (`<JBOSS_HOME]\server\default\lib>`):

   a. `CRX\server\lib\container\jcr-1.0.jar`

   b. `CRX\server\lib\container\crx-shared.jar`

3. Restart JBoss.

4. Unpack the `crx-explorer_crx.war` file located in the <*cq-installation-dir*>`\crx-quickstart\server\webapps` folder.

Note

In Windows, change the `.war` extension to `.zip` and unpack like any zip file. In Linux, type `jar xvf crx-explorer_crx.war` to unpack.

5. In the **`WEB-INF`** folder, open `log4j.xml`.

6. Remove or comment the line `<appender-ref ref="console"/>` which is in the Loggers section of the file and save your changes and exit the file. This disables console logging in the CRX web application.

7. In the **`WEB-INF`** folder, navigate to the **`lib`** folder and delete the following files:

   - jcr-1.0.jar
   - jackrabbit-api-1.4.jar
   - day-commons-naming-1.1.1.jar
   - crx-api-1.4.1.jar

8. Pack the `crx-explorer_crx.war` file.

Note

In Windows, run the zip utility to compress it and rename the `crx-explorer_crx.zip` file to `crx-explorer_crx.war`. In Linux, type `jar cvf crx-explorer_crx.war`.

9. Deploy the following web applications; they can be found in <*cq-installation-dir*>`\crx-quickstart\server\webapps`:

   a. CRX webapp; `crx-explorer_crx.war`.

   For example, deploy with the context path `/crx`.

   b. Launchpad webapp; `crx-launchpad.war`.

For example, deploy with the context path `/launchpad`.

JBoss supports *Hot Deployment*, so you can simply drag the two files to `<JBOSS_HOME>\server\default\deploy`.

10. Start the two applications.
11. Register your CRX license:
    a. Access your CRX installation:

       `http://<server>:<port>/<context-path>/index.jsp`

       for example: `http://<server>:<port>/crx/index.jsp`

    b. Click the red warning message - “Click here...” (the message is a link).

    c. Enter your license key.

3.5.4.2 Configure the JBoss Server Login Module

By default JBoss' default login configuration attempts to authenticate users against a list of users in the `users.properties` file. You must configure JBoss as follows to let login attempts by unknown users to pass to the web application (CRX Explorer). The web application will then process authentication by itself.

1. Open the file for editing:

   `<JBOSS_HOME>\server\default\conf\login-config.xml`

2. In the section *`application-policy name="other"`* (at the bottom of the file) add the attribute:

   `unauthenticatedIdentity="nobody"`

   to the *`login-module`* entry.

3.5.4.3 Install your Content Packages

Once you have installed CQ5 you will want to install content packages.

1. Stop CQ5.
2. Edit the workspace configuration file:
   a. Open the following file for edit:

      `<cq-installation-dir>\crx-quickstart\repository\workspaces\workspace.xml`

   b. Scroll down to the *<SearchIndex>* parameter section.

   c. Add *<param name="indexingConfiguration" value=""/>* to the *<SearchIndex>* parameter section.
3. Start CQ5.
4. Access the CRX main console:

   `http://<server>:<port>/crx-explorer_crx/index.jsp`

5. Log in to the `crx.system` workspace as `admin`.
6. Navigate to the **`Package Manager`** in CRX.
7. Upload and install the following CQ5 package from `<cq-installation-dir>\crx-quickstart\repository\install\system`:
   - WCM Security Content Package; `cq-security-content-<cq-version>.jar`.
8. Switch to the `crx.default` workspace, again as `admin`
9. Upload and install the following CQ5 packages from `<cq-installation-dir>\crx-quickstart\repository\install` in the following order:
   a. Sling Content Package; `0001-cq-wcm-sling-content-<cq-version>.jar`.
   b. WCM Content Package; `0002-cq-wcm-content-<cq-version>.jar`.
10. Restart JBoss.

3.5.4.4 Enable Replication for Author instances of CQ5

For an author instance of CQ5 you must configure it to start in “author run mode” so that you can perform replications.

1. Open the file: `<cq-installation-dir>\crx-quickstart\launchpad\sling.properties` for edit.
2. Add the following two properties to the file:

```
sling.jcrinstall.folder.name.regexp = .*/(install|config)(.author)?$
sling.run.modes = author
```

3. Restart the `crx-launchpad` web application.

3.5.5 Generic Procedures

After installing the appropriate web application server you:

3.5.5.1 Generic Installation Procedure

This section provides generic information about installing CQ5 with an application server.

1. Unpack the installation files of the CQ5 Quickstart into a directory (without starting the server); the installation directory will be referred to as `<cq-installation-dir>`:
   - Start the CQ5 Quickstart jar with the option `-unpack`; for example:

     `java -jar cq-wcm-quickstart-5.2.0.jar -unpack`

     This will create a folder `<cq-installation-dir>crx-quickstart` containing the files and folders used for installation, without actually starting the installation.

     Important

     This *must* be done from the command line. If you open the jar file directly you will activate the Quickstart installation and start the server.
2. Copy the following jar files to the application server folder holding shared libraries:
   a. `CRX\server\lib\container\jcr-1.0.jar`

b. `CRX\server\lib\container\crx-shared.jar`

3. Restart your web application server.
4. Deploy the following web applications; they can be found in `<cq-installation-dir>\crx-quickstart\server\webapps`:
   a. CRX webapp; `crx-explorer_crx.war`.

      For example, deploy with the context path `/crx`.

   b. Launchpad webapp; `crx-launchpad.war`.

      For example, deploy with the context path `/launchpad`.
5. Start the two applications.
6. Register your CRX license:
   a. Access your CRX installation:

      `http://<server>:<port>/<context-path>/index.jsp`

      for example: `http://<server>:<port>/crx/index.jsp`

   b. Click the red warning message - “Click here...” (the message is a link).

   c. Enter your license key.

3.5.5.2 Install Content Packages

Once you have installed CQ5 you will want to install content packages.

1. Stop CQ5.
2. Edit the workspace configuration file:
   a. Open the following file for edit:

      `<cq-installation-dir>\crx-quickstart\repository\workspaces\workspace.xml`

   b. Scroll down to the `<SearchIndex>` parameter section.

   c. Add `<param name="indexingConfiguration" value=""/>` to the `<SearchIndex>` parameter section.
3. Start CQ5.
4. Access the CRX main console.
5. Log in to the `crx.system` workspace as `admin`.
6. Navigate to the **`Package Manager`** in CRX.
7. Upload and install the following CQ5 package from `<cq-installation-dir>\crx-quickstart\repository\install\system`:
   - WCM Security Content Package; `cq-security-content-<cq-version>.jar`.
8. Switch to the `crx.default` workspace, again as `admin`

9. Upload and install the following CQ5 packages from `<cq-installation-dir>\crx-quickstart\repository\install` in the following order:

   a. Sling Content Package; `0001-cq-wcm-sling-content-<cq-version>.jar`.

   b. WCM Content Package; `0002-cq-wcm-content-<cq-version>.jar`.

3.5.5.3 Enable Replication for Author instances of CQ5

For an author instance of CQ5 you must configure it to start in “author run mode” so that you can perform replications.

1. Open the file: `<cq-installation-dir>\crx-quickstart\launchpad\sling.properties` for edit.
2. Add the following two properties to the file:

```
sling.jcrinstall.folder.name.regexp = .*/(install|config)(.author)?$
sling.run.modes = author
```

3. Restart the `crx-launchpad` web application.

3.6 Actions after Installation

Although there are many possibilities to configure CQ WCM, certain actions should be taken, or at least reviewed immediately after installation:

- For admin accounts change the default passwords to ensure your system is secure.
- Read the remaining points on the security checklist for tasks required to ensure that your system remains secure.
- Review the list of default users and groups which are installed with CQ WCM. Check whether you want to take action on any other accounts - see Security and User Administration for further details.

3.7 Troubleshooting

This section includes detailed information on logs available to help you troubleshoot and also includes information on some of the problems you might encounter when installing CQ WCM.

3.7.1 Working with Logs

CQ WCM records detailed logs. After you unpack and start Quickstart, you can find logs in various locations, including:

- `crx-quickstart/launchpad/logs`
- `crx-quickstart/server/logs`
- `crx-quickstart/repository/...`
- `crx-quickstart/logs`

3.7.1.1 Finding the Log Files

Various log files are held on the file server where you installed CQ5:

- `<cq-installation-dir>\crx-quickstart\launchpad\logs`

- `error.log`

  Error messages, of varying levels of severity, related to the CRX Quick launchpad are registered here.

- `<`*`cq-installation-dir`*`>\crx-quickstart\logs`
  - `access.log`

    All access requests to CQ WCM and the repository are registered here.

  - `error.log`

    Error messages (of varying levels of severity) are registered here.

  - `request.log`

    Each access request is registered here together with the response.

  - `server.log`

    All actions made by the server are registered here.

  - `stderr.log`

    Holds error messages, again of varying levels of severity, generated during startup.

  - `stdout.log`

    Holds logging messages indicating events during startup.

- `<`*`cq-installation-dir`*`>\crx-quickstart\logs\crx`
  - `error.log`

    Error messages (of varying levels of severity) related to the repository are registered here.

  - `translation.log`

    Translation journaling.

- `<`*`cq-installation-dir`*`>\crx-quickstart\repository\`
  - `revision.log`

    Revision journaling information.

- `<`*`cq-installation-dir`*`>\crx-quickstart\repository\repository\index`
  - `redo.log`

    Redo event journaling.

- `<`*`cq-installation-dir`*`>\crx-quickstart\repository\shared\journal`
  - `journal.log<.`*`x`*`>`

    Event journaling on the repository; multiple versions.

- `<`*`cq-installation-dir`*`>\crx-quickstart\repository\workspaces\<`*`workspace`*`>\index`
  - `indexing_queue.log`

Index journaling.

- `redo.log`

  Redo event journaling.

- *<cq-installation-dir>*`\crx-quickstart\server\logs`
  - `access.log`

    All access requests to CQ WCM and the repository are registered here.

  - `startup.log`

    Server startup.

3.7.1.2 Activating the DEBUG Log Level

The default log level is INFO, that is, DEBUG messages are not logged.

To activate DEBUG log level, use the CRX explorer to set the `/libs/sling/config/org.apache.sling.commons.log.LogManager/org.apache.sling.commons.log.level` property to `debug`.

Caution

Do not leave the log at the DEBUG log level longer than necessary, as it generates a lot of log entries, thus consuming resources.

A line in the debug file usually starts with DEBUG, and then provides the log level, the installer action and the log message. For example:

```
DEBUG 3 WebApp Panel: WebApp successfully deployed
```

The log levels are as follows:

Table 3.1. Log Levels

0	Fatal error	The action has failed, and the installer cannot proceed.
1	Error	The action has failed. The installation proceeds, but a part of CQ WCM was not installed correctly and will not work.
2	Warning	The action has succeeded but encountered problems. CQ WCM may or may not work correctly.
3	Information	The action has succeeded.

3.7.1.3 Create a Custom Log File

In certain circumstance you may want to create a custom log file. You can do this by:

1. Open the **`Apache Felix Web Management Console`**; for example at `http://localhost:4502/system/console/`.
2. Go to the **`Configuration`** page.
3. Select *`Sling Logging Writer Configuration`* from the **`Factory Configurations`** drop down list.
4. Enter the name of your custom log file; for example, `logs/custom.log`:

5. Click **Save**.

   A new entry is now listed in the **Configurations** drop down list.

6. Select *Sling Logging Logger Configuration* from the **Factory Configurations** drop down list.

7. Configure the following:

   a. Set the **Log Level**; for example *Debug*.

   b. Enter the name of your **Log File** - the same as earlier; in this example `logs/custom.log`.

   c. Set the **Categories** to *com.day.cq.wcm.foundation*.

8. Click **Save**.
9. Restart CQ.

Note

This is necessary to ensure that any static loggers used access the new configuration.

10. Read your new log file with your chosen tool.

3.7.1.4 Using the Verbose Option for Troubleshooting

When you start CQ WCM, you can add the `-v (verbose)` option to the command line as in: `java -jar cq-wcm-quickstart-<version>.jar -v.`

The verbose option displays some of the Quickstart log output on the console, so it can be used for troubleshooting.

3.7.2 Common Installation Issues

The following section describes some installation issues and their solutions.

3.7.2.1 Double-clicking the Quickstart jar does not have any effect or opens the jar file with another program (for example, archive manager)

This usually indicates a problem with the way your operating system's desktop environment is configured to open files with extension `.jar`. It may also indicate that you do not have Java installed, or that you are using an unsupported version of Java.

As jar files use the ubiquitous ZIP format, some of the archiving programs may automatically configure the desktop to open jar files as archive files.

To troubleshoot, do the following:

- Double check that you have at least Java version 1.5 installed.
- Try a context menu (usually right-mouse click) on the CQ WCM Quickstart, and select "Open With...."
- Check if Java or Sun Java is listed, and try to run CQ WCM with it. If you have multiple Java versions installed, select the supported one.

Note

If you succeed with this step, and your operating systems offers an option to always use the selected program to run the .jar files, select it. Double-clicking should work from now on.

- Sometimes reinstalling the supported Java version helps restore the correct association.
- You can always run CRX using the command line or start/stop scripts as described earlier in this document.

3.7.2.2 My application running on CRX throws out-of-memory errors

CRX itself has a very low memory footprint. If the application running within CRX has bigger memory requirements or requests memory-heavy operations (for example, large transactions), the JVM instance where CRX runs needs to be started with appropriate memory settings. Use Java command options to define memory settings of the JVM (for example, java -Xmx512m -jar crx*.jar to set heapsize to 512MB).

Note

Specify the memory setting option while starting CQ WCM from the command line. The CQ WCM start/stop scripts or custom scripts for managing CQ WCM startup can also be modified to define the required memory settings.

3.7.2.3 The CQ WCM Welcome screen does not display in the browser after double-clicking CQ WCM Quickstart

In certain situations, the CQ WCM Welcome screens does not automatically display even though the repository itself is successfully running. This may depend on operating system setup, browser configuration, or similar factors.

The usual symptom is that the CQ WCM Quickstart window displays "CQ WCM starting up, waiting for server startup...." If that message displays for a relatively long time, enter the CQ WCM URL into the browser window manually, using the default 4502 port, or the port on which the instance is running: `http://localhost:4502/`.

Also, logs may reveal the reason for the browser not starting.

Sometimes, the CQ WCM Quickstart window has the message "CQ WCM running on http://localhost:port/" and the browser does not start automatically. In this case, click on the URL in the CQ WCM Quickstart window (it is a hyperlink) or manually enter the URL in the browser.

If everything else fails, check the logs to find out what has happened.

3.8 Uninstalling CQ WCM

Because CQ WCM installs into a single directory, there is no need for an uninstall utility. Uninstalling can be as simple as deleting the entire installation directory, although how you uninstall CQ WCM depends on what you want to achieve and what persistent storage you use.

If persistent storage is embedded in the installation directory, for example, in the default TarPM installation, deleting folders removes data as well.

Note

Day highly recommends that you back up your repository before deleting CQ WCM. If you delete the entire `<cq-installation-directory>`, you will delete the repository. To keep the repository data before deleting, move or copy the `<cq-installation-directory>/crx-quickstart/repository` folder somewhere else before deleting the other folders.

If your installation of CQ WCM uses external storage, for example, a database server, removing folder does not remove the data automatically, but it does remove the storage configuration, which makes restoring the JCR content difficult.

4 Upgrading to CQ5

4.1 Upgrading from CQ 5.1 to CQ 5.2

4.1.1 Important information to read before starting the upgrade

Please read the following points and take action where necessary:

- This upgrade procedure only works if the CQ 5.1 installation was performed with Quickstart. As opposed to installation with a third party application server.
- You must test the operation of the upgraded website; highly customized items may need to be upgraded separately.
- Beginning with CQ 5.2 the Text Components add a HTML paragraph tag around the text.

  Prior versions did not always do that. Adding the paragraph tag might cause visual differences.

  Make advance tests (edit a test object) to confirm whether this occurs with your installation. If necessary adapt the CSS selector, and note the changes necessary after the upgrade.
- Check whether you have any extremely large character strings in your repository.

  See Step 6 for further information.
- If OSGi bundles and configurations have been setup directly from the OSGi console (as opposed to storing them in the JCR repository) the configuration should be saved before the upgrade. This is done by:
  - saving the output of `/system/console/config`

  and
  - saving a copy of the `crx-quickstart/launchpad` folder

  As the `crx-quickstart/launchpad` folder is deleted before the upgrade, then the bundles and configurations must be restored manually after the upgrade,
- Be aware that certain default content is impacted by the upgrade:
  - Default content under `/etc` is replaced by the upgrade.

    If you need this content then please save the original content tree (for example, as etc_51) before the upgrade. After the upgrade is complete you can re-insert any differences manually.
  - Any changes that you have made to the repository folders containing CQ 5.1 software (`/libs` and others) might be lost after this upgrade.
  - Content packages delete everything under the path(s) that they impact. Therefore, content created under `/content/geometrixx` will be deleted by the upgrade to CQ 5.2.
  - The upgrade procedure deletes everything under:
    - /content/geometrixx
    - /apps/geometrixx
    - /etc/designs/geometrixx
    - /libs/

- The `cq-security-content` package is not reinstalled when upgrading, the old version is kept.

  You must manually update the permissions after the upgrade. You must make the group `contributor` a member of the group `uploader`.

- Replication configuration settings are reset during the update. You need to reconfigure these after the upgrade, using the new **`Replication`** configuration under **`Tools`**.

- The default workflows are overwritten during the upgrade, so any customizations are lost. Please take a copy of these if required.

4.1.2 Preparing for the upgrade from CQ 5.1 to CQ 5.2

All preparation steps must be checked to ensure a smooth upgrade:

1. Ensure you have a full backup of your CQ 5.1 instance.
2. Ensure that the CQ 5.1 installation was made using Quickstart.
3. Ensure that you have at least 2GB of free diskspace.
4. Ensure that the system account you are using has sufficient privileges to replace the jar file.
5. Ensure that the `crx-quickstart/repository` folder that contains your repository data is binary compatible between the CQ 5.1 and 5.2 versions.

   Note

   Ensure that no changes (hotfixes or customizations) have been applied to the repository folder that make it incompatible with the 5.2 version that is about to be installed.

   If you are in doubt please check with Day.

6. Request the `cleanupMessage.jsp` script from Day.

4.1.3 Upgrading your CQ 5.1 instance to CQ 5.2

Updating your CQ 5.1 instance to a CQ 5.2 instance requires the following steps:

Important

Please read the section called “Important information to read before starting the upgrade” before starting the upgrade.

Important

Please complete all preparation steps as in the section called “Preparing for the upgrade from CQ 5.1 to CQ 5.2”.

1. Delete the `crx-quickstart/launchpad` folder.
2. Run the cleanupMessage script:
   1. Copy the `cleanupMessage.jsp` file to your author environment:

      `<cq-installation-dir>/crx-quickstart/server/runtime/0/_crx/config/cleanupMessage.jsp`

      Adapt the path as necessary.

   2. Start your CQ 5.1 instance, then navigate to your CRX console, for example:

`http://localhost:4502/crx` (adapt as necessary).

3. Log in as `admin` to the workspace where CQ is installed; for example `crx.default`.
4. Open the script at:

   `http://localhost:4502/crx/config/cleanupMessage.jsp` (adapt as necessary).
5. Only the following properties are changed:
   - with the name “message” under the specified node
   - those that are longer than the specified size

   The default settings should be correct in all cases.
6. Start the script using 'clean'.
7. The output is formed of the nodes that contain a message property.

   The prefix “cleaned:” means the text was truncated, while “ignored:” means the text was short and therefore unchanged.

Tip

After the script is run, you may reduce the repository size using Tar PM Optimization.

3. Stop your CQ 5.1 instance.
4. Copy the new jar file to the location of the currently installed jar file:

   `<cq-installation-dir>/`

Caution

If you had changed the name of your CQ 5.1 jar file, then rename the CQ 5.2 jar file to that of the existing file.

For example, if it had been renamed for an author environment:

`<cq-installation-dir>/cq-author-4502.jar`

Or a corresponding publish environment:

`<cq-installation-dir>/cq-publish-4503.jar`

5. Start CQ 5.2; in the same manner as with your CQ 5.1 instance, by a **`double-click`** on the jar file, or from the command line.

Note

This step will take longer than a normal startup as the new 5.2 content packages have to be installed.

