SDL Tridion 2013 Installation Manual

Content Management Technologies Division of SDL

SDL Tridion 2013 Installation Manual

This product may include open source or similar software. The NOTICE.pdf, available on the product CD-ROM, lists third-party software used in this product.

Although SDL Tridion Development Lab B.V. takes all reasonable measures to provide accurate and comprehensive information about the product, this information is provided as-is and all warranties, conditions or other terms concerning the documentation whether express or implied by statute, common law or otherwise (including those relating to satisfactory quality and fitness for purposes) are excluded to the extent permitted by law.

To access the latest documentation, go to the SDL Tridion documentation portal: http://docportal.sdl.com/sdltridion. You require a user name and password, issued by SDL Tridion Development Lab B.V., to access this content.

Table of contents

Chapter 1 Installing SDL Tridion core software modules

This section provides an overview of the installation procedure for installing and configuring the core software modules of SDL Tridion so that you can create, manage and publish content.

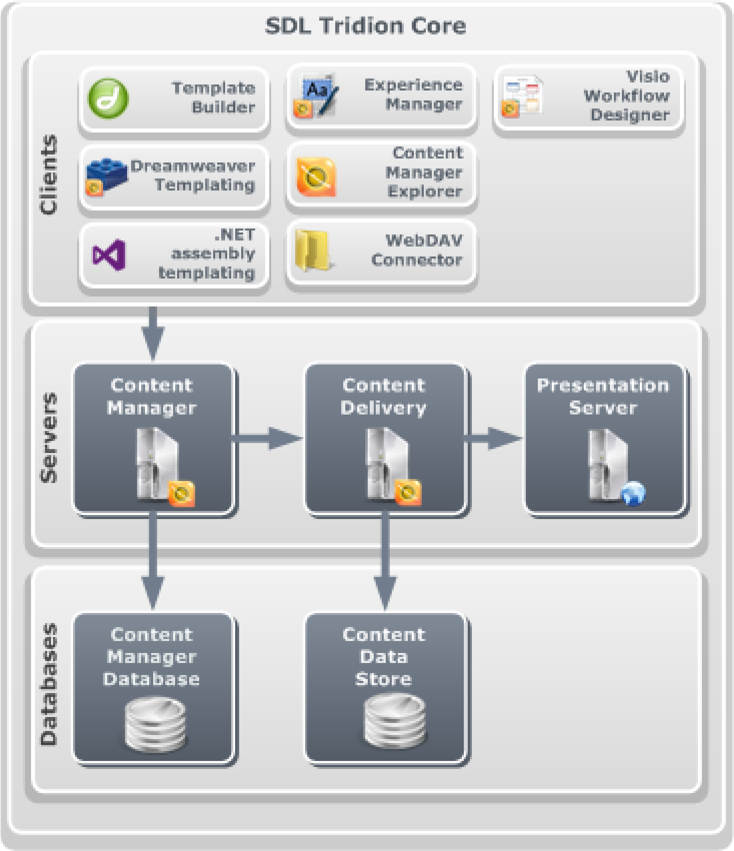
For information on how to install additional modules such as WebDAV Connector, Visio Workflow Designer, Cache Channel Service, Monitoring, User Generated Content, Content Delivery Web service, Audience Manager and Outbound E-mail, and Translation Manager, refer to the relevant implementation section in the documentation portal.

1.1 SDL Tridion core system diagram

The SDL Tridion Core consists of Content Manager and Content Delivery subsystems, the databases these subsystems require and the clients used to create content.

Content Manager is the environment for creating, managing and assembling the various Building Blocks used to build Web sites or content intended for other channels. Users create content using various **Content Manager Clients** and content is stored in the **Content Manager Database**.

Content Delivery handles and processes content published from Content Manager, transforms the content into dynamic Web site content using the Content Delivery APIs, stores the published content in the **Content Data Store** (a file system, a relational database, custom storage media, or a combination of these), and puts the content on the **Presentation Server**, the machine that serves content to Web site visitors.



1.2 Installation overview

This topic shows which steps are involved in the installation of the core software modules.

Step 1: Check hardware prerequisites

Before you start installing, check if the hardware you intend to use meets the minimal hardware requirements.

Step 2: Create the Content Manager database

Installing the Content Manager database consists of the following steps:

1. Ensure that the software prerequisites for the Content Manager database are met.
2. Create a Windows user to access the Content Manager database and act as a Content Manager user.
3. Use a PowerShell script to install the Content Manager database.

Step 3: Install and configure the Content Manager server

Installing and configuring the Content Manager server consists of the following steps:

1. Ensure that the software prerequisites for the Content Manager server are met.

2. Run the Content Manager Windows installer. This installer installs the Content Manager server, including the server-side parts of the various clients.

3. Configure the Content Manager server core.

4. Configure the server-side parts of the various clients.

Step 4: Install and configure the Content Data Store

If you intend to publish to a relational database:

1. Ensure that the software prerequisites for the Content Data Store are met.

2. Use a PowerShell script to install the Content Data Store (or run the database scripts if you are installing on IBM DB2).

Note: Installing the Content Data Store is not necessary if you intend to publish to a local file system only.

Step 5: Install and configure core Content Delivery Server Roles

A Server Role encapsulates the files and resources you need to install on a particular server to deploy and run a certain piece of Content Delivery functionality. Installing and configuring Content Delivery Server Roles consists of the following steps:

1. Ensure that the software prerequisites for the Content Delivery server are met.

2. Install the Content Delivery Server Role.

   Installing Content Delivery involves installing one or more Content Delivery Server Roles as Windows Services, Java Processes, or in a .NET or Java Web application.

   - **Content Deployer (HTTP or HTTPS) or Content Deployer (other protocols)** —you need to install and configure one of these types of Content Deployer, and the Storage Layer (included in each of these Server Roles). The Content Deployer receives content Published from the Content Manager and passes processed content to the Storage Layer, which stores content in a database or on a filesystem.

- **Experience Manager Web service and Experience Manager Web site extension**—these Server Roles are required to make the Experience Manager client work.
- **API Server Role**—this Server Role resolves dynamic links between content items.

3. Configure logging for the Server Role you installed.

4. Configure storage for the Server Role you installed.

Step 6: Set up publishing (Content Distribution)

Enable publishing from the Content Manager to Content Delivery.

Step 7: Install and configure the core Content Manager clients

Installing and configuring these clients consists of the following steps:

1. Ensure that the software prerequisites for these clients are met.

2. Install and configure the clients:
   - Content Manager Explorer
   - Template Builder
   - .NET assembly templating
   - Dreamweaver templating
   - Experience Manager client

1.3 Licenses

SDL Tridion Customer Support provides you with license files that enable modules and features based on your license agreement with SDL Tridion.

SDL Tridion Customer Support normally provides you with the following files or license keys to add to these files:

- `license.xml`—contains license keys for the various Content Manager server software components
- `cd_licenses.xml`—contains license keys for publishing content to the Content Delivery server using one or more transport protocols.

The license files contain information about the modules and features for which you are licensed. You need to specify your license files when you install SDL Tridion using the installers, otherwise you need to add license files to your Content Manager and Content Delivery installations manually.

For information about how to contact SDL Tridion Customer Support, refer to the Contact page on SDL Tridion World Web site: http://www.sdltridionworld.com.

1.4 User rights required for installation

Unless indicated otherwise, all installation tasks must be performed by a user with administrator-level user rights to the machine on which the software component is to be installed.

Chapter 2 Using PowerShell database scripts

To install or upgrade SDL Tridion database scripts, use the PowerShell scripts provided on your SDL Tridion installation media.

2.1 Executing and troubleshooting PowerShell scripts

You create databases for SDL Tridion by executing scripts using Windows PowerShell. If you are unfamiliar with PowerShell, you may need to execute perform some actions before you can execute the scripts.

Requirements

To install SDL Tridion databases on a Windows machine requires Windows PowerShell 3.0 and Microsoft .NET Framework 4.0 or higher.

Steps to execute

1. On a supported Windows machine, click the Start button in the lower-left corner of your screen and type `PowerShell` in the search box to quickly access and open the Windows PowerShell program.

2. In PowerShell, depending on your database navigate to one of the following folders on the installation media:

   - `Database\MSSQL\` or
   - `Database\Oracle\`

3. Enter `ls` to view the available scripts.

4. By default, PowerShell applies timeouts of 10 minutes to your database operations. If you expect this to be insufficient because your database is too large, change the timeout values by setting them as follows:

   ```
   $DatabaseScriptConnectTimeout=<value>
   $DatabaseScriptCommandTimeout=<value>
   ```

   where <*value*> is the timeout value in seconds (for example, 3600 sets a 1-hour timeout value).

5. Type, for example, the following command to run a script:

   ```
   & '.\Install Content Manager database.ps1'
   ```

6. If you get the following errors:

   File <*ScriptName*>.ps1 cannot be loaded because the execution of scripts is disabled on this system

   Enable the running of PowerShell scripts by entering the following PowerShell command:

```
Set-ExecutionPolicy AllSigned
```

Note: If the command fails due to registry write errors, select Windows PowerShell **Run as administrator**.

Could not load file or assembly ***<Path>*****\Database\Scripts\Tridion.Database.Management.dll or one of its dependencies**

In Windows Explorer, navigate to the installation media `\Database\Scripts\` folder, select `Tridion.Database.Management.dll` and click **Properties**. In the **General** tab, click **Unblock**. Click **OK** to close the dialog.

7. Follow the instructions in the PowerShell console to install the database.

2.2 Viewing PowerShell database script parameters

To find about the database script parameters you can specify, open a PowerShell prompt and execute the command `get-help <script> -detailed`, where <*script*> is the full path and name of a database script.

2.3 Specifying an Oracle host, port and service name for your PowerShell scripts

You can specify the host, port and service name of your Oracle database in several ways, before or during installation.

Steps to execute

1. If you have created a `TNSNAME.ora` file in which you define a Net Service Name for this database, when PowerShell prompts you for a **Net Service**, type the name of the Net Service Name you configured.

2. Alternatively, you can enter the string `%manual%` as a value for **Net Service**. This results in three new prompts: **Host name**, **Port** and **Service name**. This is useful if you are installing manually but have no `TNSNAME.ora` file.

3. Alternatively, you can enter the value of a **Net Service** directly on a single line, for example:

```
(DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST = myhost)(PORT = 1521))(CONNECT_DATA = (SERVICE_NAME = ora)))
```

   This is useful if you are installing manually, have no `TNSNAME.ora` file but do have more to configure than just a host, port and service.

4. Alternatively, you can define a variable in PowerShell and give it a nicely formatted connection string as a value, for example:

```
$MyDatabaseServer ="(DESCRIPTION =
                      (ADDRESS = (PROTOCOL = TCP)(HOST = myhost)(PORT = 1521))
                      (LOAD_BALANCE = YES)
                      (CONNECT_DATA =
                       (SERVER = DEDICATED)
                       (SERVICE_NAME = ORA)
                      )
                    )"
```

You can then call your PowerShell script with a parameter called `DatabaseServer`, set to this variable:

```
& ‘.\somescript.ps1’ –DatabaseServer $MyDatabaseServer
```

This is useful is you want to perform one or more unattended installations and have no `TNSNAME.ora` file. It also helps to make your connection string easier to read.

Chapter 3 Hardware prerequisites

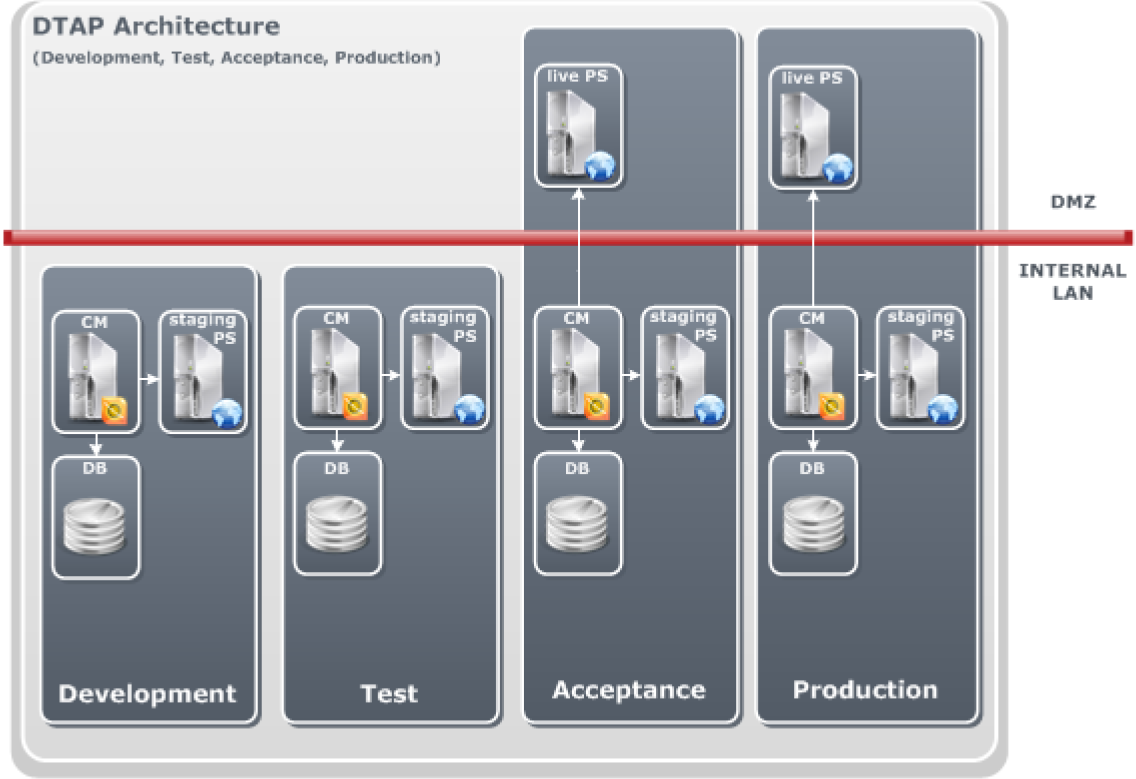
Guidelines on selecting the correct hardware for your SDL Tridion implementation. The platform supports basic, single-machine implementations and elaborate, scaled-out implementations, each with different hardware requirements.