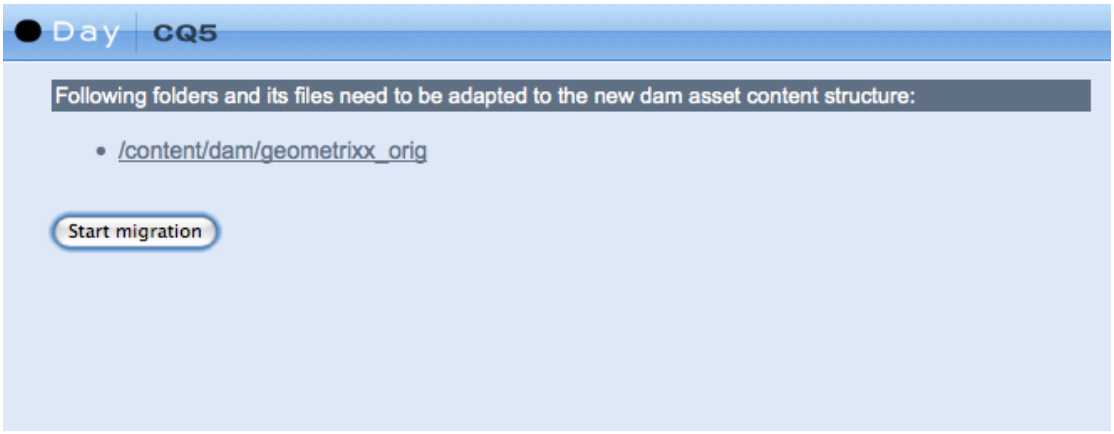
6. Restart CQ 5.2. The upgrade will now be complete.

4.1.4 Upgrading your CQ 5.1 Digital Assets

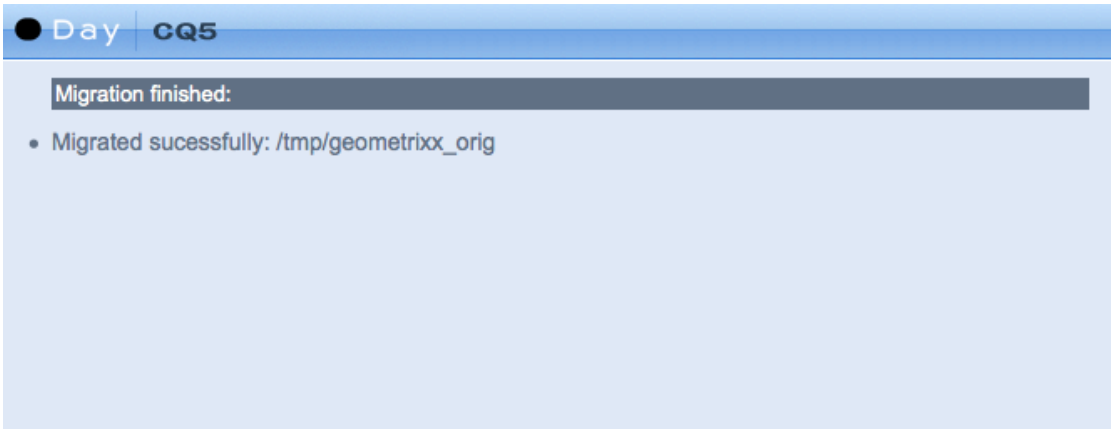
The structure of digital assets has changed between CQ 5.1 and CQ 5.2. For this reason a wizard is included to automate the process of moving the assets to the new structure:

1. Ensure that you have a backup of your newly upgraded repository.
2. Start your newly upgraded CQ 5.2 instance.
3. Navigate to:

   `http://localhost:4502/libs/dam/migrate.html`



4. Click **`Start migration`**. A status message will be shown upon completion:



Note

After the upgrade script has finished it might take some time until all assets reappear for use, as they are being processed in the background. You can monitor `stdout.log` to see when everything has been finished.

Note

The upgrade removes the replication state on the assets.

4.1.5 Tasks to perform after the upgrade

The following tasks should be performed after the upgrade:

1. The CQ 5.1 jar file is not needed after the upgrade and if still present, should be removed from the installation folder to avoid confusion.
2. Reinstall any customized OSGi bundles (if you had manually installed them before).
3. Restore any customized configurations you had made on the file system:
   - `/server/runtime/0/_crx/WEB-INF/repository.xml`; for example, search engine settings.

- `/server/etc/`; for example, LDAP configurations.

4. Restore any customized configurations from the Apache Felix Console:
   - Mail Service
   - Day Commons GFX Font Helper
   - SSO Authentication Handler
   - Apache Sling Resource Resolver

   Note

   CQ 5.2 adds a new default rule */content/-/* that you might want to remove.

5. Restore any customized configurations in CRX:
   - URL to `tracker.js`: `/libs/wcm/config.publish/com.day.cq.wcm.core.stats.PageViewStatistics`

6. Reconfigure the replication configuration settings, using the new **`Replication`** configuration under **`Tools`**.

7. The `cq-security-content` package is not reinstalled when upgrading, the old version is kept.

   You must manually update the permissions after the upgrade, by making the group `contributor` a member of the group `uploader`.

8. Check all forms on the site. If you had used the custom form action you need to change the dialog to fit the new structure.

9. Beginning with CQ 5.2 the Text Components add a HTML paragraph tag around the text.

   Prior versions did not do always do that. Adding the paragraph tag might cause visual differences. If necessary adapt the CSS selector - as according to your tests in Important information to read before starting the upgrade.

4.1.6 The final status of your upgraded instance

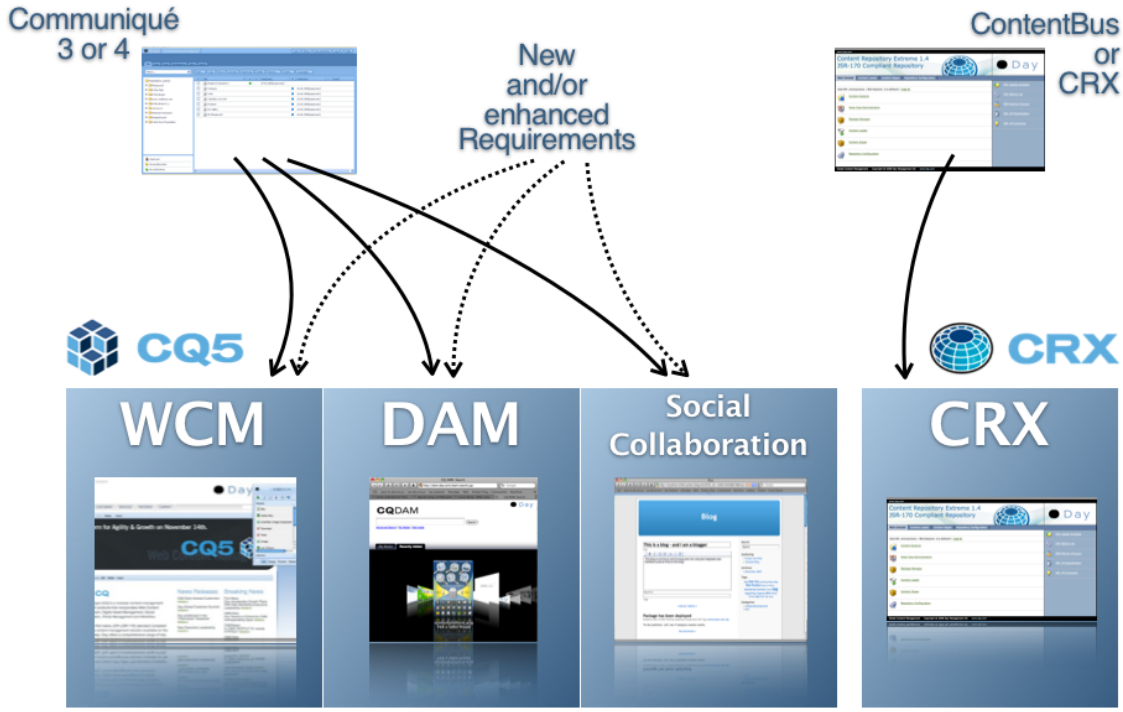
The resulting system is identical to a fresh CQ 5.2 installation, with the user's content preserved (assuming it was not stored in any repository folders that contained the CQ 5.1 software).

4.2 Upgrading from Communiqué 3 or 4

4.2.1 An Overview of the Upgrade Process

An Upgrade enables the automated, or partially automated, conversion of an existing instance, of either Communiqué 3 or 4, to CQ5.

Therefore Day has provided a *Upgrade Tool* for existing customers. This provides a straight-forward, time-efficient upgrade. This transfers the basic elements from Communiqué 3 or 4; you can then extend these to cater for new functionality available in CQ5:



4.2.1.1 What the Upgrade Tool covers

The automatic upgrade can include different parts of the instance, including:

- Content
- Designs
- Components
- Templates
- User and Group accounts, with ACLs
- Version History
- Audit Log
- Log files
- Media Library
- CFC Dialogs
- Workflows
- MSM

Note

Highly customized items might need to be upgraded separately.

4.2.1.2 What you need to start an Upgrade

- The URL to your existing Communiqué 3 or 4 installation.

- A fresh installation of CQ5; as installed without content or customization.

4.2.1.3 What the Upgrade Tool generates

After upgrading you will have an instance that contains content, design, and so on (as above). The user can start to edit content on this new CQ5 instance.

Content Pages
: Upgrade of a page will include the versioning history and operates as follows:

1. Create the CQ5 page from the oldest version in the version history.
2. For each version in the version history:
   a. Check in the page node to create a version.
   b. Check out the page node for update.
   c. Update to the version in the history.

The resulting CQ5 page holds the most recent version (HEAD) and is checked out.

Components and Templates
: Upgrading components consists of multiple parts:

- Descriptor - the component descriptor (such as component name) is upgraded into the node data of the new component.
- `display.any` - this is upgraded into the display descriptor of the new node structure.
- Script(s) - the component scripts are upgraded to the new location using the new naming convention.

To ease the upgrade, a limited ContentBus API is available to components through a JCR Adapter.

Audit Log
: Importing the Communiqué 4 Audit Log is a straight import of the file entries into a node structure.

User
: The upgrade of user definitions consists of several parts:

1. For each user a principal is generated from the User Page.
2. ACLs, these are processed on the same basis as Groups.
3. Any additional data stored, is upgraded into a Preferences structure in the CQ5 user home.

Groups
: New group principals are created and the appropriate users assigned to these groups.

Component coded applications
: Will be upgraded, with the exception of objects of the type `CFC.Tools`.

Media Library
: `/etc/medialib` will be moved to `/content/dam`.

4.2.2 How to Upgrade Your Communiqué Instance (3 or 4) to CQ5

The upgrade tool allows you to launch a standardized upgrade process. It is delivered with a configuration to be used “out of the box,” though several settings can be updated when necessary.

You:

- provide the URL of the Communiqué instance to be upgraded, together with the superuser access credentials
- can configure a list of page handles to be included in the upgrade
- can set debug parameters if required.

Note

The following procedure uses a standard Communiqué 4 installation, complete with Playground and DesignGround, to illustrate the upgrade process.

Important

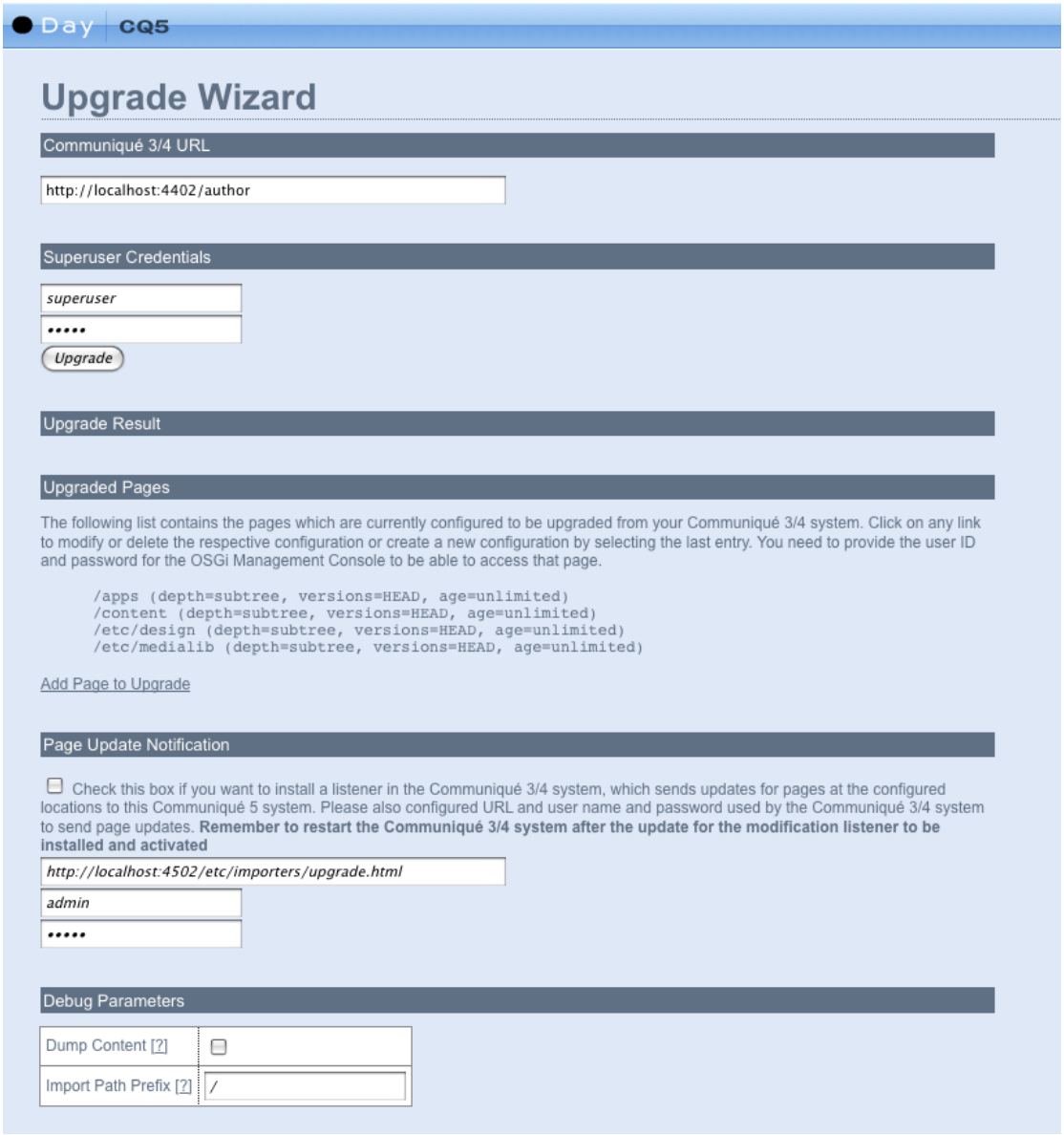
If the remote Communiqué 4 instance is accessed through the dispatcher, the dispatcher configuration must not exclude `/system` from being served. Many dispatcher configurations exclude `/system`. You can also list the URLs that need to be accessible through the dispatcher for the upgrade script.

1. Navigate to the Upgrade window, by one of the following methods:

   a. Select the **`Tools`** tab in CQ WCM, click on **`Importers`** to open the folder, then **`Double-click`** on the **`Upgrade from CQ3/CQ4`** Page to open.

   b. Navigate directly to `http://<cq-context-path>/etc/importers/upgrade.html` for example, `http://localhost:4502/etc//importers/upgrade.html` (if not already logged in you will be required to).

   In either case the following form displays:

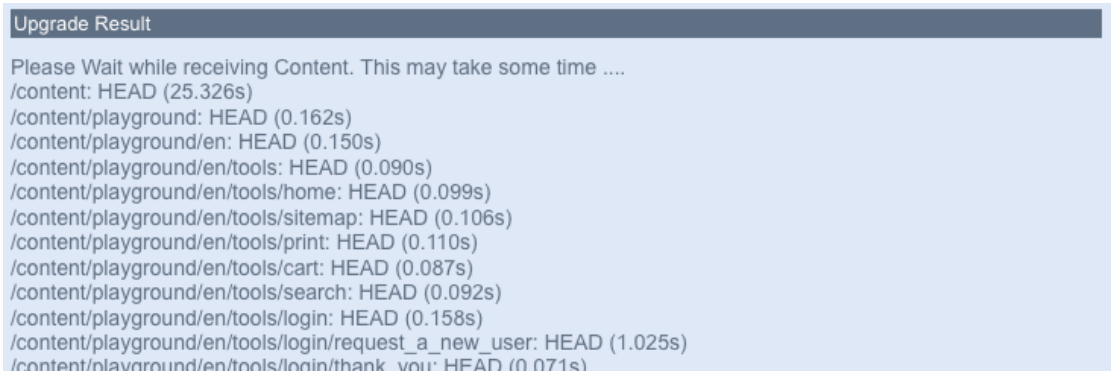


2. Update any of the default settings if required.

Important

The upgrade tool uses the password `superuser` as default. Please ensure that the password is correct for the instance you are upgrading.

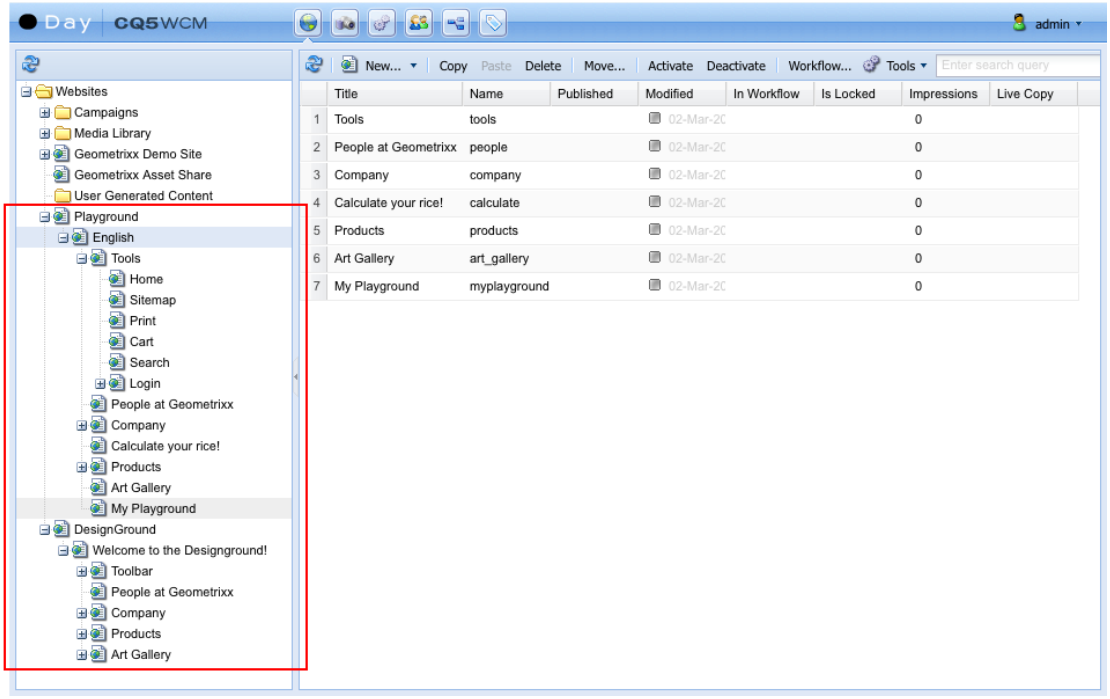
3. Click **`Migrate`** to start the process. A status message will shown and information about the individual pages being processed will be added while the upgrade is running:



4. When complete you can access the upgraded items in your CQ5 instance:

Important

You must test the operation of the upgraded website; highly customized items may need to be upgraded separately.



Note

This mechanism can also be used as the basis for a customized upgrade routine, developed as part of your project.

5 Configuring CQ WCM

CQ WCM is installed with default settings for all parameters which allows it to run straight “out of the box”. However, you can configure CQ WCM for your own specific requirements.

This can be done either:

In the repository (CRX)
: A sub-set of configuration files are available in the repository. This ensures that copying, or replicating, repository contents recreates identical configurations.

Apache Felix Web Management Console
: This is the standard location for configuring OSGi bundles.

In CQ WCM
: Various aspects can be configured within CQ WCM itself, many using the Tools tab. For example, replication agents.

In the file server
: A few configuration files reside within the file server.

Important

Configuring CQ WCM is straightforward, but you must be aware that:

- Certain changes can have a major impact on the application(s). For this reason, ensure you have the necessary experience and knowledge before you start to configure CQ, and make only the changes which you know are required.
- Any changes made via the OSGi console are *immediately* applied to the running system (no restart is required).

Tip

Performance is a major consideration when configuring any system. See How to Monitor Performance for further details on this issue.

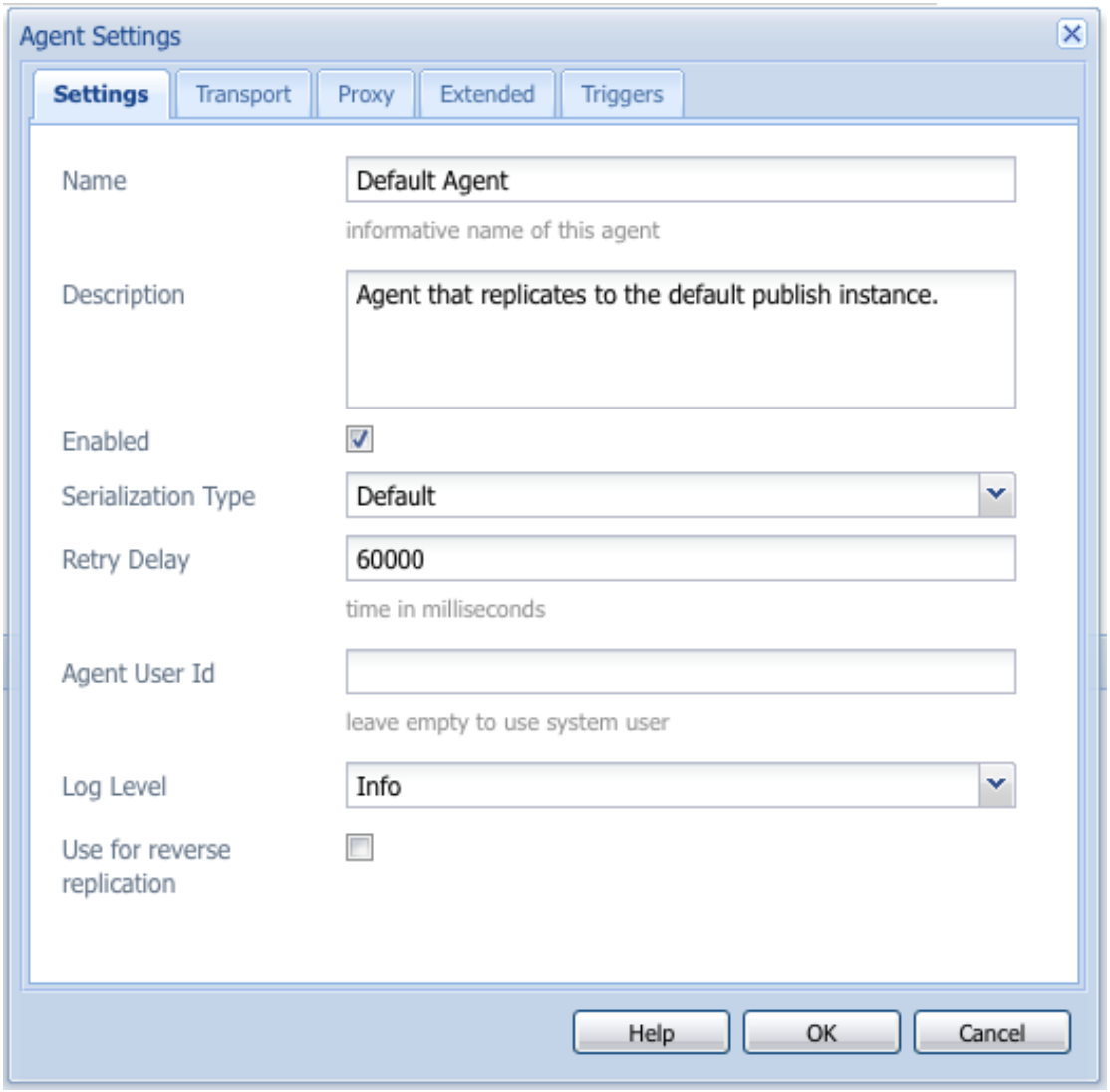
5.1 How to configure your Replication Agents

5.1.1 Configuring your Replication Agents from wcm/siteadmin

From siteadmin in the author environment you can configure replication agents that reside in either the author environment (**`Agents on author`**) or the publish environment (**`Agents on publish`**). The following procedures illustrate the configuration of an agent in the author environment, but can be used for both.

To configure a replication agent from siteadmin:

1. Access the **`Tools`** tab in CQ.
2. Click **`Replication`** (left pane to open the folder).
3. Double-click **`Agents on author`** (either the left or the right pane).
4. Click the appropriate agent name (which is a link) to show detailed information on that agent.
5. Click **`Edit`** to open the configuration dialog:



6. The values provided should be sufficient for a default installation. If you make changes then click **OK** to save them (see the section called "Replication Agents - Configuration Parameters" for more details of the individual parameters).

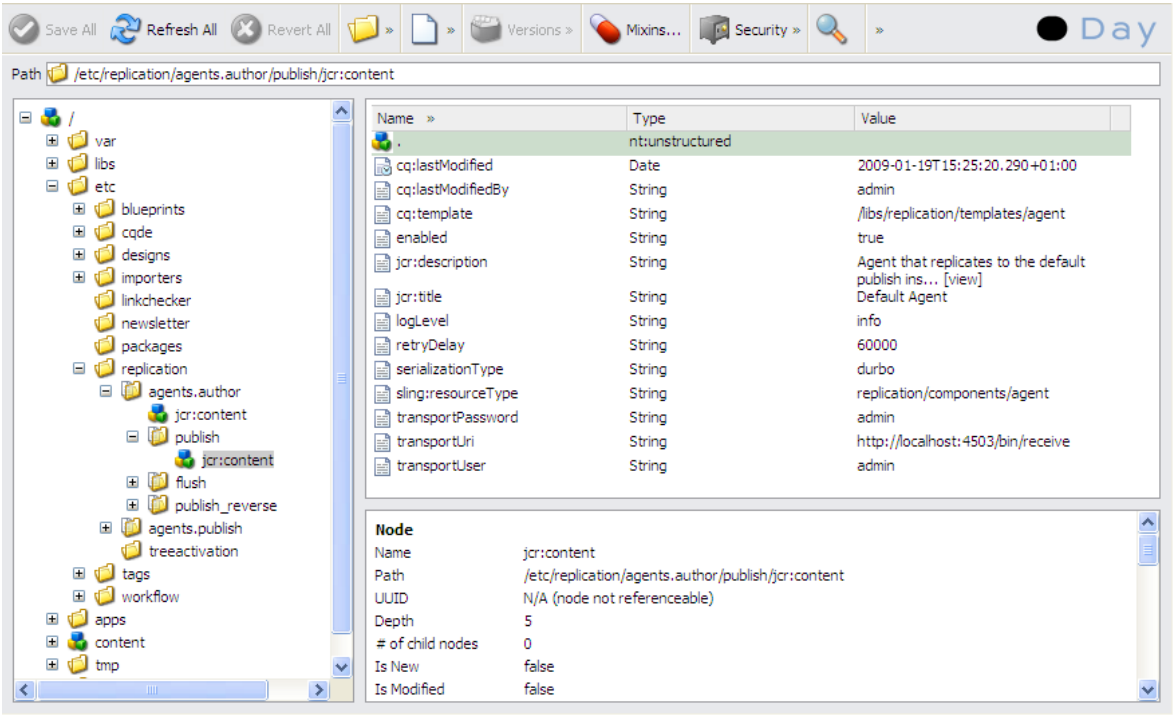
5.1.2 Configuring your Replication Agents from the CRX Explorer

Various parameters of your replication agents can be configured using the CRX Explorer.

If you navigate to /etc/replication you can see the following three nodes:

- agents.author
- agents.publish
- treeactivation

The two agents hold configuration information about the appropriate environment, and are only active when that environment is running. For example, `agents.publish` will only be used in the publish environment. The following screenshot shows the publish agent in the author environment, as included with CQ WCM:



5.1.3 Configuring Reverse Replication

Reverse replication is used to get user content generated on a publish instance back to an author instance. This is commonly used for moderated forums, blogs, surveys and registration forms, amongst others.

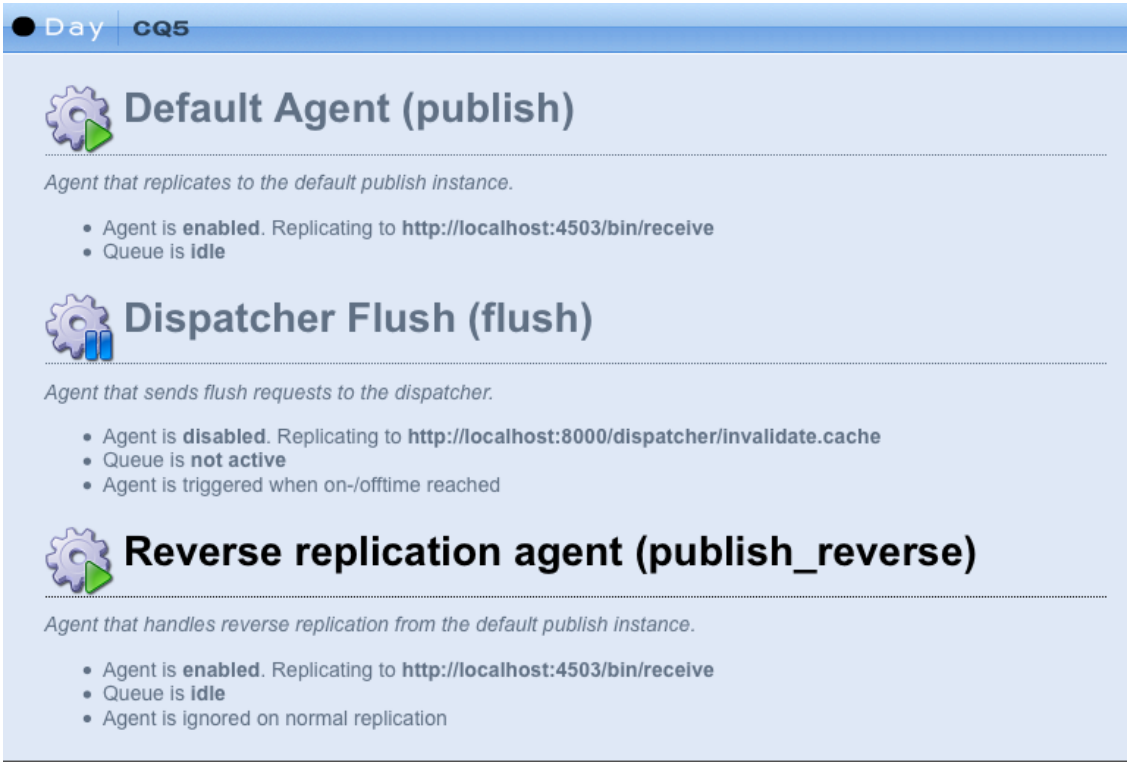
For security reasons, most network topologies do not allow connections *from* the “Demilitarized Zone” (a subnetwork that exposes the external services to an untrusted network such as the Internet).

As the publish environment is usually in the DMZ, to get content back to the author environment the connection must be initiated from the author instance. This is done with:

- an *outbox* in the publish environment where the content is placed.
- an agent (publish) in the author environment which periodically polls the outbox for new content.

To do this you need:

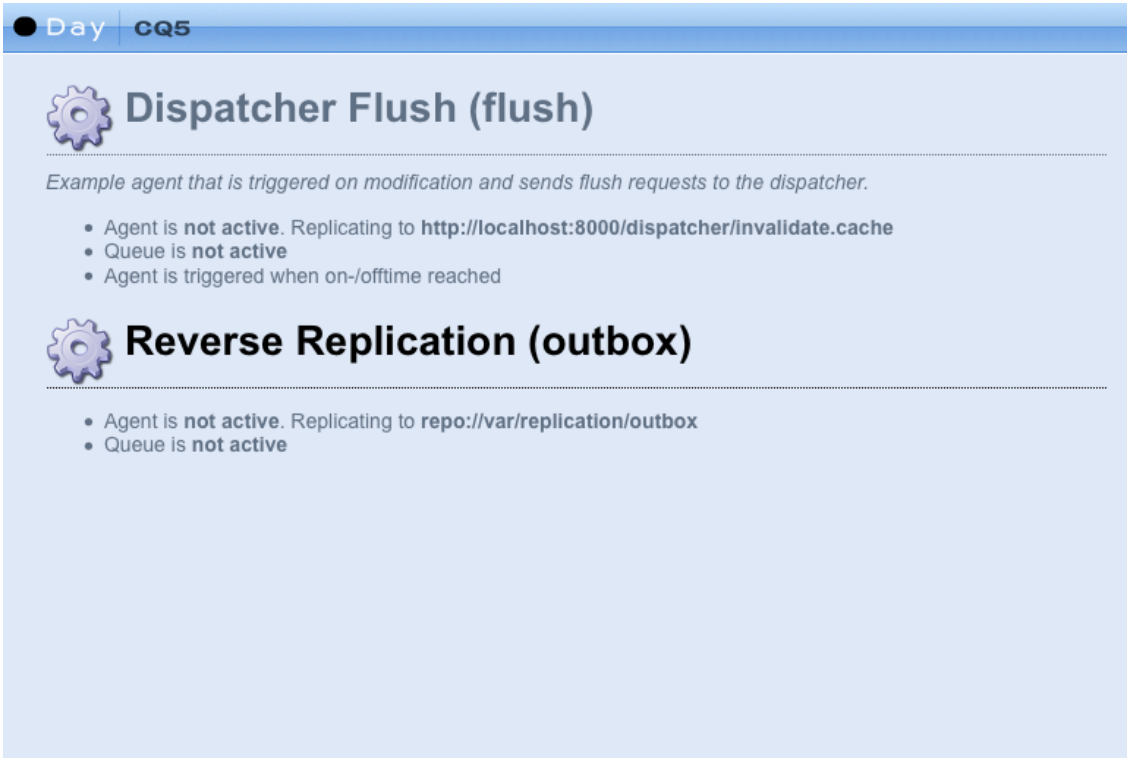
A reverse replication agent in the author environment
: This acts as the active component to collect information from the outbox in the publish environment:



If you want to use reverse replication then ensure that this agent is activated.

A reverse replication agent in the publish environment (an outbox)

This is the passive element as it acts as an “outbox.” User input is placed here, from where it is collected by the agent in the author environment.



5.1.4 Configuring Replication for Multiple Publish Instances

Upon installation a default agent is already configured for replication to a publish instance running on port 4503 of the localhost.

To configure replication for an additional publish instance you need to create, and configure, a new replication agent:

1. Log in to the site administration of CQ5 on the author instance.
2. Open the Tools tab; for example, at `http://localhost:4502/libs/wcm/content/misc.html`.
3. Select **`Replication`**, then **`Agents on author`** in the left panel.
4. Select **`New...`**.
5. Set the **`Title`** and **`Name`**, then select **`Replication Agent`**.
6. Click **`Create`** to create the new agent.
7. Double-click the new agent item to open the configuration panel.
8. Click **`Edit`** - the **`Agent Settings`** dialog will open - the **`Serialization Type`** is already defined as *`Default`*, this must remain so.
    a. In the **`Settings`** tab:
        i. Activate **`Enabled`**.
        ii. Enter a **`Description`**.
        iii. Set the **`Retry Delay`** to *`60000`*.
        iv. Leave the **`Serialization Type`** as *`Default`*.
    b. In the **`Transport`** tab:
        - Enter the required URI for the new publish instance; for example, `http://localhost:4504/bin/receive`.

    You can configure other parameters as required.
9. Click **`OK`** to save the settings.

Tip

You can then test operation by updating, then publishing, a page in the author environment.

The updates will appear on all publish instances that have been configured as above.

If you encounter any problems, you can check the logs on the author instance. Depending on the level of detail required you can also set the **`Log Level`** to *`Debug`*. using the **`Agent Settings`** dialog as above.

5.1.5 Configuring a Dispatcher Flush agent

Default agents are included with the installation. However, certain configuration is still needed and the same applies if you are defining a new agent:

1. Log in to the site administration of CQ5 on the author instance.

2. Open the Tools tab; for example, at `http://localhost:4502/libs/wcm/content/misc.html`.
3. Select **`Replication`**, then **`Agents on publish`** in the left panel.
4. **`Double-click`** on the **`Dispatcher Flush`** item to open the overview.
5. Click Edit - the **`Agent Settings`** dialog will open:
   a. In the **`Settings`** tab:
      i. Activate **`Enabled`**.
      ii. Enter a **`Description`**.
      iii. Leave the **`Serialization Type`** as *`Dispatcher Flush`*, or set it as such if creating a new agent.
   b. In the **`Transport`** tab:
      - Enter the required URI for the new publish instance; for example, `http://localhost:80/dispatcher/invalidate.cache`.

   You can configure other parameters as required.
6. Click **`OK`** to save the changes.
7. Return to the **`Tools`** tab, from here you can **`Activate`** the **`Dispatcher Flush`** agent (**`Agents on publish`**).

Note

The **`Dispatcher Flush`** replication agent is not active on author. You can access the same page in the publish environment by using the equivalent URI; for example, `http://localhost:4503/etc/replication/agents.publish/flush.html`.

5.2 Configuring CQ WCM using the repository

A sub-set of configuration nodes are available in the repository. This ensures that copying, or replicating, repository contents recreates identical configurations. To update the configuration data you update the node properties.

These configuration nodes mirror the OSGi configurations, and form a user interface to them. If you modify the configuration data in the repository the changes are applied to the relevant OSGi configuration as if the changes had been made using the Felix Web Management Console directly, with the appropriate validation and consistency checks.

The following list shows the configurations available in the repository. To update these you **must copy** the node from `libs/` to `apps/`, then make any changes on the version in `apps/`:

Warning

You must not change anything in the `libs/` path.

- Mail Service:

  `libs/cq/config/com.day.cq.mailer.DefaultMailService`

  See the section called “Mail Service”.

- Author Anonymous Membership:

`libs/security/config.author/com.day.cq.security.AnonymousMembership`

See the section called “CQ Anonymous Membership set-up”.

- Publish Anonymous Membership:

  `libs/security/config.publish/com.day.cq.security.AnonymousMembership`

  See the section called “CQ Anonymous Membership set-up”.

- Author WCM Request Filter:

  `libs/wcm/config.author/com.day.cq.wcm.core.WCMRequestFilter`

  See the section called “CQ WCM Filter”.

- Publish WCM Request Filter:

  `libs/wcm/config.publish/com.day.cq.wcm.core.WCMRequestFilter`

  See the section called “CQ WCM Filter”.

- Language Manager:

  `libs/wcm/config/com.day.cq.wcm.core.LanguageManager`

  See the section called “CQ WCM Language Manager”.

- Page Statistics:

  `libs/wcm/config.publish/com.day.cq.wcm.core.stats.PageViewStatistics`

  Holds the property `pageviewstatistics.trackingurl` which can be set to the tracking URL; for example `http://localhost:4502/libs/wcm/stats/tracker`.

- Replication Agents

  `/etc/replication`

  Holds the various replication agents (see the section called “How to configure your Replication Agents”).

5.3 Configuring CQ WCM with the Apache Felix Web Management Console

5.3.1 Using the Apache Felix Web Management Console

Apache Felix “is a community effort to implement the OSGi R4 Service Platform, which includes the OSGi framework and standard services”.

OSGi is a fundamental element in the Technology stack of CQ5, and is used to control bundles and their configuration.

The Apache Felix Web Management Console offers a selection of tabs for controlling the OSGi bundles:

- Bundles: used for installing bundles
- Components: used for controlling the status of components required for CQ5

- Configuration: used for configuring the OSGi bundles, and is therefore the underlying mechanism for configuring CQ system parameters

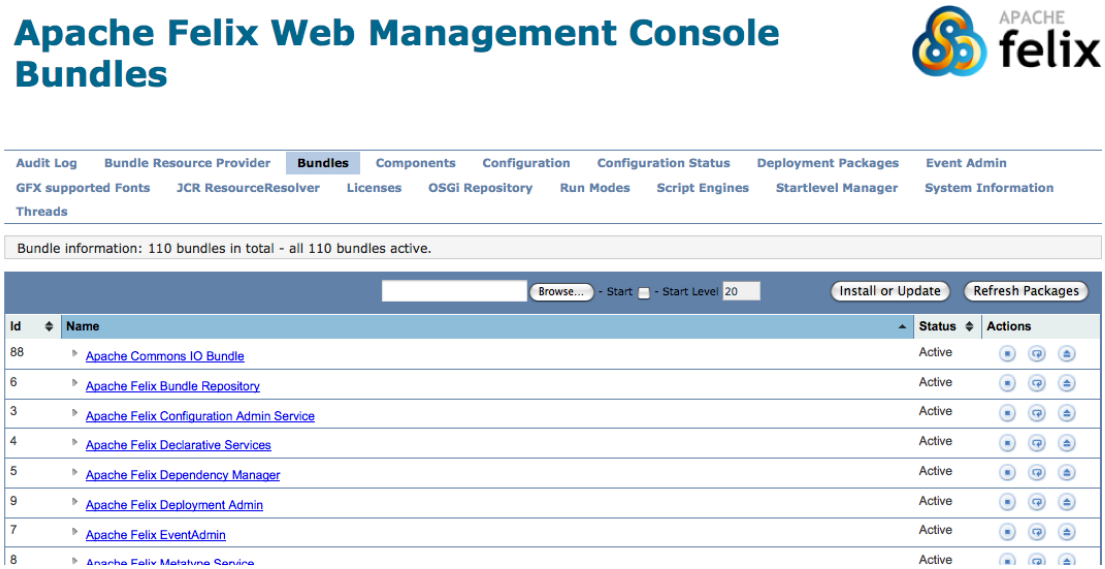
Important

Any changes made are *immediately* applied to the running system. No restart is required.

It can be accessed from `../system/console/components`; for example `http://localhost:4502/system/console/components`.

5.3.1.1 Bundles

The **`Bundles`** tab is the mechanism for installing the OSGi bundles required for CQ5.



Using this tab you can:

Install or Update
: You can **`Browse`** to find the file containing your bundle and specify whether it should **`Start`** immediately and at which **`Start Level`**.

Refresh Packages
: Refreshes the list displayed.

Start
: Starts a bundle according to the start level specified.

Stop
: Stops the bundle.