3.1 DTAP infrastructure

SDL recommends running multiple instances of SDL Tridion in parallel to manage the development process and clearly separate the project implementation phases and use Content Porter to port data from on SD: Tridion environment to another. **D**evelopment, **T**est, **A**cceptance and **P**roduction (DTAP) is the recommended infrastructure for an SDL Tridion implementation.

DTAP enables different users to manage different types of content on different Content Manager servers: the creation and management of content is distributed over several machines to prevent different types of content from getting in each other's way. You then use Content Porter to port data from one SDL Tridion environment to another.

The following diagram shows a typical hardware setup for a DTAP environment: a single-machine setup for Development and Test, and a multiple-machine setup for Acceptance and Production. Everything except the Acceptance Live Server and Production Live Server lives inside the local network.



3.2 Hardware acquisition plan

Acquiring hardware is a phased activity: buy hardware that meets minimal prerequisites or higher to acquire the Development and Testing machines; implement the product; and use your implementation to measure which (and how much) hardware to acquire your Acceptance and Production machines.

Hardware for Development and Testing environments

When developing or testing your implementation of SDL Tridion, you can follow the guidelines for minimal hardware requirements. It is possible (though not recommended) to develop and test your implementation using a single laptop.

Hardware for Acceptance and Production environments

Without a finished implementation of the product, it is impossible to know which hardware or how much hardware to acquire for your Acceptance and Production environments. This is because your specific implementation choices dramatically affect the performance. As a result, we recommend reserving a project phase in which you simulate a real-live production environment and measure its performance.

3.3 Minimal hardware prerequisites for Development and Testing environments

Both for the Development and the Testing environment, SDL recommends a single-machine setup, with the Content Manager, Content Manager database and Presentation Server on the same machine.

Hardware item	Recommendation
CPU clock rate	2.70 GHz
CPU cores	dual core or quad core
Memory	8 GB RAM
Hard disk	20 GB HDD (30 GB if using a Java Web application server)
Database DBMS	SQL Server (Oracle requires a separate server)

3.4 Performance variables for Acceptance and Production

The performance of the main areas of your SDL Tridion implementation depends on a large number of performance variables, many of which are determined by your implementation choices. This topic enumerates for each performance area which variables you might measure or examine.

Typically, performance bottlenecks in a Production environment occur in one of the following areas:

- Content Manager: users and/or processes interacting with the Content Manager server have to wait too long for a response.
- Publishing throughput: publishing content takes too long.
- Presentation Server serving of Web pages: visitors of the Web site wait too long for a Web page to be loaded in their browser

Content Manager performance variables

As a rule of thumb, automated processes, most notably publishing, often impact scaling decisions more than, say, the number of concurrent users.

Automated processes include:

- Publishing volume, that is, the total combined size of items being submitted for publishing per time unit
- The complexity and speed of templating code
- The complexity and speed of Event System code, if any
- The complexity and speed of GUI extension code, if any
- The complexity of Workflow Process Definitions, if any
- The complexity and speed of other custom code (e.g integration with back-office systems, or on-the-fly data conversion)
- The number of Core Service requests per time unit

Another set of factors is the number of users and the intensity of their interaction:

- The number of concurrent users, defined by the formula `(NAMED USERS * SINGLE USER USAGE PER PERIOD) / OVERALL USAGE PER PERIOD`. For example, 200 different users interacting an average of 30 minutes per week with the system, while the system is in use for 40 hours per week, means that there are (200 * .5)/40 = 2.5 concurrent users.
- The number of search requests per time unit
- The number of item modifications by users per time unit

The third set of factors concerns the size and complexity of your implementation:

- The expected number of Components and Pages
- The number of Organizational Items, and the depth of the structure in which they are organized
- The complexity of your Taxonomies, both the number of Keywords and the depth
- The complexity of your BluePrint structure (including the number of Publications and the BluePrint depth)
- The complexity of your security framework

And fourth, of course, the actual hardware you use affects performance. Note that because Web abd application server tiers are typically easier to scale out than database server tiers, the database tier will more often prove to be the performance bottleneck.

- The number of Content Manager server machines
- The CPU speed, number of CPUs and number of cores of your Content Manager server machine(s)
- The RAM of your Content Manager server machine(s)
- The number of your Content Manager database machines
- The CPU speed, number of CPUs and number of cores of your Content Manager database machine(s)
- The RAM of your Content Manager database machine(s)
- The maximum throughput rate of the hard disk of your Content Manager database machine(s)
- The client/server network latency

Publishing performance variables

- Publishing volume, that is, the total combined size of items being submitted for publishing per time unit
- The complexity and speed of templating code
- The complexity and speed of custom Content Deployer Modules, if any
- The bandwidth of the connection between the Content Manager and Content Delivery systems
- The type and speed of the firewall you use, and how it is configured
- How static or dynamic your Web site is (dynamic Web sites publish faster)

Web content serving performance variables

- The visitor's connection speed
- The number of page views per time unit
- The number of concurrent visitors, that is, the number of active visitor sessions at any one time
- The level of interactivity on Web pages

- The type of Web and application server used
- The amount and level of integration with other applications, such as back-office systems
- The type and speed of the firewall you use, and how it is configured
- The use of SDL Tridion's object caching
- The use of file caching (e.g. proxy caching) - this slows down performance
- The number of Content Delivery Web service requests per time unit
- The number of Content Delivery server machines
- The CPU speed, number of CPUs and number of cores of your Content Delivery server machine(s)
- The RAM of your Content Delivery server machine(s)
- The number of your Content Delivery database machines
- The CPU speed, number of CPUs and number of cores of your Content Delivery database machine(s)
- The RAM of your Content Delivery database machine(s)
- The maximum throughput rate of the hard disk of your Content Delivery database machine(s)
- The connection speed between your various Presentation Servers
- How static or dynamic your Web site is (static Web sites serve content faster)

3.5 Hard disk size variables for Acceptance and Production environments

Both your Content Manager and Content Delivery database servers need a hard disk that can accommodate the managed content size or published content size, respectively.

Content Manager database

The Content Manager database in a Production environment requires a hard disk size of 30 GB + (3 * size of managed content in bytes).

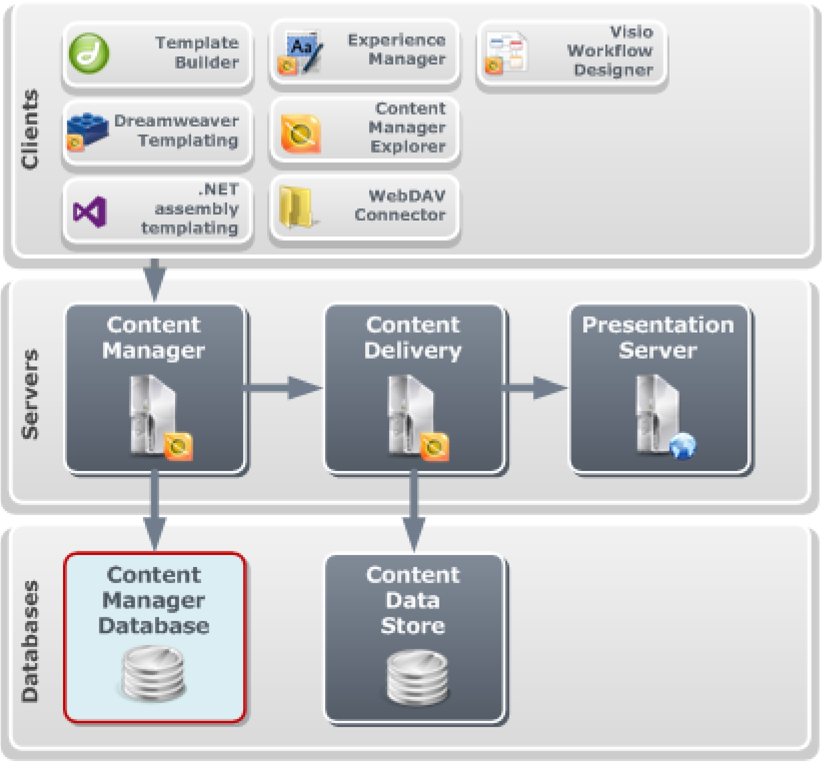
Content Delivery database

The Content Delivery database in a Production environment requires one of the following hard disk sizes of 30 GB + (3 * size of published content in bytes). Note that your published content size is difficult to predict because it depends on various aspects of your system:

- Amount of static content vs dynamic content
- How much of the managed content is rendered as published content (determined by your templating code and the choice of Template)
- Amount of Web sites you publish to

Chapter 4 Creating the Content Manager database

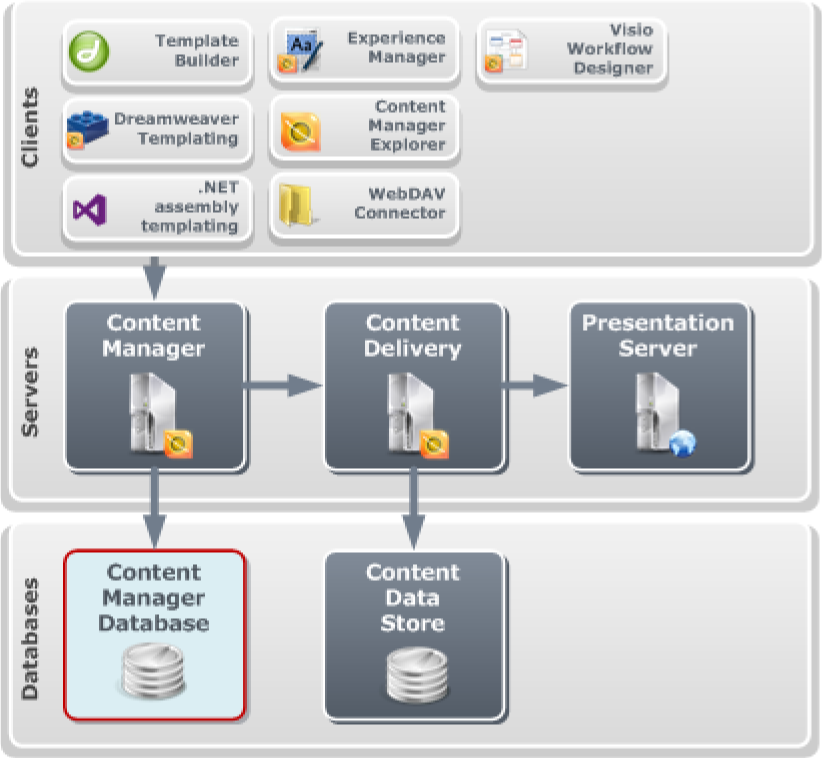
The Content Manager requires a database. Create the Content Manager database by running a PowerShell script.



4.1 Content Manager database prerequisites

The Content Manager requires an Oracle or Microsoft SQL Server database. This topic lists the required and supported software for the Content Manager database.

Note: The Content Manager database server runs on any operating system that supports the specific DBMS you choose to use. Refer to your database vendor documentation to learn which operating systems are supported.



Recommended prerequisites

SDL recommends one of the following database servers:

- Oracle Server 11.2.0.3 with one of the following patches applied:
  - For Linux/UNIX systems: PSU 11.2.0.3.2
  - For Windows systems: 11.2.0.3 Patch 2 or higher (for Windows)
- Microsoft SQL Server 2012 SP1

Supported Microsoft SQL Server databases

Content Manager supports the following Microsoft SQL Server databases.

Supported:

- SQL Server 2012 SP1
- SQL Server 2008 R2 SP2

Deprecated: None

Supported Oracle databases

Content Manager supports the following Oracle databases.