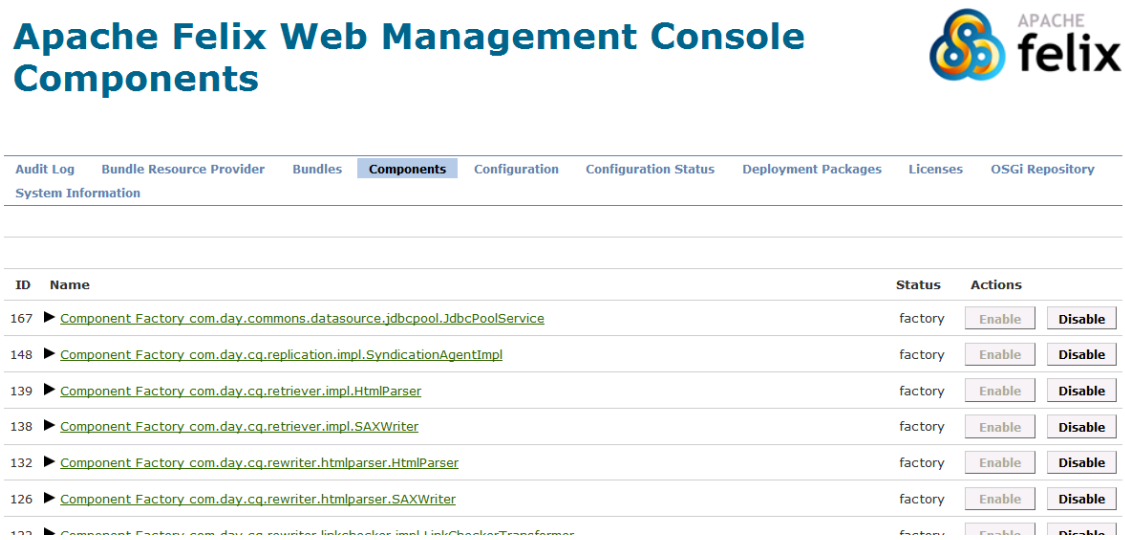
Uninstall
: Uninstalls the bundle from the system.

see the status
: The list specifies the current status of the bundle; clicking on the name of a specific bundle with show further information.

5.3.1.2 Components

The **`Components`** tab allow you to Enable and/or Disable the various components.

Clicking on the name of a particular component will also display further information on its status.

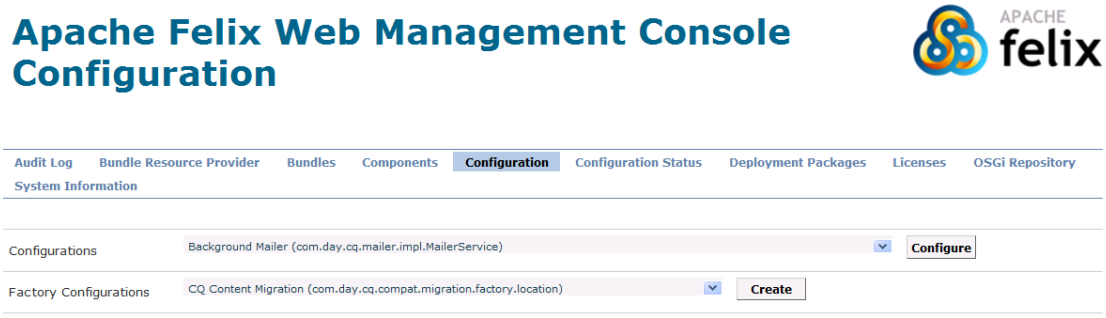


Important

Enabling, or disabling, a component will only apply until CRX is restarted. The start state is defined within the component descriptor, which is generated during development and stored in the bundle at bundle creation time.

5.3.1.3 Configuration

The **`Configuration`** tab is used for configuring the OSGi bundles, and is therefore the underlying mechanism for configuring CQ5 system parameters.



There are two types of configurations available from the drop down lists on this screen:

Configurations
: Allows you to update the existing configurations. These have a Persistent Identity (PID) and can be either:

- standard and integral to CQ5; these are required, if deleted the values return to the default settings.

- instances created from Factory Configurations; these instances are created by the user, deletion removes the instance.

Factory Configurations
: Allows you to create an instance of the required functionality object.

  This will be allocated a Persistent Identity and then listed in the Configurations drop down list.

Selecting any entry from the lists will display the parameters related to that configuration.

You can then:

Save
: Save the changes made.

  For a Factory Configuration this will create a new instance with a Persistent Identity. The new instance will then be listed under Configurations.

Reset
: Reset the parameters shown on screen to those saved last.

Delete
: Delete the current configuration. If standard, the parameters are returned to the default settings. If created from a Factory Configuration, then the specific instance is deleted.

5.3.2 OSGi Configuration Settings

5.3.2.1 CRX Configurations

This section deals with the configuration details of specific bundles and/or instances of bundles required for CRX.

5.3.2.1.1 Apache Felix OSGI Management Console

Configuration of the OSGi Management Console.

Table 5.1. OSGi Management Console

Parameter	Comments
User Name	The name of the user allowed to access the OSGi Management Console. This must match a valid account registered in CRX. To disable authentication clear this value. Default: `admin`
Realm	The name of the HTTP Authentication Realm. Default: `OSGi Management Console`
Root URI	The root path to the OSGi Management Console. This is used when starting the console; for example, using the default: `http://localhost:4502/system/console`. Default: `/system/console`
Default Page	The name of the default configuration page when invoking the OSGi Management console. The console has several pages, or tabs. This value specifies which will be shown when the console is started.

Parameter	Comments
	Default: `bundles`
Password	The password for the user allowed to access the OSGi Management Console. This must match the password as registered in CRX. Upon initial installation you should change the password of the account used to access the OSGi Management Console. After that it should be changed in compliance with security guidelines at the customer site. Default: `admin`

5.3.2.1.2 Apache Sling GET Servlet

These settings are for the use of Day Management AG.

5.3.2.1.3 Apache Sling JCR Install

These settings are for the use of Day Management AG.

5.3.2.1.4 Apache Sling JCR ResourceBundle Provider

The ResourceBundleProvider service loads the messages from the repository. Here you may configure how the provider accesses the repository.

If the user name field is left empty, the provider accesses the repository anonymously. Otherwise the user name and password specified are used to access the repository.

Failing to access the repository, effectively disables the provider.

Table 5.2. JCR ResourceBundle Provider

Parameter	Comments
Password	The password required for the specified user name. Must be valid for accessing CRX. This field is only used if the user name field is not empty.
User Name	The user name to be used for accessing the repository, which must be a valid CRX user. If this field is empty, the provider accesses the repository as the anonymous user.
Default Locale	If no locale can be resolved, then this will be used as the default. The value specified must be in a form acceptable to the java.util.Locale class.

5.3.2.1.5 Apache Sling Java Script Handler

Configuration of the Java Script Handler.

Table 5.3. OSGi Management Console

Parameter	Comments
Source Encoding	The encoding to be used to read the source files.

Parameter	Comments
	Default: `UTF-8` Note: This should only be changed in very specific circumstances.
Source VM	Java Specification to be used to read the source files. Default: `1.5`
Target VM	Java Specification to be used to generate the compiled output. Default: `1.5`
Development Mode	Defines whether scripting is to be used in development mode (will result in a check for modification on every access). Allowed Values: `true` or `false`. Default: `true`
Compilation Location	Defines the directory to be used when compiling Java files. Default: `/var/classes`
Modification Check Interval	Specifies the time interval between checks for modification on a given Java file. Setting this to `-1` will cause the source to be checked on every access. Default: `-1`
Generate Debug Info	Defines whether the class file should be compiled with debugging information. Allowed Values: `true` or `false`. Default: `true`

5.3.2.1.6 Apache Sling Main Servlet

These settings are for the use of Day Management AG.

5.3.2.1.7 Apache Sling POST Servlet

These settings are for the use of Day Management AG.

5.3.2.1.8 Apache Sling Repository Class Loader Factory

These settings are for the use of Day Management AG.

5.3.2.1.9 Apache Sling Request Authenticator

The Request Authenticator extracts user authentication details from the request. It does this in conjunction with authentication handlers which are registered as separate services; for example, the HTTP Authorization header handler.

Table 5.4. Request Authenticator

Parameter	Comments
Impersonation Cookie	The name of the HTTP Cookie used to set the user to be impersonated.

Parameter	Comments
	This cookie will always be a session cookie. Default: `sling.sudo`
Impersonation Parameter	The name of the request parameter initiating impersonation. Setting the specified request parameter to a user id will: • result in an impersonated session being used (instead of the authenticated session) • set a session cookie of the name defined in the Impersonation Cookie setting (see above). Default: `sudo`
Allow Anonymous Access	Specifies whether access will default to anonymous if no credentials are provided. This is primarily of interest for the publish instance. The default value is "`Enabled`" to allow access without credentials. When disabled, access to the repository is only allowed if valid credentials are provided. Default: `Enabled`

5.3.2.1.10 Apache Sling Request Logger

These settings are for the use of Day Management AG.

5.3.2.1.11 Apache Sling Resource Resolver

Specifies the information required for the Resource Resolver to perform mappings between the request URL and resource paths.

Table 5.5. Resource Resolver

Parameter	Comments
Allow Direct Mapping	Specifies whether to add a direct URL mapping to the front of the mapping list. Default: `True`
Virtual URLs	A list of virtual URLs and their mappings to real URLs. Mappings are applied on the complete request URL only. The format used is `<externalURL>-<internalURL>`. Default: `/-/`
URL Mappings	A list of mappings to apply to URLs. Incoming mappings are applied to request URLs, mapping them to Content paths.

Parameter	Comments
	Outgoing mappings are applied to map Content paths to URLs used on subsequent requests. The format used is `<externalURLPrefix><op><internalURLPrefix>`, where `<op>` is: • ">" for incoming mappings • "<" for outgoing mappings • "-" for mappings applied in both directions Mappings are applied in configuration order by comparing and replacing URL prefixes. Defaults: "`/-/`", "`/content/-/`", "`/system/docroot/-/`"
Resource Search Path	The list of absolute path prefixes which can be applied (to find resources) when the path has been specified with a relative path. Defaults: "`/apps`", "`/libs`" Note: If an empty path is specified a single entry path of "/" is assumed.
Resource Regexps	A list of regular expression substitutions that will be applied to a URL before the resource resolving. The purpose is to replace any characters which should not occur in the path. The format for definition is <regexp>|<replacement string> ; i.e. the regexp is separated from the replacement string by the | (pipe) character. Standard regular expressions are used. The default `/_([^/]+?)_|/$1:` would replace `/_a_` with `/a:` . For each regexp you need to specify a regexp which will perform the reverse substitution - see Map Regexps. Default: `/_([^/]+?)_|/$1:` *Note: This existing default should not be removed.*
Map Regexps	A list of regular expression substitutions that will be applied to a resource before mapping to the URL. The purpose is to replace any characters which should not occur in the path. See Resource Regexps above for further details. Default: `/([^/]+?):([^/]+)|/_$1_$2` *Note: This existing default should not be removed.*

5.3.2.1.12 Apache Sling Authorization Header Authenticator

These settings are for the use of Day Management AG.

5.3.2.1.13 CRX Sling Client Repository

See the Factory Configuration of the section called “CRX Sling Client Repository”.

5.3.2.1.14 JSP Script Handler

The JSP Script Handler supports development of JSP scripts used to render response content on behalf of ScriptComponents.

Internally the Jasper 5.5.20 JSP Engine is used, together with the Eclipse Java Compiler to compile generated Java code into Java class files. Some settings of Jasper may be configured as specified below.

Note

JSP scripts are expected in the JCR repository. Generated Java source and class files will be written to the JCR repository below the configured Compilation Location.

Table 5.6. JSP Script Handler

Parameter	Comments
Compilation Check Interval	If *`development`* is false and *`reloading`* is true, then background compiles are enabled. *`checkInterval`* is the time in seconds between checks to see if a JSP page needs to be recompiled. i.e. 300 indicates that a check will be made every 300 seconds. Default: `300`
Generate Debug Info	Indicates whether the class file will be compiled with debugging information. Use of debugging information can impact the performance. Default: `True`
Development Mode	Specifies whether Jasper is used in development mode (will check for JSP modification on every access). These checks may impact performance. Default: `True`
Tag Pooling	Determines whether tag handler pooling is enabled. Default: `True`
Plugin Class-ID	Indicates the class-id value to be sent to Internet Explorer when using `<jsp:plugin>` tags. Default: `clsid:8AD9C840-044E-11D1-B3E9-00805F499D93`
Char Array Strings	Specifies whether text strings will be generated as char arrays. Note: This can improve performance in some cases. Default: `False`
Keep Generated Java	Should we keep the generated Java source code for each page instead of deleting it? true or false.

Parameter	Comments
	Default: `True`
Mapped Content	Specifies whether static content (one print statement per input line) will be generated. This output can then be used to ease debugging. Default: `True`
Modification Check Interval	Checks for the modification of a given JSP file (and all its dependent files) will be performed only once every interval. This parameter specifies the interval in number of seconds. i.e. 4 indicates that the check will occur once every 4 seconds. Setting this to 0 will cause the JSP to be checked on every access. Default: 4
Check Modified JSPs	Specifies whether Jasper will check for modified JSPs. Default: `False`
Compilation Location	Defines the scratch directory to be used when compiling JSP pages. Default: The work directory for the current web application. i.e. `/var/classes`
Trim Spaces	Specifies whether white spaces in the template text (between actions and/or directives) will be trimmed. Default: `False`
Display Source Fragments	Specifies whether a source fragment should be included in exception messages, for display to the developer.

5.3.2.1.15 Simple WebDAV Servlet

These settings are for the use of Day Management AG.

5.3.2.1.16 Sling Logging Configuration

Global configuration setup for the Sling OSGi LogService implementation.

Table 5.7. Sling Logging Configuration

Parameter	Comments
Log Level	The log level setting used by the root logger. Possible values are: Trace, Debug, Information, Warnings, Error. Default: `Information`
Log File	The name of the log file, with directory path. If this is empty, the log messages are output to standard output (the console). If this path is relative it is resolved to below `${sling.home}`.
Number of Log Files	This specifies the number of versions of a log files to be kept. When the size of the log file reaches the configured maximum (see Maximum Log File Size), the existing log file is moved and a new version created.

Parameter	Comments
	Default: 5
Maximum Log File Size	The maximum size of a log file. If the specified size is reached, then the log file is moved and a new version is created. This size may be specified with size indicators KB, MB and GB (case insensitive). Default: 10mb
Message Pattern	The Message Pattern is used to format the log messages. This is a java.util.MessageFormat pattern supporting up to 6 arguments: 0 - The timestamp of type java.util.Date. 1 - The log marker. 2 - The name of the current thread. 3 - The name of the logger. 4 - The debug level. 5 - The actual debug message. If the log call includes a "Throwable," the stacktrace is appended to the message. Default: {0,date,dd.MM.yyyy HH:mm:ss.SSS} *{4}* [{2}] {3} {5}

5.3.2.1.17 Sling Servlet/Script Resolver and Error Handler

These settings are for the use of Day Management AG.

5.3.2.1.18 Sling default resource type provider

These settings are for the use of Day Management AG.

5.3.2.1.19 org apache sling scripting javascript internal RhinoJavaScriptEngineFactory name

These settings are for the use of Day Management AG.

5.3.2.2 CRX Factory Configurations

This section deals with the Factory Configuration details of bundles required for CRX. Using these, you can configure and create specific instances of such bundles.

5.3.2.2.1 Apache Sling Customizable Request Data Logger

These settings are for the use of Day Management AG.

5.3.2.2.2 CRX Sling Client Repository

Implements a sling repository that accesses an underlying CRX.

Table 5.8. CRX Sling Client Repository

Parameter	Comments
JNDI Provider URL	Specifies the DNS host name and service port number of the machine that is running the JNDI service Default: `http://jcr.day.com`
JNDI Initial Factory	Defines the initial context factory. Default: `com.day.util.jndi.provider.MemoryInitialContextFactory`
Repository Name	Name of the CRX repository to be accessed. Default: `crx`
Default Workspace	Default workspace to be used within the CRX repository.
Anonymous UserId	User name of the anonymous account. Default: `anonymous`
Anonymous Password	Password for the anonymous account. Default: `anonymous`
Admin UserId	User name of the administration account to be used to access the repository. This must match a valid account on the repository. Default: `admin`
Admin Password	Password for the administration account to be used for accessing the repository. This password must match the password of the account on the repository. Default: `admin`
Active Poll Interval	Interval between the polling checks to confirm that the repository is still active. Default: `10`
Inactive Poll Interval	If it has been determined that the repository is not active, then this interval is used for polling until it is active again. Default: `10`
Pool max wait	Specifies the time to wait for a new session to be allocated if the session pool is full. If no session can be allocated after the specified time, the session request is aborted and a log entry made.

Parameter	Comments
	Default: `1`
Pool max active	Maximum number of sessions which can be concurrently active. Default: `-1` *Note: -1 represents infinity.*
Pool max idle	Maximum number of currently idle sessions to keep in the pool. Default: `10`

5.3.2.2.3 Sling Logging Logger Configuration

The Logging Logger controls which log messages are written, and into which file. A Logger works together with a Logging Writer.

You can create and configure a new instance of a Sling Logging Logger, with levels, patterns and destinations. Creating a new instance can be used to set an additional logging level.

Note

The "`Log File`" specified for the Logger must be the same as for the Logging Writer.

Table 5.9. Sling Logging Logger Configuration

Parameter	Comments
Log Level	The log level setting used by the root logger. Possible values are: Trace, Debug, Information, Warnings, Error. Default: `Information`
Log File	The name of the log file, including directory path. If this is empty, logging messages are output to standard output (the console). If this path is relative, it is resolved below `${sling.home}`. Default: `logs/error.log`
Message Pattern	The Message Pattern is used to format the log messages. This is a `java.util.MessageFormat` pattern supporting up to 6 arguments: `0` - The timestamp of type `java.util.Date`. `1` - The log marker. `2` - The name of the current thread. `3` - The name of the logger. `4` - The debug level. `5` - The actual debug message.

Parameter	Comments
	If the log call includes a "`Throwable`", the stacktrace is appended to the message. Default: `{0,date,dd.MM.yyyy HH:mm:ss.SSS} *{4}* [{2}] {3} {5}`
Categories	A list of categories applicable for the new instance of the logger. Each category applies to any child category unless configured otherwise; e.g. a category of `org.apache.sling` applies to category `org.apache.sling.commons` unless there is a separate configuration for `org.apache.sling.commons`.

5.3.2.2.4 Sling Logging Writer Configuration

Configure a Logger Writer for Sling Logging. This takes the messages from the Sling Logging Logger, writes them to file, and controls the versioning of the log files.

Note

The "`Log File`" specified for the Logging Writer must be the same as for the Logging Logger.

Table 5.10. Sling Logging Writer Configuration

Parameter	Comments
Log File	The name of the log file, including directory path. If this is empty, logging messages are output to standard output (the console). If this path is relative, it is resolved below `${sling.home}`. Default: `logs/error.log`
Number of Log Files	This specifies the number of versions of a log files to be kept. When the size of the log file reaches the configured maximum (see Maximum Log File Size), the existing log file is moved and a new version created. Default: `5`
Maximum Log File Size	The maximum size of a log file. If the specified size is reached, then the log file is moved and a new version is created. This size may be specified with size indicators KB, MB and GB (case insensitive). Default: `10mb`

5.3.2.3 CQ Configurations

This section deals with the configuration details of specific bundles and/or instances of bundles required for CQ.

5.3.2.3.1 Apache Sling Distribution Event Handler

These settings are for the use of Day Management AG.

5.3.2.3.2 Apache Sling Event Thread Pool

These settings are for the use of Day Management AG.

5.3.2.3.3 Apache Sling Job Event Handler

These settings are for the use of Day Management AG.

5.3.2.3.4 Background Mailer

These settings are for the use of Day Management AG.

5.3.2.3.5 CQ Anonymous Membership set-up

These settings are used to configure group membership of the anonymous user at start-up.

Table 5.11. CQ Anonymous Membership set-up

Parameter	Comments
Groups Anonymous has to be member of	At start-up, adds anonymous to these groups. Default: `uploader`
Groups Anonymous must not be member of	At start-up, removes anonymous from these groups. Default: `administrators`, `surfer`

5.3.2.3.6 CQ Antispam

These settings are for the use of Day Management AG.

5.3.2.3.7 CQ Default Theme Resolver

These settings are used to resolve the theme based on the handle-mapping configuration.

Table 5.12. CQ Default Theme Resolver

Parameter	Comments
Handle Mapping	Specifies the pattern to be used for mapping from handles to themes. @ is used as the delimiter. For example, with `/content/playground*@/apps/day/themes/blue`, all handles under `/content/playground*` will be mapped to the theme under `/apps/day/themes/blue`. Default: `/content/playground*@/apps/day/themes/blue`, `/content/*@/etc/themes/default`
Default Theme	Path of the default theme, to be used if none can be applied from the mappings above. Default: `/etc/designs/default`

5.3.2.3.8 CQ HTML Library Manager

The HTML Library Manager is primarily of interest to developers, as it provides methods for including js and/or css files stored in the repository, also resolving the categories and dependencies.