Supported:

- Oracle Database 11*g* Release 2 patch set 11.2.0.3 with one of the following patches applied:
  - For Linux/UNIX systems: PSU 11.2.0.3.2
  - For Windows systems: 11.2.0.3 Patch 2 or higher (for Windows)
- Oracle Database 10*g* Release 2, patch set 10.2.0.5

Deprecated: None

PowerShell

To install SDL Tridion databases on a Windows machine requires Windows PowerShell 3.0 be installed on that machine.

.NET

To install the Content Manager database on a Windows machine, Microsoft .NET Framework 4.0 or higher must be installed on that machine.

4.2 Creating a system user (MTSUser) for Content Manager

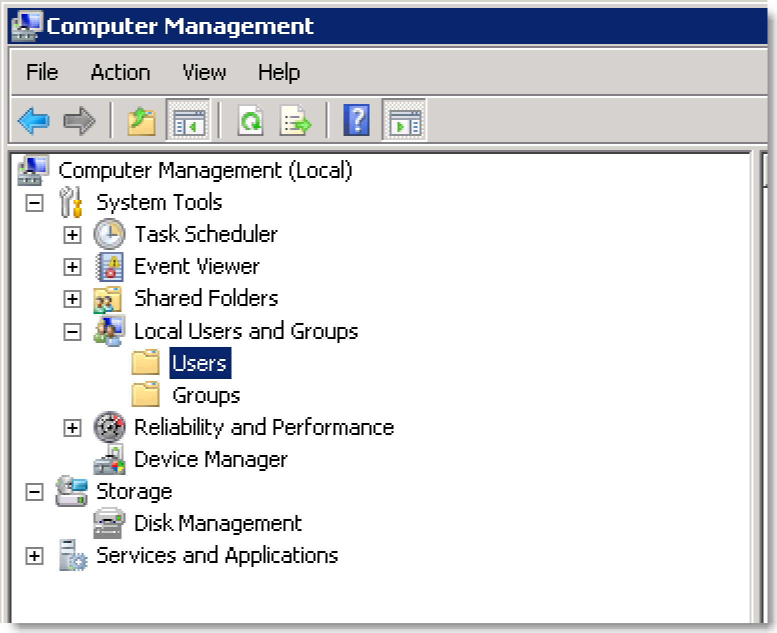
The Content Manager requires a separate Windows domain user, known as the MTS user, to access the Content Manager database, perform logging, run the Content Manager event system, and so on. Create this user before you install the Content Manager database and Content Manager server. After installation, you need to configure access permissions for this user.

Requirements

Create and configure an `MTSUser` account on the domain controller machine of your Windows domain.

Steps to execute

1. In your Windows Control Panel, access **Administrative Tools** > **Computer Management** .



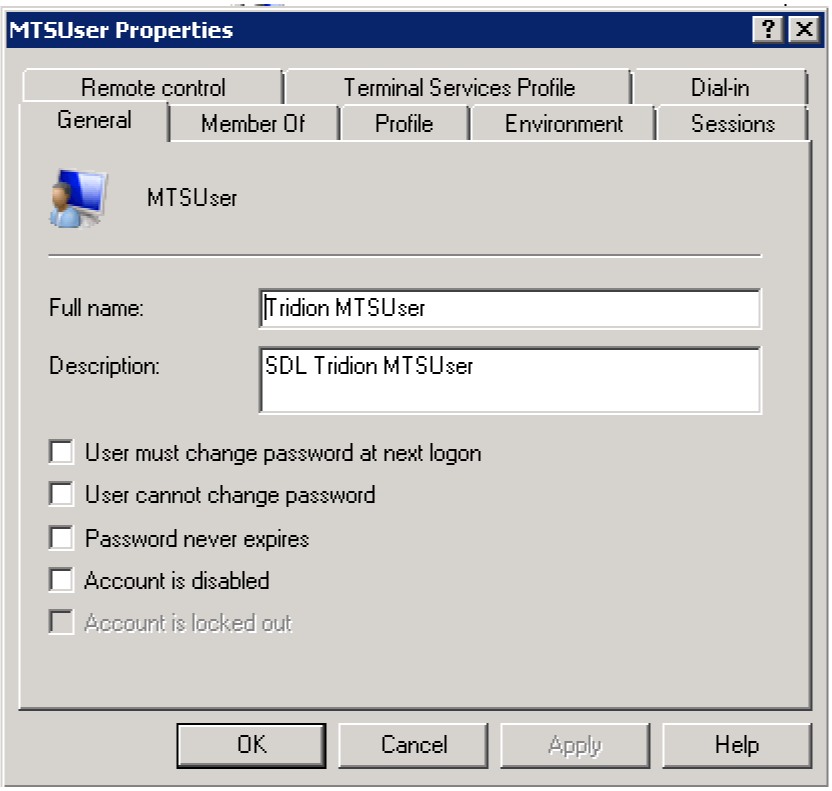
2. Navigate to **System Tools** > **Local Users and Groups** > **Users** .
3. Select **Users** and choose **New User** from the context menu. Specify user details and click **Create**:

   a. **User name**—`MTSUser`

   b. **Full name**—`MTSUser`

   c. **Description**—`SDL Tridion MTSUser`

4. Select the `MTSUser` you just created and select **Properties** from the context menu. A new window opens, showing the **General** tab of the user's properties:



Full name: Tridion MTSUser

Description: SDL Tridion MTSUser

□ User must change password at next logon

□ User cannot change password

□ Password never expires

□ Account is disabled

□ Account is locked out

5. Make sure that the two options **Account is disabled** and **Account is locked out** are both not selected.

6. Click **OK** to close the **MTSUser Properties** window.

7. Close the Computer Management screen.

Result

You have now created the MTSUser account. If, at any later time, you change the password, update the password in your search settings, whenever you next optimize your search collection, and on your scaled-out search servers (if any).

Next steps

Store the MTS user credentials in a safe location because you will need to enter the credentials you specified here when you install the Content Manager server.

4.3 Configuring Microsoft SQL Server database server

A qualified Microsoft SQL Server database administrator needs to perform the following configurations to make your SQL Server database server work with Content Manager:

- Install the Microsoft SQL Server database server with default options, with the following exception: SQL and Windows authentication must be enabled.
- Set the database to 'autogrowth'. Ensure that the database is set to grow in fixed-size increments of about 50 MB.

  Setting autogrowth to a percentage may result in timeouts on large database. For example, a 10 GB database with autogrowth set to 10% must allocate and initialize 1 GB, and execute the initializing statement, all within 30 seconds.
- Use TCP/IP for best performance of distributed transactions. Configure this setting in the Microsoft SQL Server Client Network Utility.
- Ensure that the Microsoft SQL Server instance collation is case-insensitive.

Note: For more information about Microsoft SQL Server collation settings, refer to Microsoft SQL Server Books Online that are available from the Microsoft Web site.

4.4 Configuring Oracle database server

You need to perform the following configuration on an Oracle database server before installing your Content Manager database.

Distributed Transaction Processing-enabled Oracle service

If you use Real Application Clusters (RAC) with your Oracle database, your Oracle database servers must make a Distributed Transaction Processing-enabled (DTP-enabled) Oracle service available for Content Manager to connect to. A Content Manager instance must connect to a DTP service that is available on one and only one Oracle RAC instance.

National language settings

The character set for the Oracle database must be one of the following:

- `UTF-8`
- AL32UTF8

Initialization Parameters

The following table lists the recommended settings for Oracle initialization parameters. Set these parameters in the `init.ora` or `spfile` of the Oracle Database instance used for the Content Manager. These recommendations are based on approximately 500 concurrent database connections. The recommended values apply for both Oracle Database 10*g* and 11*g*.

Parameter	Recommended value
QUERY_REWRITE_ENABLED	true (default)
QUERY_REWRITE_INTEGRITY	trusted
OPTIMIZER_DYNAMIC_SAMPLING	2 (default)
NLS_LENGTH_SEMANTICS	BYTE

Tablespaces

Database objects are created in a number of tablespaces. Oracle Databases can use either dictionary-managed tablespaces or locally managed tablespaces. It is recommended to use locally managed tablespaces.

The following table presents the minimal properties of these tablespaces when using locally managed tablespaces. Note that you may need to modify these values depending on sizing.

Tablespace	Size	Uniform allocation size
USERS	4096M	4096M
INDX	2048M	2048M
LOBS	4096M	4096M
SESSION_DATA	1024M	1024M

The Content Manager installer expects a temporary tablespace called TEMP. This tablespace must be available during Content Manager installation. The TEMP tablespace is used, in part, for temporary LOBs. If you have many concurrent users, you may need to increase your TEMP tablespace up to 2 GB.

Database schemas

By default, the Content Manager uses a database schemas called TCMDBUSER. You create this schemas when you run the PowerShell script. You can choose a different name for this schema.

When you create a new database, the database administrator you specify when you run the PowerShell script grants the following roles and privileges:

- CONNECT role
- CREATE TABLE
- CREATE PROCEDURE
- CREATE TRIGGER
- CREATE SEQUENCE
- EXECUTE ON DBMS_UTILITY
- UNLIMITED TABLESPACE
- SELECT ANY DICTIONARY (to access SYS views such as v$parameter)

The user must also have execute permission on the dbms_utility package. The PowerShell script sets these permissions during the installation process.

After the database has been created, the majority of these privileges are revoked. The following roles and privileges remain:

- Required roles for normal use:
  - `CONNECT`
- Required privileges for normal use:
  - `GRANT UNLIMITED TABLESPACE`
  - `SELECT ANY DICTIONARY`

Performance considerations

The Content Manager uses a cost-based optimization. You must therefore analyze tables and indexes regularly. To do this, use the packaged stored procedure:

```
dbms_stats.gather_schema_stats('[schema]', cascade=true)
```

where `[schema]` is the name of the schema that contains the tables and indexes to analyze.

Disk configuration

The Content Manager generates a lot of redo log data. To avoid contention on the disk where the redo log data is written, place the redo log file on its own disk or on a disk set where no other database files are placed. Make the redo log file large, because small redo log files result in frequent log switches that can degrade database performance.

The Content Manager typically uses different tablespaces for data (`USERS`), temporary data (`TEMP`), indexes (`INDX`) and binary data (`LOBS`) which can be specified during database installation when you run the PowerShell script.

You can optimize performance if each of these tablespaces is located on a separate physical disk or disk set.

Net configuration

The server running the Content Manager and the Oracle Database client must be able to connect to the Oracle database server using Local Net Service Naming.

To connect to the database server, use the Net Manager or the Net Configuration Assistant to create an entry for the database server in the local `TNSNAME.ora` file. Alternatively, you can specify the properties of your connection in PowerShell when you install or upgrade the database.

Note: Do not use Global Naming. The Oracle Services for MTS do not support Global Naming correctly, which can lead to unpredictable behavior.

Troubleshooting ORAMTS

If you receive the error, "New transaction cannot enlist in the specified transaction coordinator", do the following:

a. Check the following registry key, which should contain the computer name of the management system:

```
HKLM\SOFTWARE\ORACLE\OracleMTSRecoveryService\Protid_0\Host
```

b. Check if the `OracleMTSRecoveryService` is started.

c. Check for any pending distributed transactions. `OracleMTSRecoveryService` cannot start if this is the case.

To check this, query:

```
DBA_2PC_PENDING: SELECT LOCAL_TRAN_ID FROM DBA_2PC_PENDING;
```

If there is a pending transaction, do a rollback:

```
ROLLBACK FORCE local_tran_id from previous query; EXEC
DBMS_TRANSACTION.PURGE_LOST_DB_ENTRY (local_tran_id FROM
previous query)
```

Microsoft Transaction Server transaction recovery

To learn about Managing Recovery Scenarios related to scheduling automatic Microsoft Transaction Server Transaction recovery, consult your Oracle product documentation, specifically, one of the following resources:

Oracle 11

http://download.oracle.com/docs/cd/B28359_01/win.111/b28377/recovery.htm#CBADHGHH

Oracle 10

http://download.oracle.com/docs/cd/B19306_01/win.102/b14320/recovery.htm#NTMTS003

Creating an SQL SDL Tridion administrator user

SDL Tridion ships with a SQL script that creates an SDL Tridion administrator user that you can use to create a database when you run the PowerShell script.

The privileges and roles created for this user are the minimum requirements for database installation. Before you create a Content Manager database, run the SQL script `CreateTridionSYSUser.sql` (for example from SQL*Plus) located in the directory `Database\Scripts\Oracle\Tools` on your SDL Tridion installation media.

When you use this script, it prompts you for the following information:

Username	The name of the SDL Tridion administrator user that can create the database
Password	The password of this user
Default tablespace	The default tablespace for objects the user creates
Temporary tablespace	The tablespace for the user's temporary segments

Note: You cannot use a user that must connect as `SYSOPER` or `SYSDBA` with the PowerShell script.

4.4.1 Maintaining an Oracle database

The SDL Tridion installation media includes a number of SQL scripts in the `\Database\Scripts\oracle\Tools` folder that you can use to maintain a Content Manager Oracle database.

These scripts can be run in SQL*Plus. The user running them must be the Content Manager database user because the scripts operate on the Oracle Schema to which you logged in. Run these tools at times of low database activity to prevent unreliable results.

Statistics analysis:

Script: `AnalyzeStats.sql`

Regularly running `AnalyzeStats.sql` updates the table, column, and index statistics. The cost-based optimizer uses these statistics to select an optimal query plan. Update statistics to ensure that the optimal query plan is chosen. To see when tables, columns, and index statistics were analyzed last, use `ShowTableStats.sql`, `ShowColumnStats.sql`, and `ShowIndexStats.sql`.

Statistics scripts

Scripts:

- `ShowTableStats.sql`
- `ShowColumnStats.sql`
- `ShowIndexStats.sql`

These scripts provide information about the existing table, column or index statistics, as well as the last time these statistics were analyzed.

Rebuilding the indexes

Script: `RebuildIndexes.sql`

Run `RebuildIndexes.sql` after substantial changes are made to the database. For example, you should run this script after a bulk import, after using the Purge Tool, or after working with the system for an extended period of time. This script rebuilds the Content Manager indexes. It is necessary to rebuild these indexes to remove deleted rows and restructure the index. To determine if it is necessary to rebuild indexes, use `AnalyzeStatsEx.sql`.

Extended index statistics

Scripts:

- `PrepareAnalyzeStatsEx.sql`
- `AnalyzeStatsEx.sql`

Use `PrepareAnalyzeStatsEx.sql` and `AnalyzeStatsEx.sql` to gather extended index statistics. You can use these statistics to retrieve information about the current quality of the index. For example, use it to retrieve the number of deleted rows in the index. `PrepareAnalyzeStatsEx.sql` creates a table called `PLSQL_INDEX_STATS` in the current Oracle Schema, which `AnalyzeStatsEx.sql` then uses to gather index statistics and collect the statistics in a table.

`AnalyzeStatsEx.sql` fills the table but does not perform queries on the table. You can perform queries on the table that will provide valuable information about the index, such as:

```
SELECT * FROM PLSQL_INDEX_STATS
ORDER BY NAME;
SELECT * FROM PLSQL_INDEX_STATS
WHERE DEL_LF_ROWS > 0
ORDER BY DEL_LF_ROWS DESC;
SELECT * FROM PLSQL_INDEX_STATS
WHERE BLKS_GETS_PER_ACCESS > 5
ORDER BY BLKS_GETS_PER_ACCESS DESC;
```

4.5 Creating the Content Manager database

Content Manager Server requires a Content Manager database. Use a PowerShell script to create a Microsoft SQL Server or Oracle database.

Requirements

Software

To install SDL Tridion databases on a Windows machine requires Windows PowerShell 3.0 and Microsoft .NET Framework 4.0 or higher.

Database server

Before running a PowerShell script:

- On **Oracle**—make sure that there are no open database connections and configure the Oracle database server.
- On **Microsoft SQL Server**—configure the Microsoft SQL Server database server.

Database credentials

PowerShell scripts prompt you for the following information:

- DBA and user credentials (unless you created an Oracle SDL Tridion administrator user)
- the server and port number (if non-standard) where your database server resides

Database details

PowerShell scripts prompt you to provide the following information which you will need to store in a safe place along with Database credentials as you will need it when you install and configure the Content Manager and Content Delivery:

- the name of the database
- the name of the database user

PowerShell scripts provide default database names and user names, or you can enter your own.

Users

PowerShell scripts prompt you to provide the following information:

- A system user (MTSUser)
- The name of an administrator user
- Local system account—the default is `NT AUTHORITY\SYSTEM`

Steps to execute

1. Open Windows PowerShell from the Windows Start Menu.

2. In PowerShell, depending on your database navigate to one of the following folders on the installation media:

   - `Database\MSSQL\` or
   - `Database\Oracle\`

3. Type the following command:

   ```
   & '.\Install Content Manager database.ps1'
   ```

4. Follow the instructions in the PowerShell console to install the database.

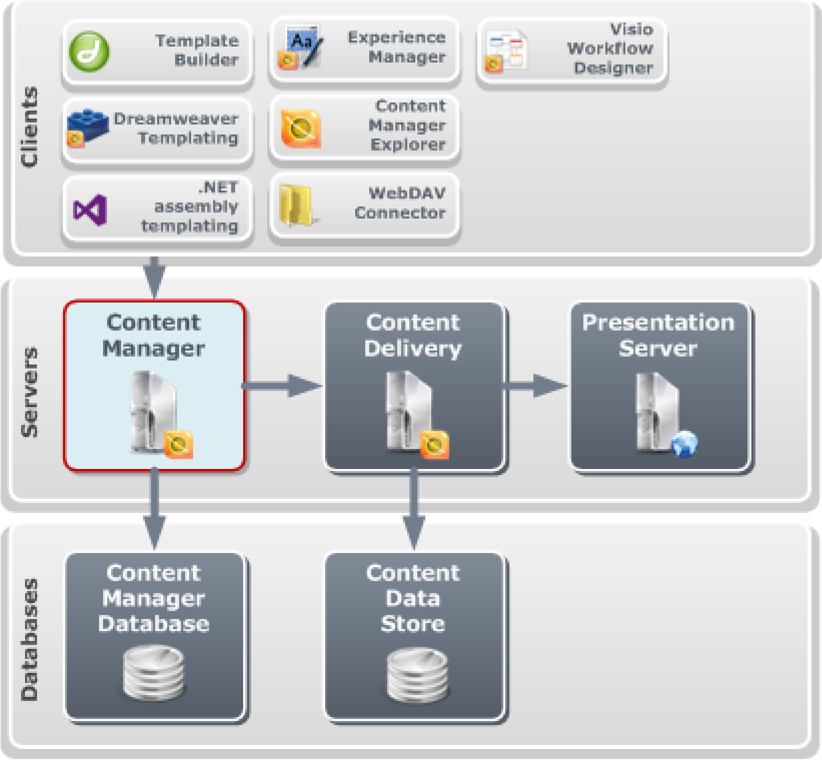
Result

You have created the Content Manager Database.

Chapter 5 Installing the Content Manager server

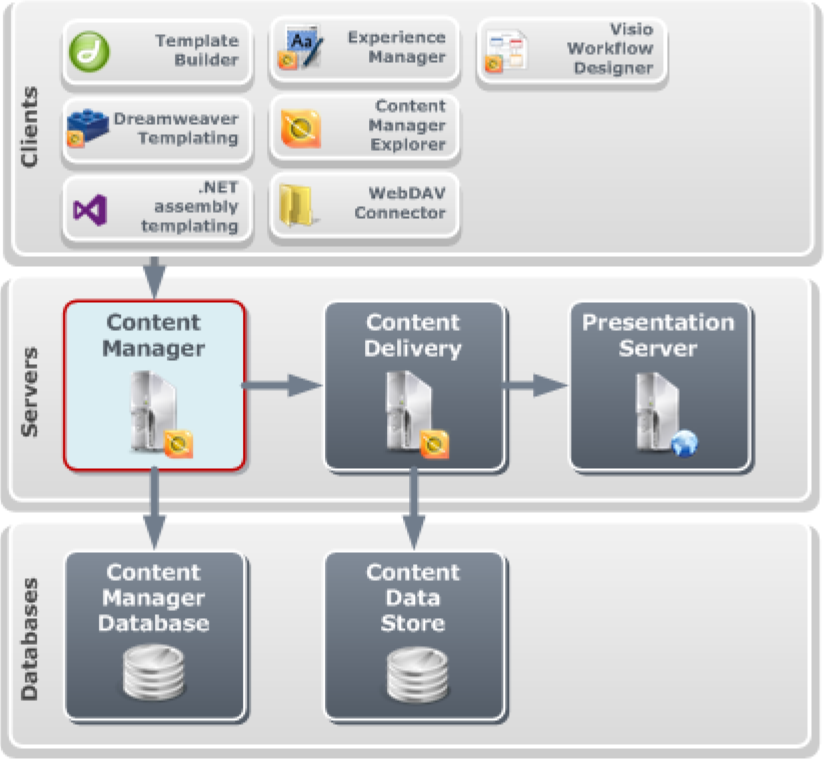
Install the Content Manager Server using the installer so that users can create, manage and assemble the Building Blocks used in Web sites and other channels.

Note: For information on managing Content Manager users, or installing various parts of the Content Manager Server on different machines (scaling) see the implementor's documentation portal.



5.1 Content Manager server prerequisites

This topic lists the required and supported software for the Content Manager server. In most cases, the installer will install a supported version of the required prerequisite if missing.



Recommended prerequisites

SDL recommends the following set of Content Manager server prerequisites:

Prerequisite	Recommended product release
Operating system	Microsoft Windows Server 2012 (x64)
Web and Application server	Microsoft IIS 8
.NET Framework	Microsoft .NET Framework 4.5
Java	Java SE 7.0 (64-bit)
Oracle Database client (if using Oracle)	ODAC 11.2.0.3

Operating system languages

The operating system on which Content Manager server runs must be in one of the following languages:

- US English
- French
- German
- Spanish
- Dutch
- Japanese

Note: Related software must use the same language. For example, using a French operating system implies using a French database.

Operating system versions and editions

The Microsoft Windows operating systems on which Content Manager server are as follows.

Supported:

- Windows Server 2012 (x64)
- Windows Server 2008 R2 SP1 (x64)

Deprecated: None

Note: The SDL Tridion installation media contains a Web page, `\Redistributables\Readme.htm`, with links to the Microsoft download pages for the Service Packs listed in the table. As a general rule, ensure that the latest updates are installed on your Microsoft Windows operating system.

Server Roles

Ensure that the following Server Roles are enabled:

- Web Server
- Application Server

Database clients

Content Manager server requires a database client that connects to one of the supported databases. The database client software required depends on the operating system you use and on the DBMS you use.

Oracle client

If your DBMS is Oracle, you require client Oracle client software (ODAC) version 11.2.0.3.

The exact requirements depend on the architecture of your operating system.

If you are running Content Manager server on a 64-bit system, you require the following:

- The 64-bit edition of the following ODAC software components:
  - Oracle Data Provider for .NET (ODP.NET)
  - Oracle Services for Microsoft Transaction Server (ORAMTS)
- The 32-bit edition of the following ODAC software components:
  - Oracle Data Provider for .NET (ODP.NET)
  - Oracle Provider for OLEDB

The installation topics provide more details on how to install Oracle clients.

Microsoft SQL Server client

Microsoft SQL Server does not have any client requirements.

Web and Application Server

Content Manager server requires one of the following Web and Application servers:

Content Delivery server role that you install as a Web application require a Web server or a Web Application Server. The following table lists the Web Servers and Web Application Servers that Content Delivery supports (your choice of Web and Application server restricts the operating systems you can use for your Content Delivery server roles):

Supported:

- Microsoft IIS 8
- Microsoft IIS 7.5

Deprecated: None

Note the following:

- You need a 32-bit compatible edition of these software components, even if your operating system has a 64-bit architecture.
- You can install the Web and Application server from the Add/Remove Programs screen in your Control Panel, by selecting Add/Remove Windows Components. In the **Add/Remove Windows Components** screen, in the **Details** screen of the item **Application Server**, you must also have the option **Enable network DTC access** checked.
- To use WebDAV, ensure that the Microsoft WebDAV Extension is *disabled* (this is the case by default).

Third-party software components

Content Manager server requires the following third-party applications.

Note: For all of these third-party software components, please note the following:

- Unless indicated otherwise, you need a 32-bit compatible edition of these software components, even if your operating system has a 64-bit architecture.
- The SDL Tridion installation media contains a Web page, `\Redistributables\Readme.htm`, with links to the download pages for the software listed above.

MS XML 4.0 SP3

If you do not install this application yourself, the installer installs it for you.

Microsoft CAPICOM 2.0.0.1 or higher

If you do not install this application yourself, the installer installs version 2.1.0.2 for you.

Only if you install CAPICOM yourself, register the CAPICOM assembly after installation by doing the following:

- Open a command prompt and navigate to the folder in which the CAPICOM assembly is located (this location defaults to `C:\Program Files\Microsoft CAPICOM 2.1.0.2 SDK\Lib\X86\`.
- From the command prompt, execute the following command:

```
regsvr32 capicom.dll
```

- The DLL is now registered.

Microsoft Windows Identity Foundation

You need Microsoft Windows Identity Foundation (x64). If you do not install this application yourself, the installer installs it for you.

Microsoft Visual C++ Redistributable

You need Microsoft Visual C++ 2010 Redistributable SP1 (x64). If you run the installer without having this software component installed, the installer automatically installs it for you.