Table 5.13. CQ HTML Library Manager

Parameter	Comments
Minify	Compress js/css files using the YUI compressor (for whitespace removal etc.) Default: `False`
Debug	Serve js/css files individually (instead of concatenating them) for better client-side debugging. Default: `False`
Gzip	Use gzip compression for delivering js/css files. Default: `True`
Timing	Enable JS load timing. This requires the function `CQ.Timing.stamp()` (from `/libs/widgets/cq-widgets.js/files/Timing.js`). Default: `Enabled`
Firebug Lite Path	Path in the workspace where the `firebug-lite.js` file is located. The script should make sure it does not activate if the real firebug is present. Default: `/libs/widgets/cq-widgets.js/files/endorsed/firebug/firebug-lite.js`
Debug Console	Enables a client-side debug console. This will include the `firebug-lite.js` file and the *`Debug Init Script`* snippet. Default: `Disabled`
Debug Init Script	A Javascript snippet for enabling debugging and/or show the console. Default: `window.CQ_initial_log_level='ALL';`

5.3.2.3.9 CQ JCR Tag Manager Factory

These settings are for the use of Day Management AG.

5.3.2.3.10 CQ Link Checker Service

These settings are for the use of Day Management AG.

5.3.2.3.11 CQ Link Checker Task

Performs link checking for both internal and external links, though the configuration only affects external links.

Table 5.14. CQ Link Checker Task

Parameter	Comments
Scheduler Period	scheduler.period.description

Parameter	Comments
Good Link Test Interval	The interval in hours to check good links.
Bad Link Test Interval	The interval in hours to check bad links.
link_unused_interval.name	link_unused_interval.description
Proxy Host	Optional proxy for accessing/checking external links.
Proxy Port	Optional proxy port for accessing/checking external links.
Proxy User	Optional user for proxy authentication.
Proxy Password	Optional password for proxy authentication.
connection.timeout.name	connection.timeout.description
NTLM Proxy Host	When using NTLM proxy authentication,set this property to the name of the host where CQ5 is running.
NTLM Proxy Domain	The domain name required for NTLM authentication. When using NTLM proxy authentication,set this property to the name of the NT domain that the specified user belongs to.

5.3.2.3.12 CQ Page Processor

These settings are for the use of Day Management AG.

5.3.2.3.13 CQ Preference Locale Resolver

The Locale Resolver reads the user's preferred language from their account properties. This is then used to determine the locale for requests.

If no user language has been defined, then the locale defaults to the language configured for this Service - see below.

Table 5.15. CQ Preference Locale Resolver

Parameter	Comments
Request default language	Specifies the language to be selected for the request if the user does not have a language preference set, or the preference cannot be resolved. Default: `en`
Preference Name	Specifies where the user's language preference is held. Default: `platform/language`

5.3.2.3.14 CQ Profile Manager

These settings are for the use of Day Management AG.

5.3.2.3.15 CQ Replication Event Listener

These settings are for the use of Day Management AG.

5.3.2.3.16 CQ Repository Theme Registration

Resolves and maintains the themes stored inside a Repository.

Table 5.16. CQ Repository Theme Registration

Parameter	Comments
Theme Root	Path where the themes are stored.

Parameter	Comments
	Default: `/etc/designs`
Default Theme Name	Name of the default theme. Default: `default`
Root Nodetype Name	Name of the NodeTypes to be used for folders. Default: `cq:Page`

5.3.2.3.17 CQ Repository change listener

These settings are for the use of Day Management AG.

5.3.2.3.18 CQ Reverse Replicator

These settings are for the use of Day Management AG.

5.3.2.3.19 CQ Root Mapping

Maps any requests to "/" to an entry page.

Table 5.17. CQ Root Mapping

Parameter	Comments
Target Path	Redirects requests to "/" to the specified path. Defaults: author: `/libs/wcm/content/welcome.html` publish: `/content`

5.3.2.3.20 CQ Social Collaboration Blog Processor

These settings are for the use of Day Management AG.

5.3.2.3.21 CQ Social Collaboration Blog Search Engine Ping

These settings are for the use of Day Management AG.

5.3.2.3.22 CQ Social Collaboration Comment Processor

These settings are for the use of Day Management AG.

5.3.2.3.23 CQ Social Collaboration Rating Processor

These settings are for the use of Day Management AG.

5.3.2.3.24 CQ User Manager Factory

These settings are for the use of Day Management AG.

5.3.2.3.25 CQ WCM Command Servlet

These settings are for the use of Day Management AG.

5.3.2.3.26 CQ WCM Debug Filter

Controls the WCM debug mode, which can be used when developing components.

Table 5.18. CQ WCM Debug Filter

Parameter	Comments
Enabled	Controls whether the WCM debug mode is enabled. Default: `Enabled`

5.3.2.3.27 CQ WCM Filter

This controls the WCM mode for component and edit context; which injects javascript for editing. This enables an author to edit or preview the component settings on a page.

Table 5.19. CQ WCM Filter

Parameter	Comments
WCM Mode	This controls the WCM mode for component and edit context; injects javascript for editing. You can select between Edit, Disabled and Preview. • `Edit` is usual for an author environment - allowing the user to edit any component settings available. • `Disabled` is required for a publish environment as no edit should be allowed. • A third possibility is `Preview` which is used when editing - to preview any changes. However, this value should not be defined here (where it would effectively operate as disable) but be set interactively by the user - as a toggle-switch to `Edit`. Default: `Enabled`

5.3.2.3.28 CQ WCM Find Replace Servlet

These settings are for the use of Day Management AG.

5.3.2.3.29 CQ WCM Language Manager

These settings are for the use of Day Management AG.

5.3.2.3.30 CQ WCM Link Checker Configuration

These settings are for the use of Day Management AG.

5.3.2.3.31 CQ WCM Newsletter

These settings are for the use of Day Management AG.

5.3.2.3.32 CQ WCM Rollout Manager

These settings are for the use of Day Management AG.

5.3.2.3.33 CQ Widget Extension Provider

These settings are for the use of Day Management AG.

5.3.2.3.34 CQDE Servlet

These settings are for the use of Day Management AG.

5.3.2.3.35 Communique 5 Polling Importer

These settings are for the use of Day Management AG.

5.3.2.3.36 Communique 5 Workflow Service

This manages:

- access to workflow sessions
- the lifecycle of the workflow engine.

Table 5.20. Communiqué 5 Workflow Service

Parameter	Comments
Model Path	Specifies the location used for storing models. Default: `/etc/workflow/models`
Instances Path	Specifies the location used for storing instances. Default: `/etc/workflow/instances`
Workspace	Specifies the workspace (in the repository) which will be used for workflows. Default: `crx.default`
Job Retry Count	Specifies how often execution of a job will be attempted before it is abandoned.
Superuser	List of superusers and administrators accounts which have access to the Workflow Engine. Default: `admin`, `administrators`
Workflow Group	Group containing all users that can participate in a workflow. Note: This group *must* be assigned to any user account required to access workflows. Default: `workflow-users`

5.3.2.3.37 ContentBus Name Mapper

These settings are for the use of Day Management AG.

5.3.2.3.38 Day Commons GFX Font Helper

These settings are for the use of Day Management AG.

5.3.2.3.39 HTTP Header Authentication Handler

These settings are for the use of Day Management AG.

5.3.2.3.40 Impersonation Servlet

These settings are for the use of Day Management AG.

5.3.2.3.41 Mail Service

Defines the mail service to be used when sending emails.

Table 5.21. Mail Service

Parameter	Comments
SMTP server host name	The SMTP to be used to send messages. Default: `smtp.somedomain.com`
SMTP server port	The port number used when connecting to the SMTP server. Default: 25
SMTP user	The user name used for SMTP authentication.
SMTP password	The password for the above account.
“From” address	The email address shown in the “From:” field of emails sent by the mail service.
Debug email	If enabled, interactions with the SMTP server are output (in dump format) to the console that runs Sling. Default: `False`

5.3.2.3.42 Query Builder

These settings are for the use of Day Management AG.

5.3.2.3.43 SSO Authentication Handler

This handler implements the authorization steps needed for single sign on systems like siteminder or IIS NTLM.

Table 5.22. Single Sign-on Authentication Handler

Parameter	Comments
Path name	
Header Names	The header names that might contain a user ID.
Cookie Names	The cookie names that might contain a user ID.
Parameter Names	The parameter names that might contain a user ID.
ID Format	The format the user ID is encoded with. Currently only `Basic` and “no format” (leave field blank) are supported.

Parameter	Comments
	Default: `Basic`
HTTP Realm name	Name of the HTTP realm - this name is displayed in the login window seen by the user. Default: `Day Communique 5`

5.3.2.3.44 Simple WebDAV Servlet

These settings are for the use of Day Management AG.

5.3.2.3.45 com day cq workflow impl job TimeoutHandler name

These settings are for the use of Day Management AG.

5.3.2.4 CQ Factory Configurations

This section deals with the Factory Configuration details of bundles required for CQ. Using these, you can configure and create specific instances of such bundles.

5.3.2.4.1 CQ Content Upgrade Tool

These settings are for the use of Day Management AG.

5.3.2.4.2 JDBC Connections Pool

Provides JDBC Datasources for the Datasource Pool Service. This is of interest to the developers when developing functionality to access the database.

Table 5.23. JDBC Connections Pool

Parameter	Comments
Pool size	Maximum number of connections that this pool can open. Default: `10`
Readonly by default?	Specifies whether JDBC connections will be set to readonly mode by default. Default: `Disabled`
Password	Password for the account used for JDBC authentication (see Username).
Datasource name	Name of the data source.
Pool wait (msec)	The maximum wait time, in milliseconds, before throwing an exception if all pooled Connections are in use. Default: `1000`
JDBC driver class	Java class name of the JDBC driver to use Default: `com.somevendor.somedriver.Driver`
JDBC connection URI	URI of the JDBC connection to use. Default: `dbc:somedriver:someDB;param=value`
Username	User name to use for JDBC authentication.

Parameter	Comments
Autocommit by default?	Specifies whether JDBC connections are set to autocommit mode by default.
Validation query	Optional JDBC query used to validate connections.

6 How to Backup your CQ WCM instance

It is good practice to take backups of:

- your software installation - before/after significant changes in configuration
- the content held within the repository - regular

6.1 Backing up your software installation

After installation, or significant changes in the configuration, it is advisable to take a backup of your software installation.

To do this you need to take a backup of your repository then:

- stop CQ WCM
- backup the entire <*cq-installation-dir*> from your file system

Caution

If you are operating a cluster then the “shared” folder might be in a different location and will also need to be backed-up. See the section called “How to Set Up a Cluster in CQ” for information about configuring a cluster.

Caution

If you are operating a third party application server, then additional folders may be in a different location and also need to be backed-up. See the section called “How to install CQ5 with an Application Server” for information about installing application servers.

Caution

Incremental backups should not be used with TarPM as file names can change.

Note

Disk mirroring can also be used as a backup mechanism.

6.2 Backing up your repository

The Backup and Restore section of the CRX documentation covers all issues related to backups of the CRX repository. Both on- and/or offline.

For full details of making an online “Hot” backup see Creating an Online Backup.

7 Configuring LDAP

LDAP authentication is required to authenticate users stored in a (central) LDAP directory such as Active Directory. This helps reduce the effort required to manage user accounts.

LDAP authentication occurs at the repository level, so is handled directly by CRX - not CQ5. Therefore, it is enabled and configured from within CRX; please see the LDAP section within the CRX documentation for further information.

For user management within CQ (including assignment of access rights) see User Administration and Security.

8 Using Clusters

8.1 How to Set Up a Cluster in CQ

A cluster is formed of two, or more, live servers linked together. Therefore, if one node fails, the other nodes are active and accessible for your applications and there is no system interruption. This allows you to recover and re-start failed nodes easily. New nodes can also be added to an existing cluster, allowing for simple extensibility.

Clustering is beneficial for:

Increased availability
: When a server breaks down, or becomes unavailable, the cluster agent relays requests to the servers that are still running. Service continues without interruption.

Increased performance
: Clustering increases system performance and availability even when nodes fail.

  While all servers in the cluster are active, you can use their combined computational power. Therefore, this solution improves performance during normal use. However, if one server breaks down you lose its performance, so the overall application performance may suffer.

The following section describes how to set up a cluster in CQ with two cluster nodes on two separately networked servers.

The master node is called node 1, the slave node is called node 2.

On the node 1 (master):

1. In the file system, create a folder `/node1`.
2. Install CQ under `/node1`. For a complete description of the installation, please refer to the section called “Installing an Author Instance”.
3. In the file system, share the folder `node1/crx-quickstart/repository/shared` so that it can be accessed from node 2.

On the node 2 (slave):

1. In the file system, create a folder `/node2`.
2. Install CQ under `/node2`. For a complete description of the installation, please refer to the section called “Installing an Author Instance”.
3. In the file system, map the folder on node 1 `node1/crx-quickstart/repository/shared` to a drive; `Z:` in our case.
4. In your browser, navigate to `http://localhost:4502/crx` to open the CRX **`Main Console`**.
5. Log in as administrator (`admin`).
6. Click **`Repository Configuration`**.
7. Under **`Tools`**, click **`Cluster`**.
8. Under **`Join Cluster`**, as **`Shared path`**, enter **`Z:\`** and click **`Join`**.

Note

In order to add more nodes to the cluster, repeat the steps on the slave node as many times as needed.

9 Installing and Configuring Additional Tools/Modules

9.1 Dispatcher

See here for information on the Dispatcher.

9.2 Portals

Based on the Java Portlet Standard 2.0 (JSR 286), CQ5 WCM Portal integration lets you display and edit CQ content within a portal server environment.

A web portal is a site that functions as a point of access to information on the web and includes information from diverse sources, such as Communiqué content. A portal page is made up of portlets, which are web applications.

This section describes the portal architecture and features and also includes information on setting up CQ WCM content to display in a portal server environment.

9.2.1 CQ Portlet Architecture

The CQ portlet provided in CQ WCM allows for easy updating, extending, and allowing hot updates whenever possible. The CQ portlet is easy to customize; for example, you can customize the SSO/ Authentication handling by deploying your own authentication service generating the required authentication information for CQ to overwrite the default behavior. The plugins use a defined API that lets you add your own functionality by building the plugin against the API. The plugin can be deployed into the running portlet.

The CQ portlet has the following features (with links to more information, as appropriate):

- **Displaying content in a portlet.** The portlet displays content, which can be either an entire page or part of a page, from CQ WCM. The template that is used in combination with the provided selectors, defines how you see the content. In particular, the portlet can show content from just one page or provide a whole substructure of a site and show the corresponding navigation. The portlet supports different display modes such as the normal view and the maximized view.

- **Configuring the portlet.** You configure the portlet to point directly to the publish CQ5 instance by using a URL that either points directly to the content or to the portlet content map within CQ WCM. If it points to the portlet content map, then the configured URL contains a key for the content and the configuration maps the key to the actual content to be displayed, which allows CQ5 to adjust references, for example if someone moves pages within CQ WCM. .The portlet itself does not know if it points directly to the content or if a configuration map is used. For editing, the base URL to the CQ WCM author instance is configured as well. See Configuring the Portlet.

- **Layout.** The portlet respects portal design/themes. The design (CSS files) are added by the portlet to the portal page, which you can configure. As this is an optional portlet API 2.0 feature, for portlet containers not supporting this feature, you manually add the CSS files to the portal server.

- **Security.** User and group information are shared with the portal. CQ WCM respects user information for personalization and access control. The user information from the portal (current user) is forwarded to CQ WCM. The user is already logged into the portal, so the portal is the driving authentication mechanism. CQ WCM (and therefore CRX) receives enough

information to handle this properly. SSO scenarios are supported through existing and/or custom authenticators. As the portlet can connect to different CQ WCM instances (especially publish and author for display and editing), potentially different user information needs to be properly handled. Because this handling can be very specific to a given environment, a mechanism is provided that allows for plugging diverse authenticators. In addition to this proper user handling, it is also possible to configure the portlet to just use one technical user. See Authentication.

- **Multi-Language.** Content from CQ WCM respects the portal language. The portlet standard specification requires that available languages for a portlet are configured per portlet. The portal forwards the preferred language of the user to the CQ portlet which, in turn, forwards this to the CQ WCM server. The mechanism to provide the correct language uses structural information about the site, that is, the request is mapped to language specific URLs inside the portlet. This mechanism is pluggable to allow for customization.

- **Links.** All links in the portlet must be rewritten properly.

- **Caching.** The portlet can be configured to point to the dispatcher instead of directly pointing to the CQ server. In this scenario, the dispatcher cache is used. The portlet also has its own cache. The implementation of the cache can be replaced by a custom implementation if needed. See Caching.

- **Editing Content.** The CQ portlet provides links to configure the portlet and edit its content. The CQ WCM part of the configuration determines how to handle the content that is shown in the portlet. You edit in the CQ WCM authoring environment. If necessary, you can also access site administration through an additional link in the tree instead of directly editing a page. It might still be necessary to go to the site administration instead of directly editing a page. The editing functionality is only available if the current user has the credentials to edit a page. By default, the CQ5 author instance is queried to detect if the user is an author.

- **Non Functional.** Wherever possible, portlet integration is based on standards (in particular JSR 286), so it is generic for all portal environments that support this standard. Setups with one portal and multiple CQ WCM instances as well as multiple portal instances and one CQ WCM instance are supported.

9.2.2 How to install, configure, and use CQ5 WCM in a portlet

To access content provided by CQ5 WCM, the portal server needs to be fitted with the CQ5 Portal Director Portlet. You do this by installing, configuring, and adding the portlet to the portal page by using the steps provided in this section.

By default, the portlet connects to the publish instance at `localhost:4503` and to the author instance at `localhost:4502`. These values can be changed during deployment of the portlet by changing the according preferences. The portal director can be found in the `opt` directory after CQ quickstart has been started for the first time.

Important

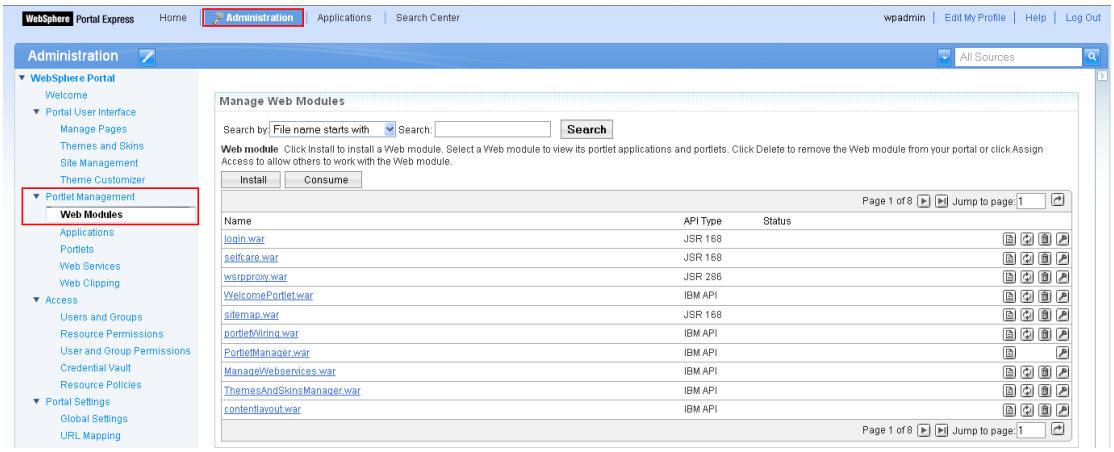
These procedures use the Websphere portal as an example, therefore the procedures and included screenshots vary for other web portals. Although the steps are essentially identical for all web portals, you need to repurpose the steps for your web portal (for example, navigating to the **`Administration`** section of the web portal may differ from the Websphere portal).

9.2.2.1 Installing the portlet

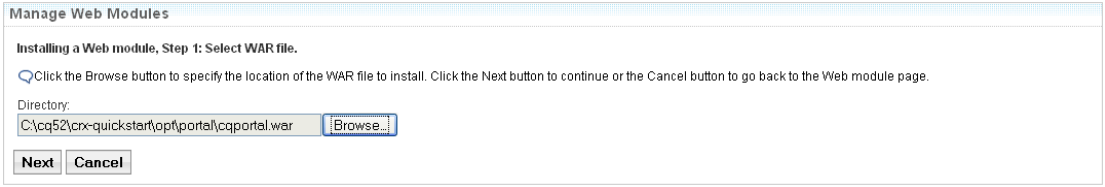
To install the portlet:

1. Log in to the portal with administrator privileges.

2. Navigate to the Portlet Management part of your web portal. For example, in Websphere, click **Administration** and then select **Web Modules** from the **Portlet Management** tree to access portlet administration.