.NET Framework

Content Manager server requires one of the following .NET Framework releases:

Supported:

- Microsoft .NET Framework version 4.5, CLR 4.5
- Microsoft .NET Framework version 4.0, CLR 4.0

Deprecated: None

Note: You need a 32-bit compatible edition of these software components, even if your operating system has a 64-bit architecture.

Java

Content Manager server requires Java for the following functionality:

- Publishing content from the Content Manager
- Searching content in the Content Manager

For publishing, Content Manager has the following Java prerequisites:

Supported:

- Sun Java Development Kit (JDK), 64-bit edition, version 7.0
- Sun Java Runtime Environment (JRE), 64-bit edition, version 7.0

Deprecated:

- Sun Java Development Kit (JDK), 64-bit edition, version 6.0
- Sun Java Runtime Environment (JRE), 64-bit edition, version 6.0

If you do not have either of these versions installed, the installer installs JRE 7.0, 64-bit edition.

For searching, Content Manager has the following Java prerequisites:

Supported:

- Sun Java Runtime Environment (JRE), 64-bit edition, version 7.0

Deprecated:

- Sun Java Runtime Environment (JRE), 64-bit edition, version 6.0

If you do not have either of these versions installed, the installer installs JRE 7.0, 64-bit edition.

User directories

The Content Manager server supports the following user directories (it does not require either):

User directory type	Security platform
NTLM	Microsoft Active Directory Service
LDAP	All LDAP-compliant security platforms Note: LDAP is a general standard supported by, for example, Sun ONE Directory Server and Novell E-Directory.

5.2 Installing the Content Manager server

Install Content Manager Server by running the installer.

Requirements

Before you install the Content Manager server:

Create the Content Manager database

Create the Content Manager database.

Check prerequisites

Only install if all software requirements for the Content Manager server are satisfied.

Also, SDL strongly recommends installing the Content Manager server software on a machine that is in a Windows domain, (that is, not on a standalone machine or a machine in a Workgroup). Without a domain, managing the MTS user and other user accounts becomes much more difficult. The installation procedure assumes that both the Content Manager server machine and its users are in a domain.

Log in as a member of the local administrator group

Only install if you are logged in as a member of the local administrator group.

You cannot install if you are logged in as a non-administrator: right-click the shortcut and select **Run as Administrator**. You can, however, log in as an administrator-level user or double-click the shortcut. In the latter case, you will be prompted for administrator-level user credentials during the installation.

Decide if you want to install legacy features

Only if you are an existing customer performing a fresh install with the intention of implementing old SDL Tridion features that are now deprecated, start the installer executable from a command line and add the key-value pair `LEGACY_VISIBLE=TRUE`. This lets you select deprecated features (shown in italics) in the **Advanced** screen during installation.

Configure a host header

If you intend to use a host header for HTTP communication with the Content Manager server, navigate to the `system32\drivers\etc\` subfolder of your Windows folder and in the `hosts` file located there, add an entry that maps the IP address 127.0.0.1 to a host name of your choice.

Licenses

You can use the installer to install your Content Manager server licenses. Your license agreement with SDL determines the modules and functionality that you can use and access. Normally, SDL Customer Support gives you all SDL Tridion licenses in the form of two license files:

- `license.xml`—contains license keys for the various Content Manager server software components
- `cd_licenses.xml`—contains license keys for publishing content to the Content Delivery server using one or more transport protocols.

Information required by the installer

You need to have the following information ready beause the Content Manager installer will prompt you for it:

- Database and database user credentials
- The name of the Content Manager database you created
- The server on which the Content Manager database resides
- The user name and password of the MTS user you created
- The domain of the server on which you are installing the Content Manager server
- The host header, if you configured one.
- The names and locations of the license files. You can also install one or both of these licenses manually later.

Steps to execute

1. Access the SDL Tridion installation media.
2. Navigate to the `Content Manager\` directory.
3. Double-click the executable `SDLTridion2013CM.exe`.
4. Follow the instructions on the screen:
   - You can select individual software components to install by choosing the **Advanced** option (by default, the installer installs all software components).
   - You can install software components you choose not to install by running the installer again and selecting the **Modify** option.
5. After the installer finishes, if the installer does not prompt you to restart your machine, perform a manual reboot.

Result

The installer has created a log file in subdirectory `SDL\Tridion\Logs\` of the path indicated by the `%PROGRAMDATA%` environment variable (defaults to `c:\ProgramData\`).

5.3 Software installed by Content Manager installer

A list of software components that you can install using the Content Manager installer. By default, the Content Manager installer installs all these software components, except for legacy features.

Monitoring Server

A module that lets you monitor SDL Tridion Services.

Search Host

A module supporting the outscaling of the Search Index Server so that you can set up a Search Host Master Server for indexing (and optionally search) and any number of SDL Tridion Search Host Slave Servers, only for search.

Content Manager

The SDL Tridion Content Manager server-side application.

Publisher Service

A module that publishes all content from SDL Tridion Content Manager.

Search Indexing Server

A module that indexes content from SDL Tridion Content Manager.

Workflow Agent Service

A module that executes scripts defined in SDL Tridion Content Manager workflow processes.

Tridion Content Manager Batch Processor

A scalable service for processing multiple items in batch.

Interfaces

Various user interfaces to SDL Tridion Content Manager, such as the browser-based GUI, Content Manager Explorer, and a WebDAV interface.

IISHost

A module that is required for the Core Service, a Web service used for integration of your external application with the Content Manager core.

Visio Workflow Server

A server-side module that lets you define workflow processes using SDL Tridion Workflow Designer.

WebDAV Server

A module that lets you access SDL Tridion Content Manager from any WebDAV-compliant client application or operating system.

Content Manager Monitoring Service

A module that provides monitoring of SDL Tridion Services.

Templating

The Modular Templating Framework allows you to create C#, .NET assembly, and Dreamweaver Template Building Blocks (TBBs). A TBB performs a modular task.

Template Builder

An interface for assembling Template Building Blocks together into a Compound Template.

Content Manager Explorer

The browser-based content management interface for SDL Tridion Content Manager.

- **Device Preview**—Device Preview allows you to preview and edit content in Experience Manager as it is displayed in a handheld device such as a mobile, smartphone, or tablet.
- **Safeguard**—Safeguard allows you to analyze a Web page in Experience Manager to ensure consistency with branding and standards guidelines.

Experience Manager

The browser-based content authoring interface for SDL Tridion Content Manager.

Documentation

Content Manager Explorer Help.

SDL Tridion Tools & Development

A collection of tools and code samples.

Event System Source Code

Sample source code for an Event System implementation.

User Generated Content

A module that allows visitors to your Web site to rate your content and leave comments and then manage these ratings and comments.

Audience Manager

A module for gathering information about Contacts and audiences such as interests, preferences and characteristic.

GUI Extensions

Adds Audience Manager and Outbound E-mail to the Content Manager Explorer.

Synchronization service

Synchronizes Contact information and Segments between the Audience Manager database and the Subscription database on the presentation side.

Outbound E-mail

A module for managing e-mail campaigns (Mailings and Distribution Lists).

Modular Templating Support

Outbound E-mail provides predefined Template Building Blocks that you can use to develop Compound Templates used when sending Mailings.

Script Extension

The Outbound E-mail Script Extension allows you to create Page Templates or Component Templates using VBScript or Jscript (supported for backwards compatibility).

Queue Triggered Mailings service

The service is responsible for handling triggered mailings.

Collect Tracking Data service

The service retrieves tracking information from the Tracking database on the presentation side.

Bounce Processor service

The service keeps track of bounced e-mails and updates the mailing statistics by periodically polling a preconfigured POP3 or IMAP mailbox account for bounced e-mails.

Mailer service

The service sends out the mailings to the SMTP Gateway server(s).

Translation Manager

A module that allows you to send content stored in the Content Manager of SDL Tridion for translation to a translation management system.

CME Translation Manager Extensions

Adds Translation Manager to the Content Manager Explorer.

Translation Manager Service

Synchronizes data (translation jobs and items in translations jobs) between the translation management system and Content Manager.

External Content Library

The External Content Library is a module for exposing multimedia contained in an external system in SDL Tridion so that you can use the media in SDL Tridion-driven Web sites.

Legacy features

If you ran the installer from the command line with `LEGACY_VISIBLE=TRUE`, you may have also selected to install the following:

Content Manager Legacy Feature

General software to support old implementations of SDL Tridion.

Business Connector

Deprecated software to connect to the Content Manager from a third-party application.

Tools & Development (includes Legacy Event System Source Code)

Tools and sample code for developing deprecated features.

5.4 Installing the Content Manager server license file

You can install the Content Manager license file by placing `license.xml` in the default location, or you can rename the file and/or change its location and specify these custom settings in the SDL Tridion MMC Snap-in.

Requirements

You only need to install the Content Manager server license file if you did not specify a license file during installation.

Context

A Content Manager license file is specific to the machine on which you installed the Content Manager server. You cannot use this license file on any other computer.

Steps to execute

1. To install the Content Manager server license file, do one of the following:
   - Add the license file to the `bin\` subdirectory of the Content Manager root location (defaults to `C:\Program Files (x86)\Tridion\`) under the file name `license.xml`.
   - Add the license file to another location and/or change its name.
2. If you changed the file name, the file location, or both, modify the default information in the SDL Tridion MMC Snap-in.
3. Start the MMC Snap-in by selecting **Programs** > **SDL Tridion** > **SDL Tridion Content Manager configuration** in the Microsoft Windows Start menu.
4. Select the **SDL Tridion Content Manager** > **General settings** node.
5. Double-click **License file name or path** and modify the settings as needed.
6. Restart your machine for the licensing to take effect.

5.5 Installing the Content Delivery license file on Content Manager

You need a Content Delivery license file on your Content Manager server to enable publishing; without the license file, users cannot publish content.

Requirements

You only need to install the Content Manager server license file if you did not specify a license file during installation.

Steps to execute

1. To install the Content Manager server license file, do one of the following:
   - Add the license file to the config\ subdirectory of the Content Manager root location (defaults to `C:\Program Files (x86)\Tridion\`) under the file name `cd_licenses.xml`.
   - Add the license file to another location and/or change its name.

2. If you changed the file name, the file location, or both, modify the default information in the Transport Service configuration file. This file is called `cd_transport_config.xml` and is located in the `config\` subdirectory of root . Open this file in a plain-text editor and search for the string "`License`". The editor finds the following XML fragment:

```
<License Location="c:\Program Files\Tridion\config\cd_transport_license.xml" />
</TransportService>
```

3. Edit the value of the `Location` attribute (the string enclosed in quotation marks) so that it points to the correct location and file name.

4. Access Windows **Services** and restart the Windows service called **Tridion Content Distributor Transport Service** for the licensing to take effect.

5.6 Configuring Microsoft Windows for the Content Manager server

You need to perform a number of configuration tasks relating to Windows permissions and security to ensure the correct functioning of the Content Manager.

5.6.1 Configuring access permissions for the MTSUser

You need to grant the `MTSUser` under which the Content Manager is running Read/Write access rights to the `c:\windows\temp` folder.

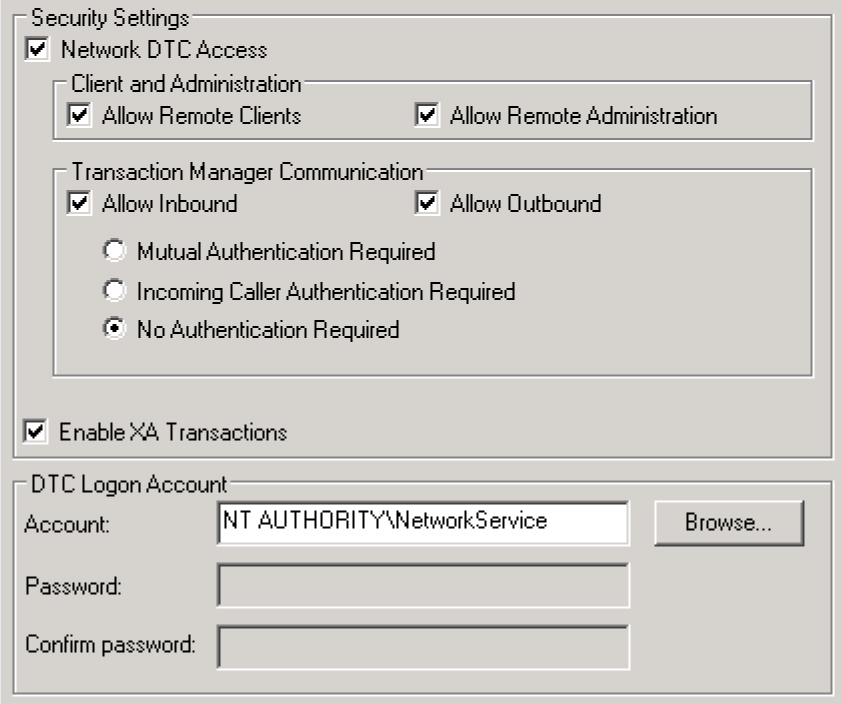
5.6.2 Changing DTC security settings

You need to change the DTC security settings on your Content Manager server (and if necessary on your Content Manager database server) otherwise the machine will experience problems receiving requests (reporting for example 'Transaction cannot enlist' in the Windows Event Log).

Steps to execute

1. Access your Content Manager server machine.
2. Do one of the following:
   - In Windows 2008 R2, from the Windows **Control Panel**, access **Administrative Tools** > **Component Services** .
   - In Windows 2012, press **Windows+F** to bring up the search screen, select **Apps** and enter **Component Services**. Then select **Component Services** on the left.

   The **Component Services** dialog opens.
3. Select **Component Services** > **My Computer** > **Distributed Transaction Coordinator** > **Local DTC** and choose **Properties** from the context menu. A window called **Local DTC Properties** opens. Select the **Security** tab.
4. To fix the DTC security settings:
   - In the **Transaction Manager Communication** area, select **No Authentication Required** or, if your communication between Content Manager server and database server requires authentication, ensure that each of the two machines can resolve or find the other machine's NetBios name.
   - If your database is Oracle, in the **Security Settings** area, select **Enable XA Transactions**.
   - If your database is SQL Server, ensure that the check boxes **Network DTC Access**, **Allow Inbound** and **Allow Outbound** are selected.



Security Settings

☑ Network DTC Access

Client and Administration

☑ Allow Remote Clients ☑ Allow Remote Administration

Transaction Manager Communication

☑ Allow Inbound ☑ Allow Outbound

○ Mutual Authentication Required

○ Incoming Caller Authentication Required

● No Authentication Required

☑ Enable XA Transactions

DTC Logon Account

Account: NT AUTHORITY\NetworkService

Password:

Confirm password:

5. Click **OK** to close the dialogs and close the **Component Services** window.

6. Additionally, ensure that the following ports are open on the Content Manager server:

   - port 135: this port is used by MSDTC for RPCs (Remote Procedure Calls).
   - ports 5000-5500: MSDTC uses a port in this range as a randomly allocated high TCP port.

   Consult with the security administrator in your organization to find out if these ports are open, and if not, how to open them.

7. If your Content Manager database server runs on a Windows machine, access that machine, navigate to the same DTC security configuration screen described in step 2, and apply the same changes as on the Content Manager server: that is, for Oracle, ensure that **Enable XA Transactions** is selected, or for SQL Server, ensure that **Network DTC Access**, **Allow Inbound** and **Allow Outbound** are selected.

8. Alternatively, if your Content Manager database server is Oracle and runs on a non-Windows machine, enable XA transactions for Oracle.

9. Also, open the same ports as on the Content Manager server: port 135 and ports 5000-5500. Consult with the security administrator in your organization to find out if these ports are open, and if not, how to open them.

Next steps

For more detailed information on this configuration, refer to Microsoft Knowledge Base article 250367, found online at http://support.microsoft.com/kb/250367.

5.6.3 Enabling ASP.NET

The Content Manager server requires that ASP.NET functionality is enabled.

Steps to execute

1. If your Content Manager server runs on **Windows 2008 R2 (IIS 7.5)**:

    a. Access your **Server Manager**.

    b. In the dialog that opens, in the tree on the left, navigate to the node **Roles** > **Web server (IIS)** .

    c. In the content area on the right, scroll down to the area called **Role Services** and check that the item **Web Server** > **Application Development** > **ASP.NET** is listed as **Installed**.

    d. If this Role Service is not installed, install it now.

2. Alternatively, if your Content Manager server runs on **Windows 2012 (IIS 8)**:

    a. Start **Server Manager** and select **Dashboard** on the left.

    b. In the dialog that opens, select **Add Roles and Features** to open the **Add Roles and Features Wizard**.

    c. Step through the wizard, selecting this server, until you reach the **Server Roles** step.

    d. In the list of **Roles**, expand the options to navigate to **Web Server (IIS)** > **Web Server** > **Application Development** .

    e. Depending on the version of ASP.NET you use, select **ASP.NET 3.5** or **ASP.NET 4.5**.

       The wizard suggests further features to installed as required by your selection.

    f. Continue on to the **Confirmation** step and select **Install** to install these features.

5.7 Configuring the Content Manager server

After you have installed the Content Manager you need to configure various parts of the Content Manager to get it up and running.

Note: The Content Manager Explorer and the Monitoring Server require no server-side configuration.

5.7.1 Setting up 32-bit Templating on the Content Manager server

Only if you intend to create and use in-process 32-bit COM server DLLs as Template Building Blocks in a Template, manually replace your 64-bit Publisher Windows service with its 32-bit equivalent, and set an attribute in your Content Manager configuration file.

Steps to execute

1. Access your list of Windows **Services**.
2. In the list of Services, find the item called **Tridion Content Manager Publisher**, select it and stop this service.
3. Without closing the **Services** dialog, open a command prompt and enter the following:

   ```
   sc config TcmPublisher binPath="C:\Program Files
       (x86)\Tridion\bin\TcmPublisher_x86.exe"
   ```

   where `C:\Program Files (x86)\Tridion\` can be replaced by the actual root location of the Content Manager, if you specified an alternative path during installation.
4. Close the command window.
5. In the **Services** dialog, restart the **Tridion Content Manager Publisher** Windows service.
6. Access the `config\` subdirectory of the Content Manager root location (defaults to `C:\Program Files (x86)\Tridion\`).
7. Open the file `Tridion.ContentManager.config` in this location in a plain-text or XML editor.
8. Find the `tridion.templating` element. This element has a subelement called `debugging`. Ensure that it has an attribute called `force32Bit` set to `true`.
9. Then save and close `Tridion.ContentManager.config`.

Result

You have now configured a 32-bit Publisher service and can use 32-bit DLLs as Template Building Blocks.

5.7.2 Setting up Oracle client software

If your Content Manager database is Oracle, you need to install and configure Oracle client software on the machine on which you installed Content Manager server.

Steps to execute

1. Access the SDL Tridion installation media and in the `Redistributables\` folder, open the Web page `Readme.htm`. The page contains a link to a page where you can download the Oracle client software. Download and unpack it to a location of your choice, then start the 64-bit Oracle client installer.

   After a few minutes of initializing, the installer starts, showing its initial screen.

2. Select **Custom** as your installation type and click **Next**.

   The **Select Product Languages** step appears.

3. Select the languages in which you want the client software to be available and click **Next**.

   The **Specify Installation Location** step appears.

4. Specify an **Oracle Base** location. Because you will be installing the 32-bit client as well, provide a specific name for the 64-bit client under **Software Location** by appending, say, `client_64` to the path you would normally use. Then click **Next**.

   The **Available Product Components** step appears.

5. In this screen, select the following items:

   - SQL*Plus
   - Oracle Net
   - Oracle Services for Microsoft Transaction Server
   - Oracle Data Provider for .NET

   and click **Next**.

   The **Oracle Services for Microsoft Transaction Services** step appears.

6. Leave the port number as it is, and click **Next**.

   The **Summary** screen appears.

7. Click **Finish**.

   The installer installs the client software and reports success.

8. Click **Close** to exit the installer.

   You have now installed the 64-bit client.

9. Find the 32-bit client installer in the software package you downloaded and run it.

   The initial **Select Installation Type** screen appears.

10. Select **Custom** and click **Next**.

    The **Select Product Languages** step appears.

11. Select the languages in which you want the client software to be available and click **Next**.

    The **Specify installation location** step appears.

12. Specify an **Oracle Base** location. Provide a specific name for the 32-bit client under **Software Location** by appending, say, `client_32` to the path you would normally use. Then click **Next**.

    The **Available Product Components** step appears.

13. Select the following items:

    - SQL*Plus
    - Oracle Net
    - Oracle Services for Microsoft Transaction Server
    - Oracle Provider for OLE DB
    - Oracle Data Provider for .NET

    and click **Next** to continue.

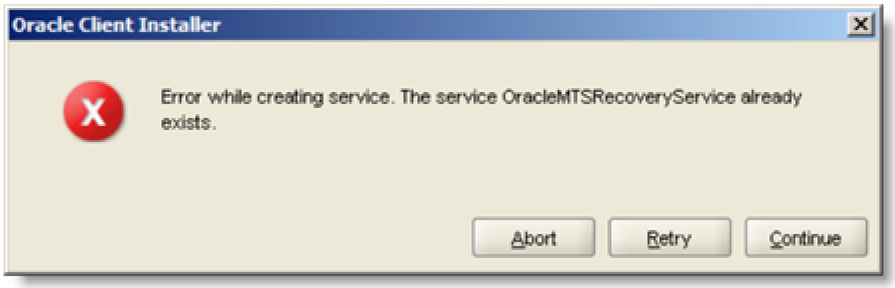
    The **Oracle Services for Microsoft Transaction Services** step appears.

14. Leave the port number as it is and click **Next**.

    The **Summary** step appears.

15. Click **Finish** to start installing the client.

    You now see an error message.

    Oracle Client Installer

    Error while creating service. The service OracleMTSRecoveryService already exists.

    Ignore this message and click **Continue** to close this popup.

    The installer installs the client software and reports success.

16. Click **Close** to close the installer.

    You have now successfully installed all client software for Oracle.

17. Configure and test your installed client or clients. Start by creating a plain text file with the following content:

```
# TNSNAMES.ORA Network Configuration file
```

```
myoracleserver=
 (DESCRIPTION =
   (ADDRESS_LIST =
     (ADDRESS =
       (PROTOCOL = TCP)
       (HOST = myoracleserver)
       (PORT = 1521)
     )
     (CONNECT_DATA =
       (SERVICE_NAME = myoracleserver)
     )
   )
 )
```

where `myoracleserver` is the name of both your Oracle server and database instance.

Save this file as `TNSNAMES.ORA` in the Oracle home directories that you specified during installation.

18. Open a command prompt and access the `bin\` subdirectory of the Oracle home directory or home directories and execute the following command:

    ```
    tnsping.exe myoracleserver
    ```

    where `myoracleserver` is the same as before.

    The command prompt reports back with an OK message.

19. Check the folder `c:\Windows\assembly` for the presence of a number of new assemblies called `Oracle.DataAccess` or `Policy.2.111.Oracle.DataAccess` (the number may differ).

5.7.3 Impersonation user identity

If you installed the (deprecated) Business Connector, the WebDAV Connector or the Visio Workflow Server, the installer created a Windows impersonation user. This impersonation user is trusted to take on the identify of (that is, to impersonate) an arbitrary Content Manager user. You can use the same impersonation user for all three interfaces.

The impersonation user is the user who uses the **SDL Tridion** Application Pool. This identity defaults to the `NetworkService` account, which resolves to one of the following:

- On French Windows: `AUTORITE NT\SERVICE RÉSEAU`
- On German Windows: `NT AUTORITÄT\NETZWERKDIENST`
- On Italian Windows: `NT AUTHORITY\SERVIZIO DI RETE`
- On Spanish Windows: `NT AUTHORITY\SERVICIO DE RED`
- On all other languages of Windows: `NT AUTHORITY\NETWORK SERVICE`.

5.7.4 Configuring the Content Manager Explorer Web site

Check the redirection key, and execute a Microsoft Knowledge Base article, to configure the Web site.

Steps to execute

1. Navigate to the root directory of your Content Manager Explorer Web site. By default, this location is the subfolder `web\WebUI\WebRoot\` of the Content Manager root location (defaults to `C:\Program Files (x86)\Tridion\`). In this subfolder, find the file `web.config` and open it in a plain-text or XML editor.

2. In `web.config`, do the following:

   a. Ensure the presence of the following key and make sure that it specifies the base hostname and port used for the HTTP protocol:

   ```
   <add key="Tridion.WCF.RedirectTo" value="127.0.0.1:80"/>
   ```

   or

   ```
   <add key="Tridion.WCF.RedirectTo" value="myhost.mydom:80"/>
   ```

   b. Follow the instructions in MSDN Knowledge Base Article 926642: http://support.microsoft.com/kb/926642

5.7.5 Configuring funneling

You may want to funnel different Web sites with different ports and/or host headers to the same Content Manager Explorer location.

Steps to execute

1. Access the machine on which Content Manager is installed and open the `web.config` file of the Content Manager Explorer Web site in a plain-text or XML editor.

   By default, this file is located in the `WebUI\WebRoot` subfolder of the Content Manager root location (defaults to `C:\Program Files (x86)\Tridion\`).

2. In the `appSettings` section of the file, uncomment the following element and set its `value` attribute:

   ```
   <add key="Tridion.WCF.RedirectTo" value="myproxyserver.foo.bar:90"/>
   ```

   where `myproxyserver.foo.bar` is the fully qualified domain name of the alternative site, and `90` is the port number you intend to use.

3. Find the `serviceHostingEnvironment` element and add an attribute called `multipleSiteBindingsEnabled`, setting it to the value `true`.