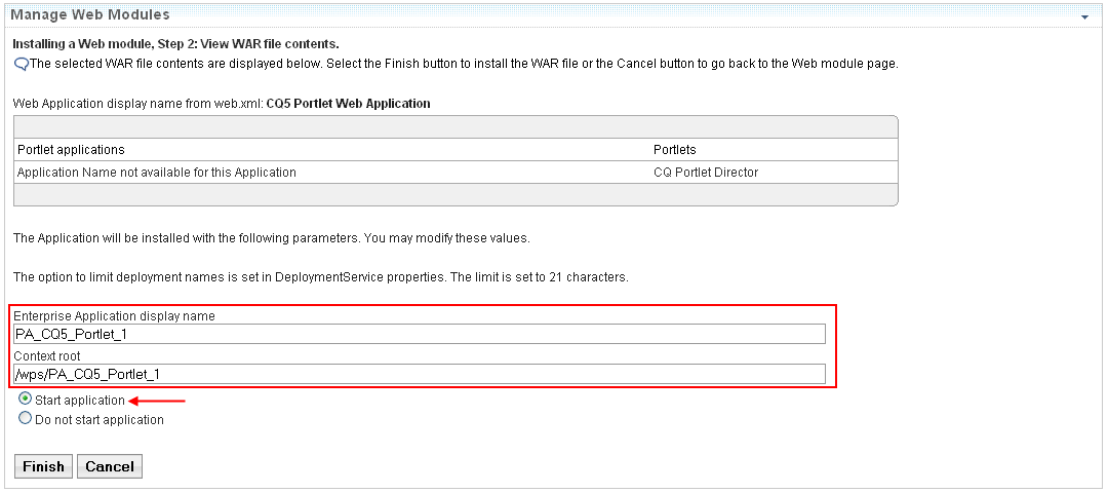
3. Click **`Install`** and browse to the supplied CQ5 portlet application (`cqportlet.war`) located in the `opt` director of the crx quickstart folder and click **`Next`** to enter other important information about the portlet.



Note

For other essential portlet information, you can either accept the defaults or change the values. If you accept the default values, the portlet is available at `http://<wps-host>:<port>/wps/PA_CQ5_Portlet`. The OSGi administration console provided by the portlet is available at `http://<wps-host>:<port>/wps/ PA_CQ5_Portlet/cqbridge/system/console` (default username/password is `admin/admin`).

4. Ensure that **`Start application`** is selected and click **`Finish`** to finish the installation.



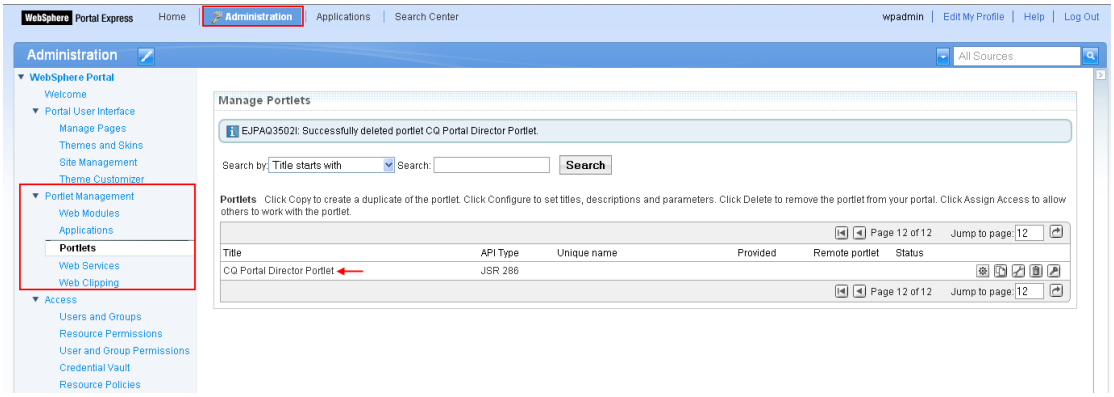
You see a message that your installation was successful.

9.2.2.2 Configuring the Portlet

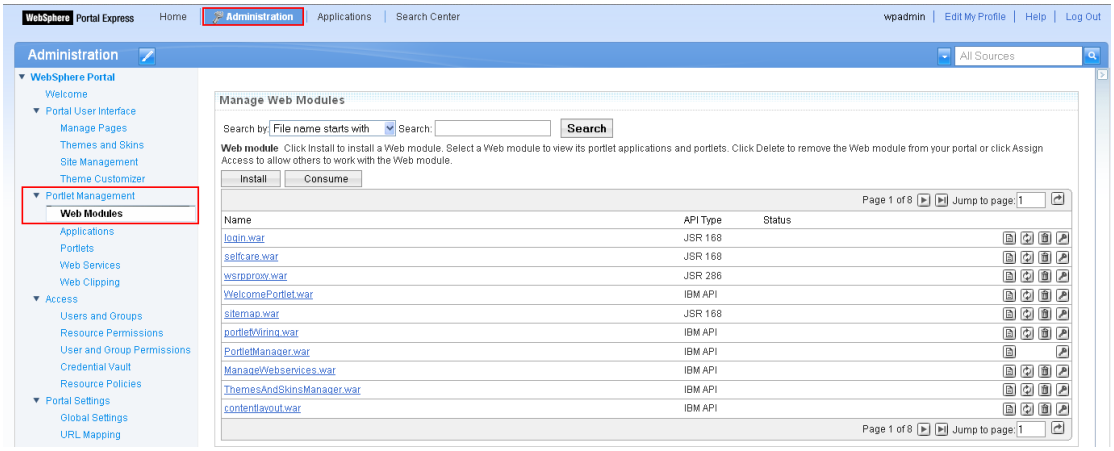
After you install the portlet, you need to configure it so that it knows the URLs of the underlying CQ5 instances (author and publish). In addition, you can configure additional options.

To configure the portlet:

1. Be sure you are in the **`Administration`** screen and click **`Portal Management`** and then **`Portlets`**. All portlets are listed. Skip to the last page to see the portlet you just added.



2. Click **`Configure Portlet`**. You see the current portlet configurations.



3. Modify portlet configurations, as necessary. For example, you may need to change the URL for the author and publish instances and the URL for the start path. The following are the default configurations:

Table 9.1. Default Portlet Configurations

authorBaseUrl	http://localhost:4502	The base URL to the author instance (without a trailing slash /)
publishBaseUrl	http://localhost:4503	The base URL to the publish instance (without a trailing slash /)
startPath	/content/geometrixx/en.html	The start path for the content to display (with leading slash / and extension)
htmlSelector	portlet	The selector to be appended to each content URL (see `appendMaxWindowStateAsSelector`, `appendHelpViewModeAsSelector`, and `addCssToPortalHeader`.)
appendMaxWindowStateAsSelector	TRUE	Indicates whether a selector is appended to a URL if the window is maximized
appendHelpViewModeAsSelector	TRUE	Indicates whether a selector is appended to a URL if the portlet is in Help mode
addCssToPortalHeader	TRUE	Indicates that the included CSS files be added to the portal page header

4. Click **OK** to save the configuration changes. After you have configured the portlet, you can add it to portal pages and use the portal.

9.2.2.3 Adding the Portlet to the Portal Page

To add the portlet to the portal page:

1. Be sure you are in the **`Administration`** screen and click **`Manage Pages`**.

2. Click the name of the portlet.

3. Select an existing page or create a new page.
4. Edit the page layout.
5. Click **Add Portlets** and select the portlet to add it to a container.
6. Click **Done** on the Edit Layout page to save your changes.

9.2.2.4 Using the Portlet

To access the page you added to the portlet in Adding the Portlet to the Portal Page:

1. In the portlet's personalization menu, configure the portlet as you configured it in the portal.
2. Click **Change**. The portlet displays the publish start URL configured in the portlet's configuration.

9.2.3 Caching and Cache Invalidation

The portlet, in its default configuration, caches the responses it receives from CQ WCM in a user-specific cache. The caches need to be invalidated when changes occur in the content of the publish instance. For this purpose, in CQ WCM a replication agent must be configured on the author instance. The cache can also be flushed manually. This section describes both of those procedures.

9.2.3.1 Flushing the Cache via Replication Agent

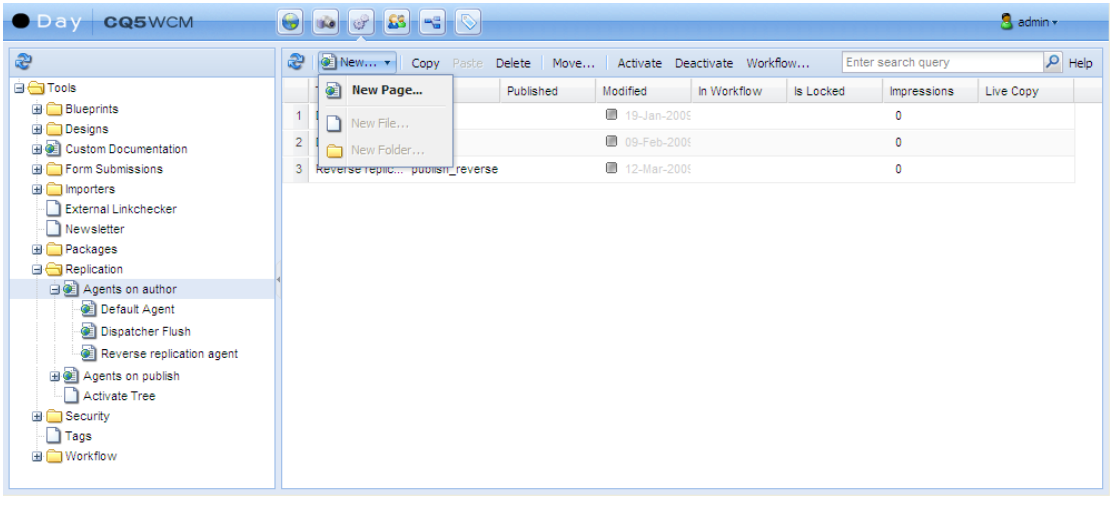
Just like the normal dispatcher invalidation, a replication agent can be configured to target the portal's CQ5 portlet cache. After you configure the replication agent, every regular page activation flushes the portal cache.

Note

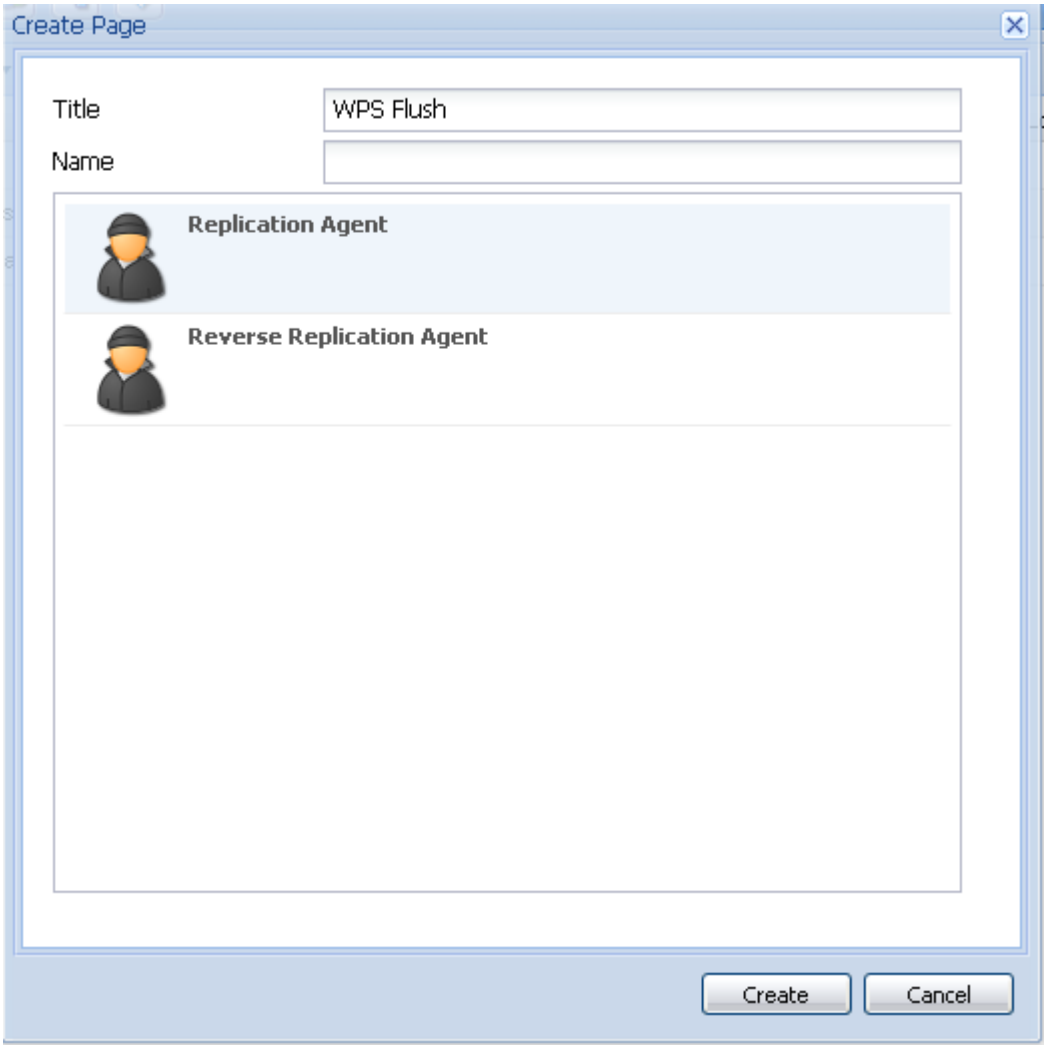
If you operate several portal nodes running the CQ portlet, you need to create an agent for each node as described in this procedure.

To configure a replication agent for the portal:

1. Log in to the author instance.
2. In site administration, click the **Tools** tab.
3. Click **New Page...** in the replication agents **New...** menu.



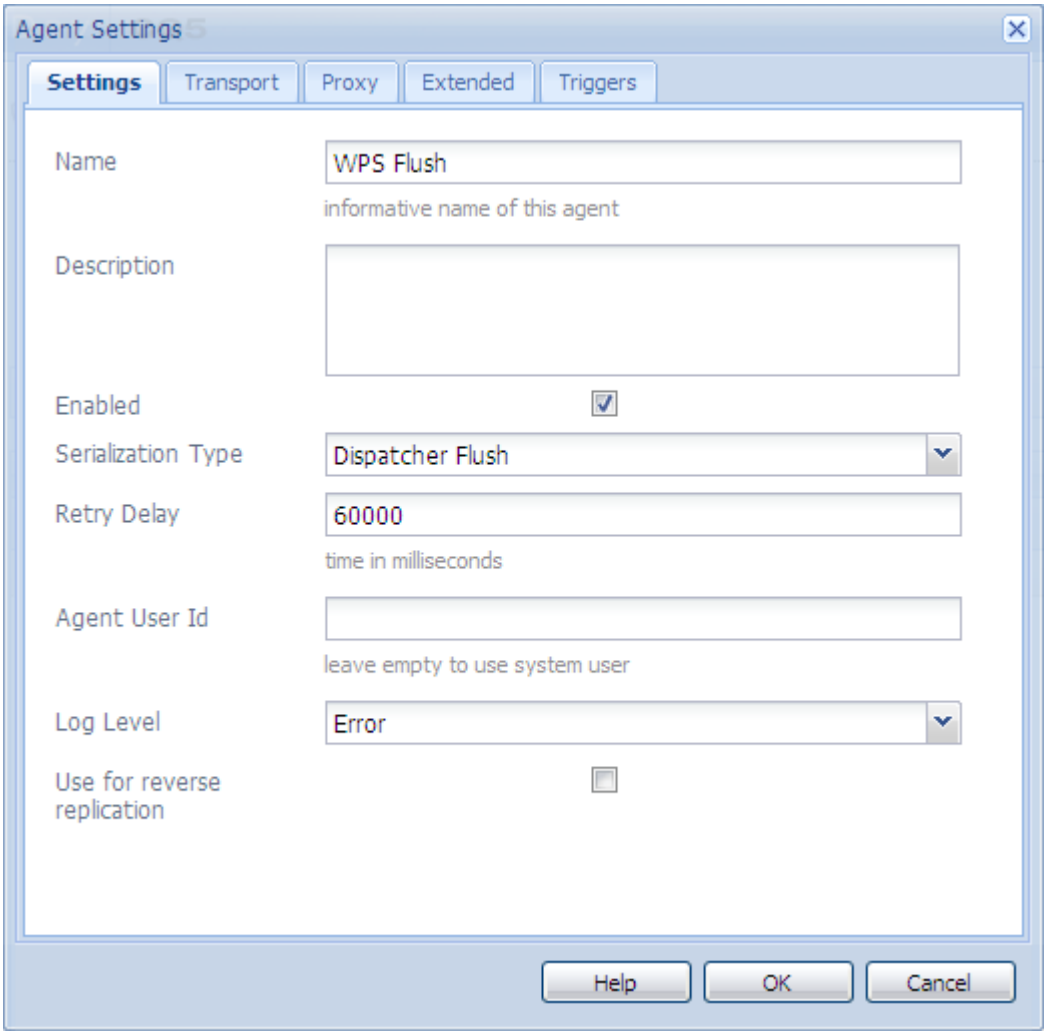
4. In **Template**, select **Replication Agent**, and enter a name for the agent. Click **Create**.



5. Double-click the replication agent you just created. It displays as invalid as it has not yet been configured.

6. Click **`Edit.`**

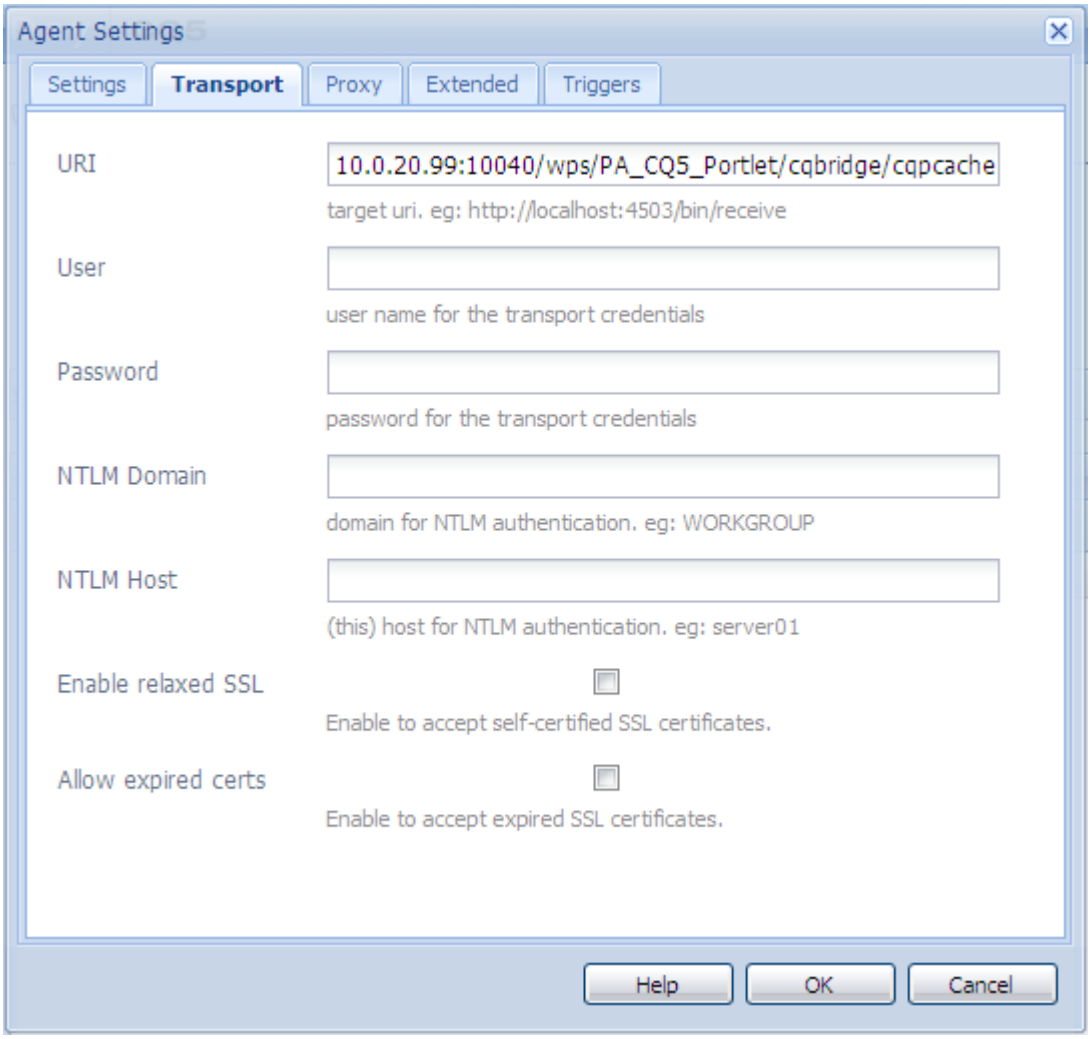
7. In the **`Settings`** tab, select the **`Enabled`** check box, select **`Dispatcher Flush`** as the serialization type, and enter a retry timeout (for example, 60000).