4. Inside the `serviceHostingEnvironment` element, uncomment the `<baseAddressPrefixFilters>` element and modify it so that it reads:

   ```
   <baseAddressPrefixFilters>
     <add prefix="http://myproxyserver.foo.bar:90"/>
   </baseAddressPrefixFilters>
   ```

   where `myproxyserver.foo.bar` is the fully qualified domain name of an additional site you want to specify, and `90` is the port number you intend to use. You can add as many add elements as you like.

5. Save and close this `web.config` file.

6. Open IIS Manager in one of the following ways :
    - If your Content Manager server runs on a Windows 2008 R2 machine, do the following:
        - From **Administrative Tools**, access your **Server Manager**.
        - In the dialog that opens, in the tree on the left, navigate to the node **Roles** > **Web server (IIS)** > **Internet Information Services (IIS) Manager** . A second tree appears to its right.
    - If your Content Manager server runs on a Windows 2012 machine, do the following:
        - Start**Server Manager**.
        - Select **Tools** and **Internet Information Services (IIS) Manager**.
7. In the IIS tree, navigate to the Content Manager Explorer Web site on the local computer.
8. From the list of options on the far right, click **Bindings**.

    A **Site Bindings** dialogs opens.
9. For each additional staging site you configured in a `siteMapping` element, click **Add** in the **Site Bindings** dialog.

    An **Add Site Binding** dialog opens.
10. Under **Port**, fill in the port number you specified for this additional staging site in the `web.config` file. Following our example in step 2, you would fill in "90" at this point. Alternatively, if you have a host header configured, fill it in under **Host name**.
11. Click **OK** to close the dialog add this identity to the list of identities.
12. When you are finished adding port numbers for additional staging Web sites, click **Close** in the **Site Bindings** dialog to return to IIS Manager.

    You have now configured the additional proxy Web site identities for each of the staging Web sites.
13. Restart IIS.

    The Content Manager Explorer Web site now contains your additional Web site identities.

Next steps

You have now finished configuring additional Web sites.

5.7.6 Configuring search

Configure search if you want to raise the maximum number of words indexed per content item, or if you want to change the ports used for non-secure or SSL connections.

Context

- The maximum number of words indexed per content item defaults to 10,000 words, roughly 14 single-spaced pages of 11-point text.
- The default port used for non-secure connections is 8983 and for SSL 8443 (these may be in use by other Java applications).

Steps to execute

1. If some of your Content Manager items contain more than 10,000 words and you want reindex them using the search engine:

    a. Navigate to the `solr-home\tridion\conf\` subfolder of the Content Manager root location (defaults to `C:\Program Files (x86)\Tridion\`).

    b. Open `solrconfig.xml` in a plain-text or XML editor.

    c. Find the element `<maxFieldLength>` and change its value to the maximum number of words you expect the search engine to have to index.

    d. Save and close `solrconfig.xml`.

    Note: The `<maxFieldLength>` is bound by the available memory, which by default is set to 1 GB. If you specify a very large field length, and try to index a content item with that many words, an out-of-memory exception may occur. You will therefore need to allocate more memory.

2. You can allocate more memory to the search engine as follows:

    a. Navigate to the `bin\` subfolder of the root.

    b. Open `TcmSearchHost.exe.config` in a plain-text or XML editor.

    c. Find the `solrHost` element. This element has an attribute called `jvmArguments` which contains a large number of settings.

    d. Find a setting called `jvmOptions`, which has a value that starts with `-Xmx1024M`.

    e. Change `1024M` (that is, 1024 MB or 1 GB) to a memory value you expect to be sufficient for your `<maxFieldLength>` value.

    f. Save and close `TcmSearchHost.exe.config`.

3. To configure a different port for normal or SSL communication:

    a. Navigate to the `solr-jetty\conf\` directory of the Content Manager root location (defaults to `C:\Program Files (x86)\Tridion\`).

b. Open jetty.xml in an XML or plain-text editor.

c. To configure a different port for a non-SSL connection, find the SystemProperty element with its name attribute set to jetty.port and change its default attribute to a port number of your choice.

d. To configure a different port for an SSL connection, edit the value of the Set element with its name attribute set to confidentialPort.

e. Save and close jetty.xml.

4. If you changed the SSL port:

   a. Open jetty-ssl.xml.

   b. Find the Set element with its name attribute set to the value Port, and change the value of that element to the same value you gave confidentialPort in the jetty.xml file.

   c. Save and close jetty-ssl.xml.

5. Access your Windows **Services**. Find the Windows service called **Tridion Content Manager Search Host** and restart it.

5.7.7 Optimizing the search collection

Regularly optimize your search collection to keep your search collection up-to-date and fast.

Steps to execute

1. To optimize the search collection, access one of the following URLs:
   - for an HTTP connection—http://cmsserver:8983/tridion/update?optimize=true
   - for an HTTPS connection over SSL—https://cmsserver:8443/tridion/update?optimize=true

2. If the HTTPS URL prompts you for a user name and password, enter the name of the MTS User account (which is called MTSUser by default) and the password you configured for that user account.