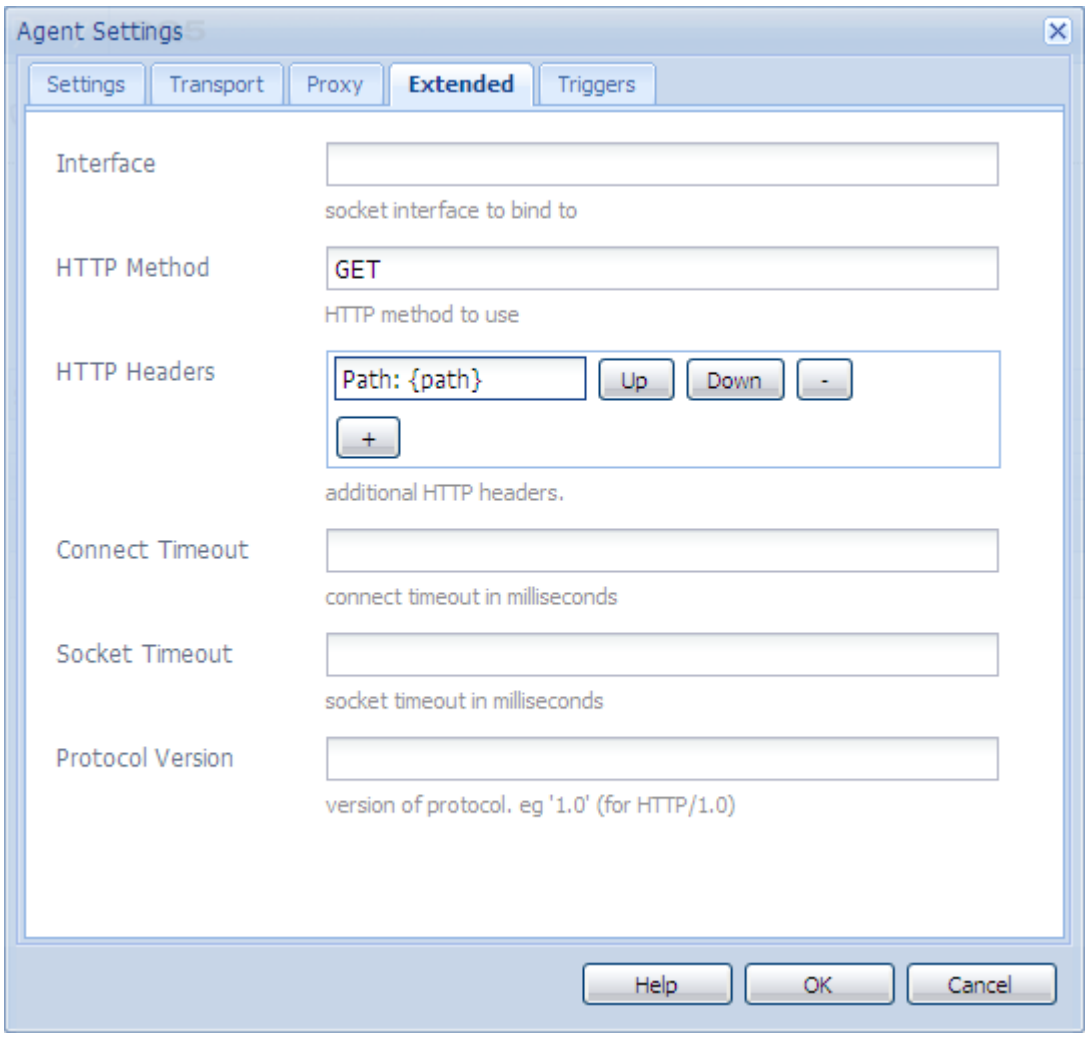
8. Click the **`Transport`** tab.

9. In the **`URI`** field, enter the flush URI (URL) of the portlet. The URI is in the following form:

```
http://<wps-host>:<port>/<wps-context>/<cq5-portlet-context>/cqbridge/cqpcache
```



10. Click the **`Extended`** tab.



11. In the **`HTTP Method`** field, type **`GET`**.
12. In the **`HTTP Headers`** field, click **`+`** to add a new entry and type **`Path: {path}`**.
13. If necessary, click the **`Proxy`** tab and enter proxy information to the agent.
14. Click **`OK`** to save changes.
15. To test the connection, click the **`Test Connection`** link. A log message appears that indicates whether the replication test succeeded. For example:

```
Replication test to http://10.0.20.99:10040/wps/PA_CQ5_Portlet/cqbridge/cqpcache
04.02.2009 16:21:03 - Create new HttpClient for WPS Flush
04.02.2009 16:21:03 - * HTTP Version: 1.1
04.02.2009 16:21:03 - adding header: Path:/content
04.02.2009 16:21:03 - deserialize content for delivery
04.02.2009 16:21:03 - content is empty. no message body
04.02.2009 16:21:03 - Connecting...
04.02.2009 16:21:03 - sending content, using GET
04.02.2009 16:21:03 - sent. Resonse: 200 OK
04.02.2009 16:21:03 - ------------------------------------------------
04.02.2009 16:21:03 - Sending message to 10.0.20.99:10040
04.02.2009 16:21:03 - >> GET /wps/PA_CQ5_Portlet/cqbridge/cqpcache HTTP/1.0
04.02.2009 16:21:03 - >> Path: /content
04.02.2009 16:21:03 - >> Content-Length: 0
04.02.2009 16:21:03 - >> Content-Type: application/octet-stream
04.02.2009 16:21:03 - --
04.02.2009 16:21:03 - << HTTP/1.1 200 OK
04.02.2009 16:21:03 - << Content-Language: en-US
04.02.2009 16:21:03 - << Content-Length: 98
04.02.2009 16:21:03 - << Date: Wed, 04 Feb 2009 15:21:02 GMT
04.02.2009 16:21:03 - << Server: WebSphere Application Server/6.1
04.02.2009 16:21:03 - <<
04.02.2009 16:21:03 - << <html><body><h1>Flushing Cache</h1><p>
04.02.2009 16:21:03 - << Cache flushed with path parameter
04.02.2009 16:21:03 - << /content
04.02.2009 16:21:03 - << </body></html>
04.02.2009 16:21:03 - Message sent.
04.02.2009 16:21:03 - ------------------------------------------------
04.02.2009 16:21:03 - Replication (TEST) of /content successful.
Replication test succeeded
```

9.2.3.2 Manually Flushing the Portlet Cache

You can manually flush the portlet cache by accessing the same URL configured for the replication agent. See Flushing the Cache for the form of the URL. In addition, the URL needs to be extended with a URL parameter `Path=<path>` to indicate what to flush.

For example:

`http://10.0.20.99:10040/wps/PA_CQ5_Portlet/cqbridge/cqpcache?Path=*` flushes the complete cache. `http://10.0.20.99:10040/wps/PA_CQ5_Portlet/cqbridge/cqpcache?Path=/content/mypage/xyz` flushes `/content/mypage/xyz` from the cache.

9.2.4 Authentication

This section describes the available authentication modes the portlet can use in communicating with the underlying CQ WCM instances.

9.2.4.1 Accessing the Portlet's Authentication Configuration

Authentication configuration options that the portlet uses in CQ WCM instances are available in the Apache Felix Web Management console (OSGi configuration),

To access the portlet's authentication configuration:

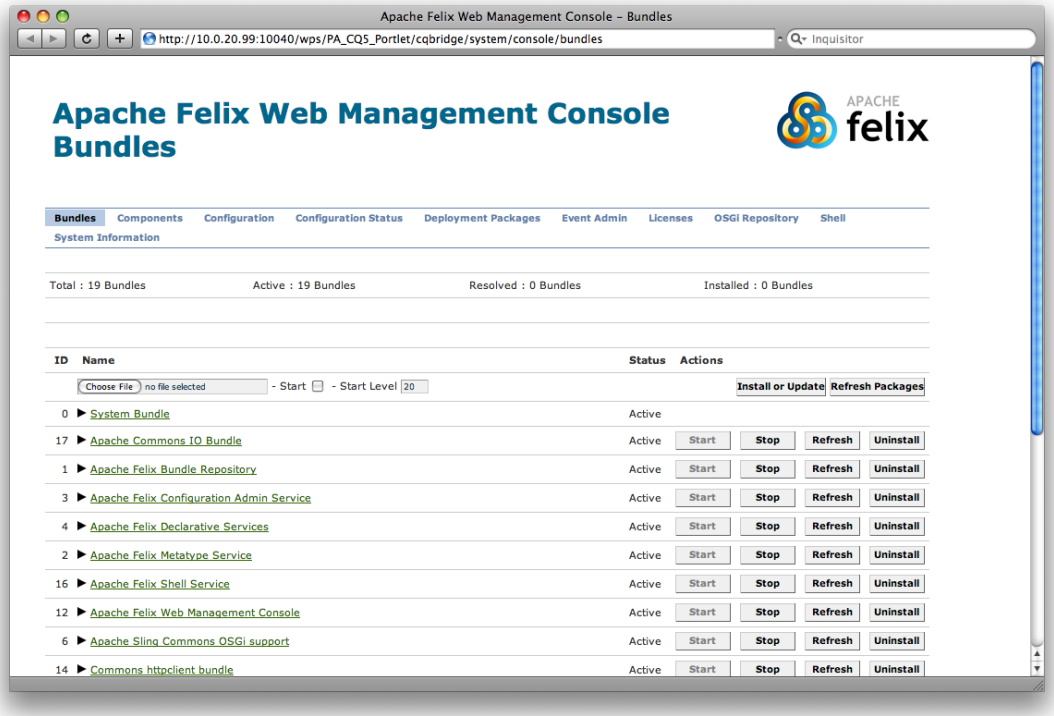
1. Access the Apache Felix Web Management console at the following URL:

   ```
   http://<wps-host>:<port>/<wps-context>/<cq5-portlet-context>/cqbridge/system/console
   ```

   For example, in its default configuration:

   ```
   http://wps-host:10040/wps/PA_CQ5_Portlet/cqbridge/system/console
   ```

2. Log in to the Apache Felix console. The default credentials are `admin/admin`. The Apache Felix Web Management Console opens:



3. In the Apache Felix Console, select **`Configuration`**.

4. In the **Configuration** menu, select a particular service to configure. Services are provided by the portlet in the OSGi framework.

The following services can be configured:

Table 9.2. Services that can be configured

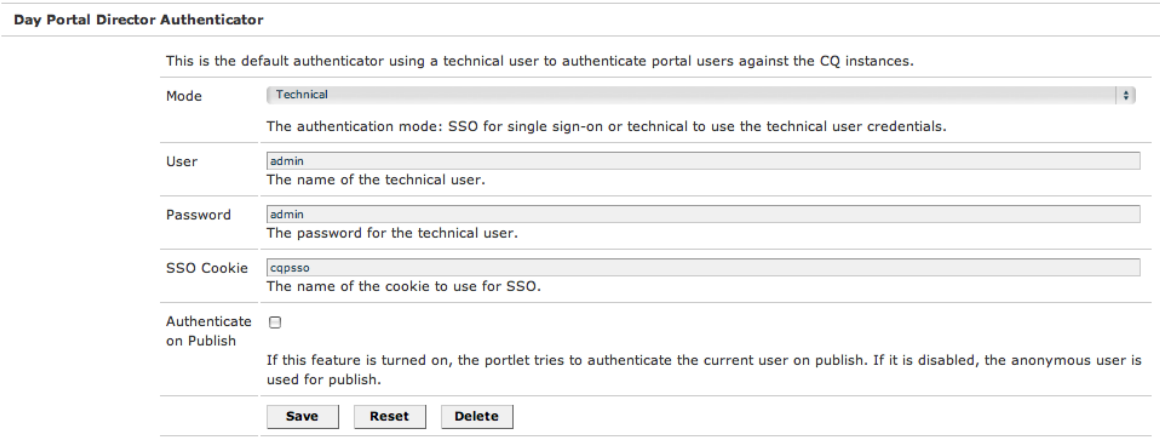
Service Name	Description
Day Portal Director Authenticator	Configure which authentication mode is used for CQ WCM instances. Depending on the selected mode, a technical user or the name of the SSO cookie can be specified. Also, authentication for CQ WCM publish instances can be enabled.

Service Name	Description
Day Portal Director File Cache	Configure the parameters of how the portlet caches responses it receives from CQ WCM instances.
Day Portal Director HTTP Client Service	Configure how the portlet connects via HTTP to underlying CQ WCM instances. You can, for example, specify a proxy server.
Day Portal Director Locale Handler	Configure which locales the portlet supports. Requests to CQ WCM instances are based on the user locale; for example, user language **German** would request `/content/geometrixx/de/...`.
Day Portal Director Privilege Manager	Configure whether the portlet should test the site administration based on the currently logged in user.
Day Portal Director Toolbar Renderer	Customize the rendering of the portlet's toolbar.

5. In addition, you can configure the OSGi management console and the logging service. For example, you can change the admin credentials for the Apache Felix Console by clicking the OSGi Management Console link.

9.2.4.2 Technical User Mode

In default mode, all requests issued by the portlet for the CQ WCM author instance are authenticated using the same technical user, regardless of the current portal user. Technical User mode is enabled by default. You enable/disable this mode in the respective configuration screen in the OSGi management console:



The technical user specified must exist on the CQ WCM author instance and on the publish instance if **`Authenticate on Publish`** is enabled. Be sure to give the user access privileges sufficient for authoring work.

9.2.4.3 SSO Authentication Mode

The portlet can authenticate for CQ WCM using the Single Sign On (SSO) scheme. In this mode, the user currently logged in to the portal is forwarded to CQ WCM in the form of an SSO cookie. If SSO mode is used, all portal users with access to the CQ portlet must be known to the underlying CQ WCM instances, most commonly in the form of CQ WCM being connected to LDAP, or by having manually created the users beforehand. Also, before enabling SSO in the portlet, the underlying CQ WCM author instance (and the publish instance, if **`Authenticate on Publish`** is enabled) needs to be configured to accept SSO-based requests.

To configure the portlet to use SSO authentication mode, complete the following steps (described in detail in the following sections):

1. Enable CQ WCM's CRX to accept trusted credentials.
2. Enable SSO authentication in the CQ WCM.
3. Enable SSO Authentication in the CQ portlet.

9.2.4.3.1 Enabling CQ WCM's CRX to accept trusted credentials

Before SSO can be enabled for CQ WCM, the underlying CRX instance needs to be configured to accept the trusted credentials provided by CQ WCM. To do this, you configure CRX's `repository.xml`.

To enable CRX to accept trusted credentials:

1. In the file system where CQ WCM is installed, open the following file:

```
/<cq-install-root>/crx-quickstart/server/runtime/0/_crx/WEB-INF/repository.xml
```

2. In the XML file, find the entry for the **`LoginModule`** and add the `trust_credentials_attribute` to its configuration:

```
<LoginModule class="com.day.crx.security.authentication.CRXLoginModule">
            <!--
            configuration options:
            - deny_anonymous_access (true|false)
              Per default this login-module allows anonymous access. This can
              be turned off by setting this parameter to true
            - trust_credentials_attribute [credential attribute name]
              Set the name of the SimpleCredential attribute, which indicates
              that LoginModule should trust credentials without further
 authentication.
              This can be used to let CRX participate in an SSO environment
              Defaults to empty
            - anonymous_principal [principal-name]
              Set the principal's name to be assigned to an anonymous authentication
 request.
              This can be used to assign more elaborted access-rights for
              anonymous users.
              Defaults to "anonymous"
            -->
            <param name="trust_credentials_attribute" value="TrustedInfo"/>
            <param name="anonymous_principal" value="anonymous"/>
</LoginModule>
```

3. Restart CQ WCM for the changes to take effect.

9.2.4.3.2 Enabling SSO authentication in the CQ WCM

To enable SSO in CQ WCM, access the relevant configuration entry in the CQ WCM's Apache Felix Web Management Console (OSGi):

1. Access the console through its URI at `http://<cq5-host>:<port>/system/console`.
2. In the **`Configuration`** menu, select **`SSO Authentication Handler`**. In this example, the SSO handler accepts SSO requests for all paths based on the cookie provided by the CQ portlet. Your configuration may vary.

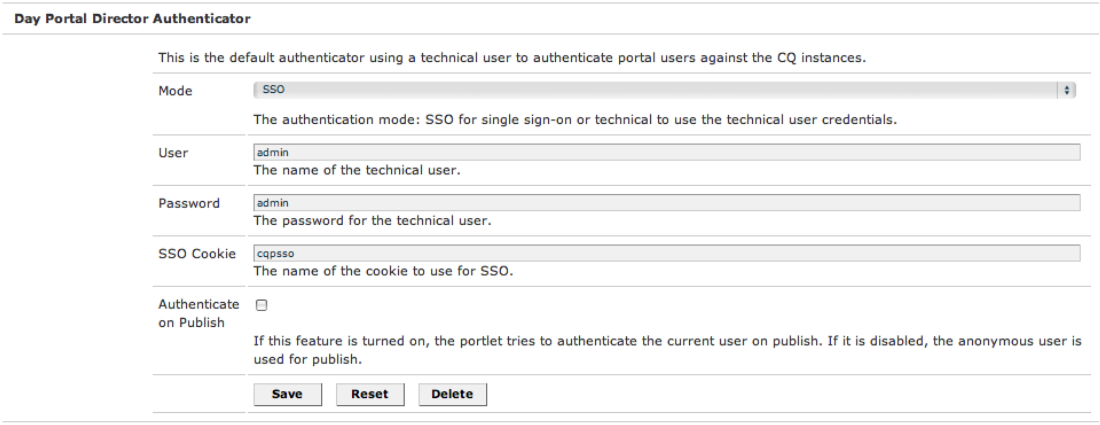


Table 9.3. Example configuration parameters

Path	/	Enables SSO handler for all requests
Cookie Names	cqpsso	Name of the cookie provided by the portlet as configured in the portlet's OSGi console

3. Click **Save** to enable SSO. SSO is now the primary authentication scheme.

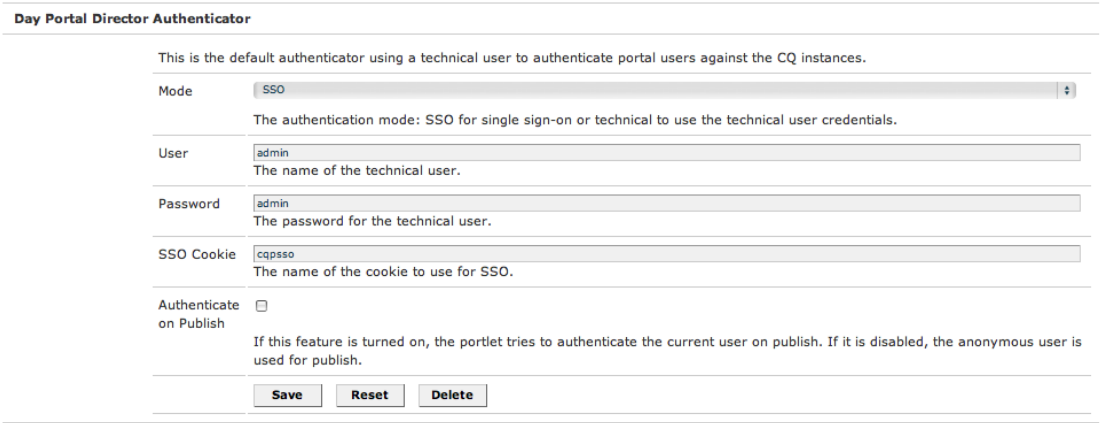
For every request CQ WCM receives, first the SSO-based authentication is attempted. Upon failure, a fallback to the usual basic authentication scheme is performed. As such, normal connections to CQ WCM without SSO remain possible.

9.2.4.3.3 Enabling SSO Authentication in a CQ Portlet

In order for the underlying CQ WCM instance to accept SSO requests, the portlet's authentication mode has to be switched from **Technical** to **SSO**.

To enable SSO authentication in a CQ portlet:

1. Access the console through its URI at `http://<cq5-host>:<port>/system/console`.
2. In the **Configuration** menu, select **Day Portal Director Authenticator** from the list of available configurations.
3. In **Mode**, select **SSO**. Leave the other parameters with their default values.



4. Click **Save** to enable SSO for the portlet.

Note

For testing purposes, access the portlet with your portal's administrative user, after you create the same user in CQ WCM with administrator privileges.

After performing this procedure, requests are authenticated using SSO. A typical snippet from the HTTP communication reveals the presence of the following SSO and Portlet specific headers:

```
C-12-#001898 -> [GET /mynet/en/_jcr_content/par/textimage/image.img.png HTTP/1.1 ]
C-12-#001963 -> [cq5:locale: en ]
C-12-#001979 -> [cq5:used-locale: en ]
C-12-#002000 -> [cq5:locales: en,en_US ]
C-12-#002023 -> [cqp:user: wpadmin ]
C-12-#002042 -> [cqp:portal: IBM WebSphere Portal/6.1 ]
C-12-#002080 -> [cqp:windowid: 7_CGAH47L000CE302V2KFNOG0084 ]
C-12-#002124 -> [cqp:windowstate: normal ]
C-12-#002149 -> [cqp:portletmode: view ]
C-12-#002172 -> [User-Agent: Jakarta Commons-HttpClient/3.1 ]
C-12-#002216 -> [Host: 10.0.0.68:4502 ]
C-12-#002238 -> [Cookie: $Version=0; cqpsso=Basic+d3BhZG1pbg%3D%3D ]
C-12-#002289 -> [ ]
```

9.2.5 Customizing the Portlet Toolbar

The portlet's toolbar layout can be customized by installing a bundle through the portlet's Felix Web Console, which contains custom CSS/HTML at a predefined location.

9.2.5.1 Bundle Structure

The following is an example bundle structure:

```
$ jar tvf target/toolbarlayout-0.0.1-SNAPSHOT.jar | awk '{print $8}'
META-INF/
META-INF/MANIFEST.MF
/com/day/cq/portlet/toolbar/layout/
/com/day/cq/portlet/toolbar/layout/author.gif
/com/day/cq/portlet/toolbar/layout/back.gif
/com/day/cq/portlet/toolbar/layout/button.html
/com/day/cq/portlet/toolbar/layout/edit.gif
/com/day/cq/portlet/toolbar/layout/manage.html
/com/day/cq/portlet/toolbar/layout/publish.html
/com/day/cq/portlet/toolbar/layout/refresh.gif
/com/day/cq/portlet/toolbar/layout/siteadmin.gif
/com/day/cq/portlet/toolbar/layout/toolbar.css
```

The `META-INF` folder contains the `MANIFEST.MF` file required by OSGi to identify it as a bundle. It appears as follows:

```
Manifest-Version: 1.0
Built-By: djaeggi
Created-By: Apache Maven Bundle Plugin
Import-Package: com.day.cq.portlet.toolbar.layout
Bnd-LastModified: 1234178347159
Export-Package: com.day.cq.portlet.toolbar.layout
Bundle-Version: 0.0.1.SNAPSHOT
Bundle-Name: Company CQ5 Portal Director Portlet Toolbar Layout
Bundle-Description: This bundle provides a custom layout for the CQ5 P
 ortal Director Portlet Toolbar.
Build-Jdk: 1.5.0_16
Bundle-ManifestVersion: 2
Bundle-SymbolicName: com.day.cq.portlet.company.toolbarlayout
Tool: Bnd-0.0.255
```

The fact that the HTML/CSS/images are within the `/com/day/cq/portlet/toolbar/layout` folder is mandated by the portlet and cannot be changed. Along the same lines, the `Import-`

`Package` and `Export-Package` headers in `MANIFEST.MF` must be called /com/day/cq/portlet/toolbar/layout as well. The `Bundle-SymbolicName` must be a unique, fully qualified package name.

You can build it using a tool such as maven or manually create such a jar file with the relevant header set as shown in this section.

9.2.5.2 Portlet Toolbar Views

The portlet's toolbar basically has two view states. Each view and associated buttons can be customized with a respective HTML file.

9.2.5.2.1 Publish View

The publish view only has one button that switches the toolbar to the **`Manage`** view. The publish view is represented by the `publish.html` file in previous bundle. In the HTML, you can use the following placeholders, which are replaced by the portlet with the respective contents when rendered:

Table 9.4. Publish View Placeholders

Placeholder String	Description
{buttonManage}	Placeholder is replaced by the **`Manage`** button, which switches the portlet state into the management state.

9.2.5.2.2 Manage View

The manage view has four buttons: **`Edit, Site Admin, Refresh`** and **`Back`**. The manage view is represented by the `manage.html` file in the previous bundle. In the HTML, you can use the following placeholders, which are replaced by the portlet with the respective contents when rendered:

Table 9.5. Manage View Placeholders

Placeholder String	Description
{buttonEdit}	Placeholder is replaced by the **`Edit`** button, which opens a new window with the current page in CQ's edit mode.
{buttonSiteadmin}	Placeholder, replaced by a button which opens the site administration of CQ WCM.
{buttonRefresh}	Refreshes the current view.
{buttonBack}	Switches the portlet back into the publish view.

9.2.5.2.3 Buttons

Buttons, on whichever view they appear, use the same common HTML, defined in `button.html`.

In the HTML, you can use the following placeholders, which are replaced by the portlet with the respective contents when rendered:

Table 9.6. Manage and Publish View Buttons

Placeholder String	Description
{name}	Name of the button, for example, **`author, back, refresh,`** and so on.

Placeholder String	Description
{id}	CSS id of the button.
{url}	URL for the button's target.
{text}	Label of the button.
{onclick}	Javascript **`onclick`** function (contains {url}).

Example of a `button.html` file:

```
<div class="cqp_button">

 <a href="#" onclick="{onclick}">

 <img src="/wps/PA_CQ5_Portlet/cqbridge/static/{id}.gif" alt="{text}"
title="{text}"/>

 </a>
</div>
```

9.2.5.3 Installing a Custom Layout

To install a custom layout, access the portlet's OSGI Web console **`Bundles`** section and upload the bundle.

9.3 Packages

If you need to upload, or create, packages for your installation, then see Package Manager in the CRX documentation for detailed instructions.

10 Checklists

The following lists highlight some specific high-level points; the lists are not meant to be exhaustive but aim to give some pointers and provide a basis for your own checklists. This checklist deals with WCM issues, and should be considered together with your own System Administration tasks.

Table 10.1. Installation Checklist

Check	Considerations	Comments / Actions
Supported Platforms		See Supported Platforms.
Prerequisites		See Prequisites.
Sizing		See Minimum Sizing Requirements.
New Installation or Upgrade?		
	From CQ 5.1 to CQ 5.2?	See Upgrading from CQ 5.1 to CQ 5.2.
	From Communiqué 3 or 4?	See Upgrading from Communiqué 3 or 4.
Default or Custom Installation?		
	Will you be using a third party application server?	See How to install CQ5 with an Application Server.
	Will you be using the default port number?	See Changing the Port Number by Renaming the File.
	Will you be using the default persistence manager (TarPM)?	See Persistance Managers and Databases. and Configuring a Persistence Manager other than Tar PM.
	Do you want to start CQ WCM as a Windows Service?	See Installing and Starting CQ5 WCM as a Windows Service.
Do you want to use the Dispatcher?		See Dispatcher.

Table 10.2. Configuration Checklist

Check	Considerations	Comments / Actions
Do you need the geometrixx website?		If not needed this can be deleted from CQ WCM Siteadmin.
Do you need all the modules that are installed as standard?		If not needed these can be removed.
Is the root mapping correct?		The default root mappings are: **author**: `/libs/wcm/content/welcome.html` **publish**:

Check	Considerations	Comments / Actions
		`/content` If necessary update these as required. See CQ Root Mapping.
Configuring the replication agents.		See How to configure your Replication Agents.
Security Checklist		
	Default accounts and passwords	See Change Default Passwords.
	WebDAV	See Disable WebDAV.
	Access via the Dispatcher	See Restrict Access via the Dispatcher.
	Cross-scripting	See Check for Cross-Site Scripting (XSS).
Configuring the Dispatcher		See Dispatcher.
Do you want to configure a cluster?		See Using Clusters.
Will you be using LDAP?		See Configuring LDAP.
Will you need to repeat the installation and configuration? For example, during development and testing, or over multiple servers?	Some configurations are not saved in the repository, and therefore not included in packages.	Explicitly save your OSGi configurations to a file for easy deployment.

Table 10.3. Maintenance Checklist

Check	Considerations	Comments / Actions
Backup plan.		See How To Backup your CQ Instance.
Disaster recovery plan.	Your company's disaster recovery guidelines.	

Appendix A. Security Checklist

This section deals with various steps you should take to ensure that your CQ5 installation is secure.

A.1 Change Default Passwords

Day strongly recommends that you change the passwords for the following (privileged) `admin` accounts (on all instances) after installation:

1. The **`CQ`** `admin` account.

   Important

   The CQ admin account and the CRX admin accounts are actually one and the same. So once you have changed the password for the “CQ admin” account, you will need to use the new password when accessing CRX.

   Important

   To change the password for the CQ / CRX `admin` account, you need to make changes in **both** CRX and the OSGi Console. See the section called “Changing the CQ admin password in the CRX console” and the section called “Changing the CQ admin password in the OSGi Apache Felix console”.

2. The **`CQSE`** (Communiqué Servlet Engine) `admin` account.

3. The **`Apache Felix Web Management Console`** `admin` password.

Note

Further actions are described in the table the section called “Default Users and Groups”, which gives an overview of the default users and groups included in the standard installation.

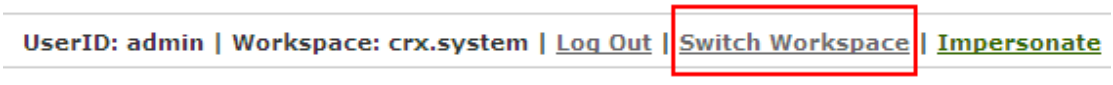
A.1.1 Changing the CQ admin password

To change the password for the CQ `admin` account, you need to make changes in **both** CRX and the OSGi Console.

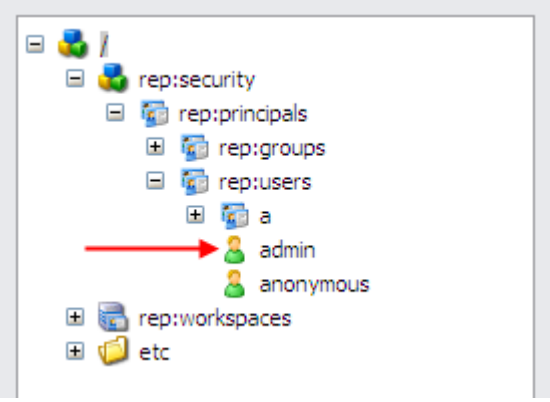
A.1.1.1 Changing the CQ admin password in the CRX console

To change the `admin` account in the CRX console:

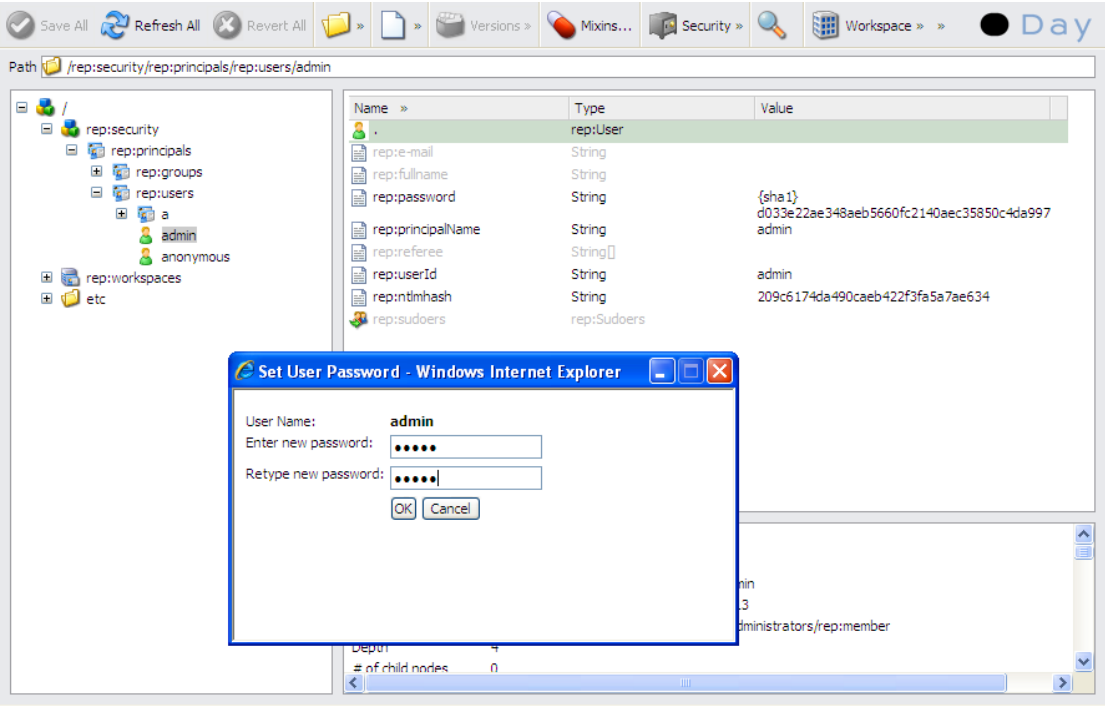
1. Navigate to `http://<server>:<port_number>/crx` to open the CRX console.
2. Log in as `admin` to the `crx.system` workspace.



3. Open the Content Explorer and navigate to the `admin` user and select it.



4. In the **Security** menu, select **Set User Password**. A **Set User Password** window opens.



5. Enter the new password and re-enter to confirm and click **OK** to save your changes.

Note

The new password is instantly persisted in the repository, a dedicated click on **Save All** is not required.

A.1.1.2 Changing the CQ admin password in the OSGi Apache Felix console

To change the `admin` account in the OSGi Apache Felix console:

1. Navigate to `http://<server>:<port_number>/system/console/configMgr`, and login as `admin`, to open **Configurations** in the Apache Felix console.

2. In the **Configurations** menu, select **CRX Sling Client Repository**.

3. In the **`Admin`** password field, change the password to match the one you entered in the CRX console.
4. Click **`Save`** to save your changes.

A.1.2 Changing the admin password for CQSE

To change the `admin` account in the CQSE console:

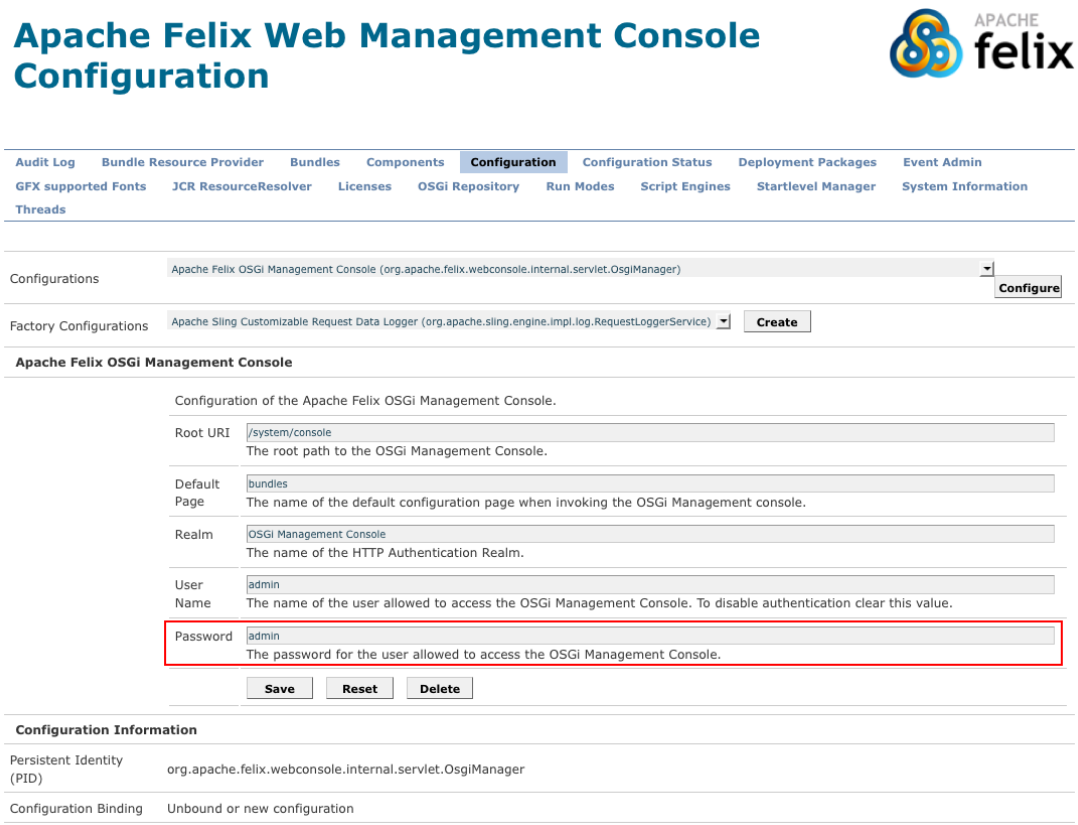
1. Navigate to `http://<server>:<port_number>/admin` to open the CRX console.
2. Log in as `admin` (the default password is `admin`).
3. Select the **`Change Password`** tab:

4. Enter the **Old Password**, your **New Password**, then **Confirm** the new password.
5. Click **Change** to save the new password.

A.1.3 Changing the admin password for the Apache Felix Web Management Console

To change the admin account in the OSGi Apache Felix console:

1. Navigate to http://<server>:<port_number>/system/console/configMgr, and login as admin, to open **Configurations** in the Apache Felix console.
2. In the **Configurations** menu, select **Apache Felix OSGi Management Console**.



3. In the **Password** field, change the password.
4. Click **Save** to save your changes.

A.2 Disable WebDAV

WebDAV should be disabled on the publish environment.

This needs action at two levels:

1. Reconfigure the CRX webapp:
   a. Open the web.xml of the CRX webapp; this is usually found in <cq-installation-dir>/crx-quickstart/server/runtime/0/_crx/WEB-INF.
   b. Comment out the following servlets in the *SERVLETMAPPING* section to effectively disable WebDAV access to the repository:
      - Webdav

- JCRWebdavServer
- CqResource

c. The resulting configuration should look like:

```
<!-- ========================================================================= -->
  <!-- S E R V L E T   M A P P I N G                                             -->
  <!-- ========================================================================= -->
  <servlet-mapping>
    <servlet-name>NodeTree</servlet-name>
    <url-pattern>/ui/nodetree/*</url-pattern>
  </servlet-mapping>
  <!--servlet-mapping>
    <servlet-name>Webdav</servlet-name>
    <url-pattern>/repository/*</url-pattern>
  </servlet-mapping-->
  <servlet-mapping>
    <servlet-name>Export</servlet-name>
    <url-pattern>/export/*</url-pattern>
  </servlet-mapping>
  <!--servlet-mapping>
    <servlet-name>JCRWebdavServer</servlet-name>
    <url-pattern>/server/*</url-pattern>
  </servlet-mapping-->
  <!--servlet-mapping>
    <servlet-name>CqResource</servlet-name>
    <url-pattern>/cqresource/*</url-pattern>
  </servlet-mapping-->
  <servlet-mapping>
    <servlet-name>JCRExplorer</servlet-name>
    <url-pattern>/</url-pattern>
  </servlet-mapping>
```

d. Restart your CQ/CRX instance to make the changes take effect.

2. Configure CQ by stopping the appropriate bundle:

a. Connect to the **`Felix Management Console`** running on `http://<host>:<port>/system/console`; for example `http://localhost:4503/system/console/bundles`.

b. In the list of bundles, find the bundle named **`Sling - Simple WebDAV Access to repositories`**.

c. Click the stop button

to stop this bundle. A restart is not required.

A.3 Restrict Access via the Dispatcher

By configuring the Dispatcher you should restrict access so that only the following are available to external visitors:

- `/content` - Site content
- `/etc` - Miscellaneous content such as designs

The following should be entered in the configuration file `dispatcher.any`:

```
# only handle the requests in the following acl. default is 'none'
# the glob pattern is matched against the first request line
/filter
```

```
  {
  /0001
    {
    /glob "*"
    /type "deny"
    }
  /0002
    {
    /glob "* /content[./]"
    /type "allow"
    }
  /0003
    {
    /glob "* /etc[./]"
    /type "allow"
}
```

Note

This configuration includes the following restrictions:

1. Restricts access to the Servlet Engine Administration `/admin`
2. Restricts access to the Sling Console `/system`
3. Restricts access to CRX `/crx`
4. Restricts access to the following application specific folders:
   - `/apps` – Application data
   - `/libs` – CQ5 Library
   - `/var` – var folder
   - `/etc` – Miscellaneous folder
   - `/home` – User's home folder
5. Restricts access to `/tmp`
6. Denies POST requests in case forms are not used.

A.4 Check for Cross-Site Scripting (XSS)

Cross-site scripting (XSS) allows attackers to inject code into web pages viewed by other users. This security vulnerability can be exploited by malicious web users to bypass access controls.

Warning

CQ5 example code is not protected against such attacks.

Appendix B. Copyright, Licenses and Formatting Conventions

For all copyright statements and license agreements see Copyright, Licenses and Disclaimers.

B.1 Formatting Conventions

The following tables detail formatting conventions used within this guide:

Table B.1. Formatting Conventions - Text

Style	Description	Example
Cross-reference	Cross-reference to external documents.	See the *Microsoft Manual of Style for Technical Publications*.
GUI Item	User interface items.	Click **Save**.
Keyboard shortcut	Keyboard shortcuts.	Press **Ctrl+A**.
Mouse Button	Mouse buttons.	**Secondary-mouse** button (usually the **right-mouse** button).
Link	Link to anchor-points within the current document and/or external sources.	http://www.day.com
`Code`	Example of programming code.	`if (weather == sunny) smile;`
User Input	Example of text, or commands, that you type.	**ls *.xml**
<Variable User Input>	Example of variable text - you type the actual value needed.	`ls <cq-installation-dir>`
[Optional Parameter]	An optional parameter.	`ls [<option>] [<filename>]`
`Computer Output`	Logging and error messages.	`ls: cannot access error.log:`

Table B.2. Formatting Conventions - Actions

When you see this...	It means do this...
Ctrl+A	Hold down the **Ctrl** key, then press the **A** key.
Right-click	Press the **right-mouse** button (or the **left-mouse** button if your mouse has been configured for left-handed use).
Drag	Hold down the left mouse button while moving the item, then release the mouse button at the new location (or the right mouse button if your mouse has been configured for left-handed use).