   By accessing the URL, you perform an optimization of the search collection. If successful, the request yields an HTTP response like the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
  <lst name="responseHeader">
    <int name="status">0</int>
    <int name="QTime">677</int>
  </lst>
</response>
```

   You can use a tool such as curl (available from http://curl.haxx.se) to interact with the URL from the command line, which in turn enables you to schedule this task in Windows.

For more maintenance tasks of the Content Manager search collection, refer to the Solr documentation at the following URL: http://wiki.apache.org/solr/UpdateXmlMessages.

5.7.8 Scheduling cleanup of upload files and preview files

By default, the Content Manager does *not* clean up files that users upload to Content Manager, nor files that Content Manager generates when users preview content. Schedule regular cleanups of the folders that contain those files, which default to `web\WebUI\WebRoot\Upload` and `web\Preview`.

Steps to execute

1. Identify the folder in which Content Manager stores uploaded files.
   - By default, this folder is the `web\WebUI\WebRoot\Upload` subfolder of the Content Manager root location (defaults to `C:\Program Files (x86)\Tridion\`).
   - You may have configured a different location in the `<Tridion.UploadDirectory>` in the file `System.config`, located in the `web\WebUI\WebRoot\Configuration` subdirectory of the Content Manager root location.
2. Create a Windows scheduling task that regularly wipes this folder. Choose a moment of low system activity to perform the wipe, and choose a frequency based on your expectations of how often users will upload files, and how large these files will be.
3. Identify the folder in which the Content Manager stores files generated during previewing.
   - By default, this folder is the `web\Preview` subfolder of the root.
   - You may have configured a different location in the **General settings screen** of the Content Manager configuration environment (MMC Snap-in).
4. Create a Windows scheduling task that regularly wipes this folder. Choose a moment of low system activity to perform the wipe, and choose a frequency based on your expectations of how often users preview content.

5.8 Configuring the WebDAV Connector

This section explains how to configure the server-side part of the WebDAV Connector. Use the WebDAV Connector to allow users to access Content Manager from a client that supports the WebDAV protocol. The basic installation requires you to configure the WebDAV Connector so that you can use Dreamweaver templating.

5.8.1 Enabling WebDAV to handle paths with & sign

To enable WebDAV to handle paths with an & sign in the name, you need to update a registry key.

Steps to execute

1. On the Content Manager server, open the Registry Editor.
2. Navigate to the following registry key:

   ```
   HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\ASP.NET
   ```

3. Add a new key to this key called `VerificationCompatibility` (a `DWORD`) and set it to the value `1`.

   To learn more about this issue, refer to Microsoft Knowledge Base Article 826437 (http://support.microsoft.com/kb/826437).

5.8.2 Configuring WebDAV Connector authentication

To use LDAP or single sign-on (SSO) to authenticate users in Content Manager, you must allow Basic authentication. A Windows WebDAV client refuses to use Basic authentication over a non-secure (non-HTTPS) connection. As a result, to use LDAP or SSO, you must use HTTPS.

Context

For more information about this topic, refer to the following online resources:

- http://technet.microsoft.com/en-us/library/cc787023(v=ws.10).aspx
- http://support.microsoft.com/kb/2123563

5.8.3 Configuring WebDAV Connector for IIS

To use WebDAV Connector for IIS, you must have disabled or not installed Microsoft WebDAV Extension (by default, WebDAV Extension is disabled when you install a new Web site).

5.8.4 Configuring WebDAV Connector for LDAP authentication or single sign-on

No specific steps are needed to make use LDAP or SSO authentication work if you use WebDAV (if both are properly set up, as documented), as long as you ensure that your IIS authentication settings in the `webdav/` folder are the same as for the Web site itself.

5.8.5 Configuring maximum size of WebDAV uploads

The default value for uploading binary files via WebDAV is 4MB (this is an IIS setting). To increase the size limit for uploading binaries via WebDAV you need to adjust the `web.config` file for WebDAV.

Steps to execute

1. Open the WebDAV `web.config` configuration file, located by default in the `webdav` subfolder of the Content Manager root location (defaults to `C:\Program Files (x86)\Tridion\`), in a text editor.

2. Find or create a section called `<httpRuntime>` in the `<system.web>` section, and give it an attribute called `maxRequestLength`, set to the number of bytes you want to set as your maximum. For example, to raise the 4 MB limit to 200 MB, ensure the presence of the following element:

   ```
   <httpRuntime maxRequestLength="209715200"/>
   ```

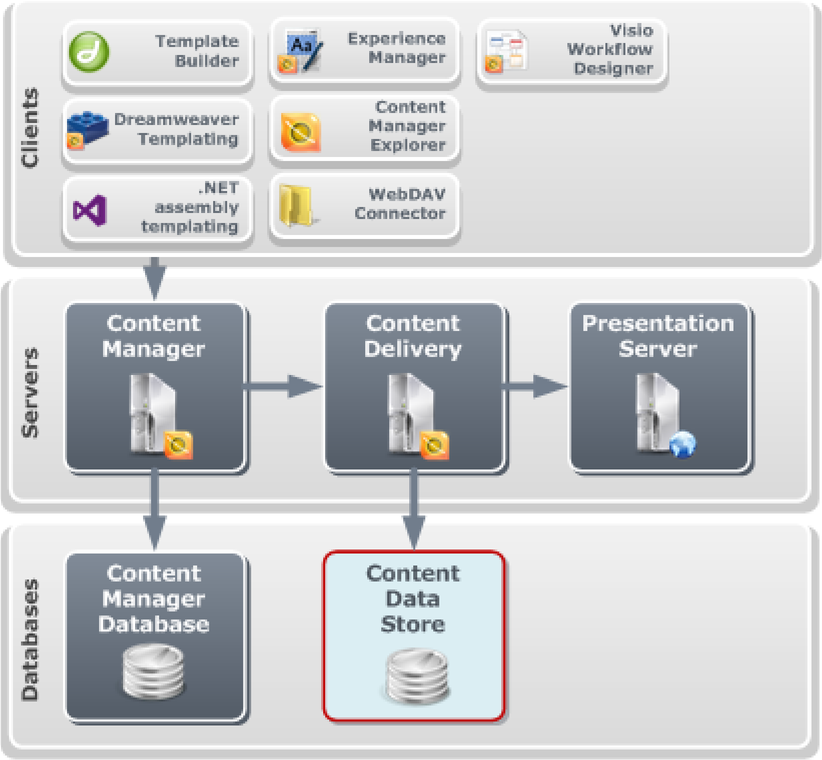
3. In the `<system.webServer>` section, find or add a `<security>` section and inside it, find or add a `<requestFiltering>` section. Again inside this element, insert a `requestLimits` element with a `maxAllowedContentLength` attribute set to your maximum upload size in bytes. For example, the following fragment again sets the maximum to 200 MB:

   ```
   <system.webServer>
     <security>
       <requestFiltering>
         <requestLimits maxAllowedContentLength="209715200"/>
       </requestFiltering>
     </security>
   </system.webServer>
   ```

4. Save and close the `web.config` file.

Chapter 6 Creating the Content Data Store for Content Delivery

This section explains how to install and configure the Content Data Store, the repository containing published content that the Content Delivery server uses.



6.1 Content Delivery storage options

The Content Data Store is a relational database or a local file system (or in a custom storage media).

Content Delivery offers the following storage options for published content:

Filesystem storage

You can store published content on the local file system—the hard disk of the Presentation Server or a network-accessible hard disk. In this case, no additional installation is required.

However, if you use the local file system as your Content Data Store, you restrict the functionality of SDL Tridion as follows:

- You cannot query and filter published pieces of content to dynamically create a Web page. For example, you cannot create a Web page that dynamically retrieves, for example, the last 5 news items created.
- You cannot use Profiling & Personalization. Profiling & Personalization allows you to track which content users see, and to use that information to present specific users with personalized content.
- You cannot use taxonomies to implement intelligent navigation in your Web sites or to retrieve content relations.

Database storage

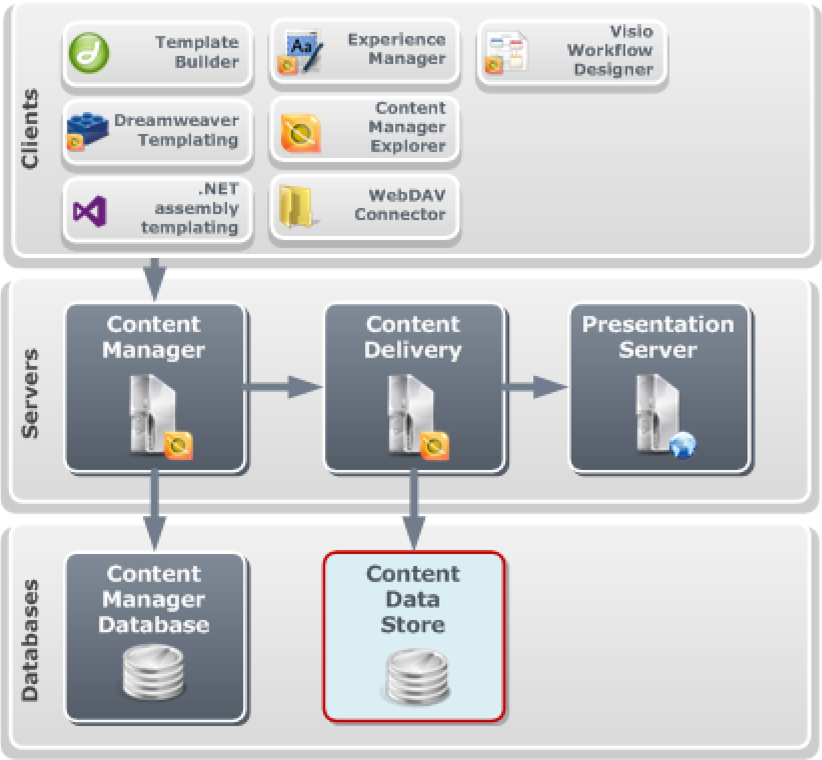
You can store published content in a relational database, in which case you need to create the Content Data Store database on one of the following database servers:

- Microsoft SQL Server
- Oracle
- IBM DB2

Note: You can create the Microsoft SQL Server and Oracle databases on Windows using a PowerShell script (recommended), or manually.

6.2 Content Data Store prerequisites

This topic lists the required and supported software for the Content Data Store where a relational database is used to store published content.



Recommended database servers

SDL recommends one of the following database servers:

- Oracle Server 11.2.0.3 with one of the following patches applied:
  - For Linux/UNIX systems: PSU 11.2.0.3.2
  - For Windows systems: 11.2.0.3 Patch 2 or higher (for Windows)
- Microsoft SQL Server 2012 SP1

Supported Microsoft SQL Server databases

The Content Data Store supports the following Microsoft SQL Server databases:

Supported:

- Microsoft SQL Server 2012 SP1
- Microsoft SQL Server 2008 R2 SP2

Deprecated: None

Supported Oracle databases

The Content Data Store supports the following Oracle databases:

Supported:

- Oracle Database 10*g* Release 2 patch set 10.2.0.5
- Oracle Database 11*g* Release 2 patch set 11.2.0.3 with one of the following patches applied:
  - For Linux/UNIX systems: PSU 11.2.0.3.2
  - For Windows systems: 11.2.0.3 Patch 2 or higher (for Windows)

Deprecated: None

Supported IBM DB2 databases

The Content Data Store supports the following IBM DB2 database:

Deprecated:

DB2 version	Dropped in next version release?
IBM DB2 9.7	Under review

The Content Data Store can reside on any operating system that supports the specific DBMS you choose to use. Refer to your database vendor documentation to learn which operating systems are supported.

You can also store published content on the local file system. However, the following Content Delivery features require a database:

- Dynamically assembling content onto a Web page through querying and filtering.
- Profiling & Personalization which enables you to track visitors and offer them personalized content.
- Intelligent navigation (taxonomies).
- Outbound E-mail (Tracking database) and Audience Manager (Subscription database)—note that IBM DB2 is not supported.
- User-Generated Content (UGC)—note that IBM DB2 is not supported.

Licenses

If you use a relational database as a Content Data Store, your Content Delivery system requires a license. To obtain a license, contact Customer Support.

6.3 Creating the Content Data Store using PowerShell

Create a Content Data Store using PowerShell if you use Windows.

Requirements

To install SDL Tridion databases on a Windows machine requires Windows PowerShell 3.0 and Microsoft .NET Framework 4.0 or higher.

Context

Note: To create an IBM DB2 database, or to create an Oracle database on a non-Windows system, you must run database scripts manually.

Steps to execute

1. Open Windows PowerShell from the Windows Start Menu.
2. In PowerShell, depending on your database navigate to one of the following folders on the installation media:
   - `Database\MSSQL\` or
   - `Database\Oracle\`
3. Type the following command:

   ```
   & '.\Install Content Data Store.ps1'
   ```

4. Follow the instructions in the PowerShell console to install the database.

Result

You have created the Content Data Store.

Next steps

When installing a Server Role, you configure the Content Data Store in the Storage Layer configuration file.

6.4 Creating Content Data Store manually

You can create your Content Data Store database manually by running database scripts provided.

6.4.1 Creating an IBM DB2 Content Data Store database manually

Create your IBM DB2 Content Data Store database manually by running database scripts that create the database and its tables.

Steps to execute

1. Open the DB2 Control Center.
2. In the tree control on the left, right-click **All Databases** and select **Create Database** > **Standard** .

   The **Create Database Wizard** opens. Create a database as you would normally, but take into account the following settings.
3. In the step **1. Name**, do the following:
   - Under **Database name**, choose a name for the database.
   - From the dropdown under **Default bufferpool and table space page size**, select a value of **8K** or higher.
4. In the step **3. Region**, from the dropdown under **Code set**, select **UTF-8**.
5. Proceed with the wizard until you are finished.

   The newly created database appears in the tree as a subitem of **All Databases**.
6. To create the tables for the Content Data Store database in DB2, select the newly created database node in the tree.

   A number of options appears at bottom right.
7. From these options, select **Query**.

   A new **Command Editor** tab opens in the Control Center, with **Target** set to your new database.
8. At the bottom of your screen, enter an exclamation mark ('!') next to **Statement termination character**.
9. From the toolbar, click **Open**, browse to the following folder on the SDL Tridion installation media: `Database\Scripts\db2\create\`
10. Open and run each of the following scripts in the order listed:

    a. `CD_BROKER_TABLES.db2`

    b. `CD_BROKER_CONSTRAINTS.db2`

    c. `CD_BROKER_INDEXES.db2`

    d. `CD_WAI_TABLES.db2`

    e. `CD_WAI_CONSTRAINTS.db2`

f. `CD_WAI_INDEXES.db2`

11. Navigate to the folder `Database\Scripts\db2\shared\70\` and in that location, run the script `GENERAL_SET_VERSION.db2`.

Result

You have created the Content Data Store.

Next steps

When installing a Server Role, you configure the Content Data Store in the Storage Layer configuration file.

6.4.2 Creating a Microsoft SQL Server Content Data Store database manually

Create your SQL Server Content Data Store database manually by running the database scripts provided.

Steps to execute

1. Close all connections to your Microsoft SQL Server database server.
2. Access your SDL Tridion installation media.
3. Navigate to the folder `Database\Scripts\MSSQL\Create\` and run the following scripts:
   - `CD_BROKER_CREATE.sql`
   - `CD_BROKER_TABLES.sql`
   - `CD_BROKER_CONSTRAINTS.sql`
   - `CD_BROKER_INDEXES.sql`
   - `CD_BROKER_DATA.sql`
   - `CD_WAI_TABLES.sql`
   - `CD_WAI_CONSTRAINTS.sql`
   - `CD_WAI_INDEXES.sql`
4. Navigate to the folder `Database\Scripts\MSSQL\Shared\70\` and run the following script\:
   - `GENERAL_SET_VERSION.sql`

Result

You have created the Content Data Store.

Next steps

When installing a Server Role, you configure the Content Data Store in the Storage Layer configuration file.

6.4.3 Creating an Oracle Content Data Store database manually

Create your Oracle Content Data Store database manually by running the database scripts provided.

Steps to execute

1. Access your SDL Tridion installation media.
2. In the `Database\Scripts\Oracle\Create\` folder, run the following script as the SQL SDL Tridion administrator (defaults to `TRIDION_SYS`):
   - `CD_BROKER_CREATE_USER.sql`

   The script creates a new user (schema owner).
3. In the `Database\Scripts\Oracle\Shared\70\` folder, run the following script as the newly created user (schema owner):
   - `TCM_SYSTEM.sql`

   The script creates a new user (schema owner).
4. In the `Database\Scripts\Oracle\Create\` folder, run the following scripts as the newly created user (schema owner):
   - `CD_BROKER_TABLES.sql`
   - `CD_BROKER_CONSTRAINTS.sql`
   - `CD_BROKER_INDEXES.sql`
   - `CD_WAI_TABLES.sql`
   - `CD_WAI_SEQUENCES.sql`
   - `CD_WAI_CONSTRAINTS.sql`
   - `CD_WAI_INDEXES.sql`
5. In the folder `Database\Scripts\Oracle\Shared\70\`, run the following script as the newly created user (schema owner):
   - `GENERAL_SET_VERSION.sql`
6. In the `Database\Scripts\Oracle\Create\` folder, run the following script as the SQL SDL Tridion administrator (defaults to `TRIDION_SYS`):
   - `CD_FINALIZE_RIGHTS.sql`

   The script removes rights that the new user no longer needs.
7. In the folder `Database\Scripts\Oracle\Shared\70\`, run the following script as the SQL SDL Tridion administrator (defaults to `TRIDION_SYS`):
   - `TCM_FINALIZE.sql`

Result

You have manually created the Content Data Store as an Oracle database.

Next steps

When installing a Server Role, you configure the Content Data Store in the Storage Layer configuration file.

Chapter 7 Creating the Experience Manager database

This section explains how to install and configure the Experience Manager database, the repository containing published and edited content that Experience Manager uses.

7.1 Experience Manager database prerequisites

This topic lists the required and supported software for the Experience Manager database.

Recommended database servers

SDL recommends one of the following database servers:

- Oracle Server 11.2.0.3 with one of the following patches applied:
  - For Linux/UNIX systems: PSU 11.2.0.3.2
  - For Windows systems: 11.2.0.3 Patch 2 or higher (for Windows)
- Microsoft SQL Server 2012 SP1

Supported Microsoft SQL Server databases

The Experience Manager database supports the following Microsoft SQL Server databases:

Supported:

- Microsoft SQL Server 2012 SP1
- Microsoft SQL Server 2008 R2 SP2

Deprecated: None

Supported Oracle databases

The Experience Manager database supports the following Oracle databases:

Supported:

- Oracle Database 10*g* Release 2 patch set 10.2.0.5
- Oracle Database 11*g* Release 2 patch set 11.2.0.3 with one of the following patches applied:
  - For Linux/UNIX systems: PSU 11.2.0.3.2
  - For Windows systems: 11.2.0.3 Patch 2 or higher (for Windows)

Deprecated: None

Supported IBM DB2 databases

The Experience Manager database supports the following IBM DB2 database:

Deprecated:

DB2 version	Dropped in next version release?
IBM DB2 9.7	Under review

7.2 Creating the Experience Manager database using PowerShell

You can use a PowerShell script to create a Microsoft SQL Server or Oracle database from a Windows system.

Requirements

To install SDL Tridion databases on a Windows machine requires Windows PowerShell 3.0 and Microsoft .NET Framework 4.0 or higher.

Context

Note: To create an IBM DB2 database, or to create an Oracle database on a non-Windows system, you must run database scripts manually.

Steps to execute

1. Open Windows PowerShell from the Windows Start Menu.
2. In PowerShell, depending on your database navigate to one of the following folders on the installation media:
   - `Database\MSSQL\` or
   - `Database\Oracle\`
3. Type the following command:

   ```
   & '.\Install Content Data Store.ps1'
   ```

4. Follow the instructions in the PowerShell console to install the database.

7.3 Creating the Experience Manager database manually

You can create your Experience Manager database manually by running database scripts provided.

7.3.1 Creating an IBM DB2 Experience Manager database manually

Create your IBM DB2 Experience Manager database manually by running database scripts that create the database and its tables.

Steps to execute

1. Open the DB2 Control Center.
2. In the tree control on the left, right-click **All Databases** and select **Create Database** > **Standard** .

   The **Create Database Wizard** opens. Create a database as you would normally, but take into account the following settings.
3. In the step **1. Name**, do the following:
   - Under **Database name**, choose a name for the database.
   - From the dropdown under **Default bufferpool and table space page size**, select a value of **8K** or higher.
4. In the step **3. Region**, from the dropdown under **Code set**, select **UTF-8**.
5. Proceed with the wizard until you are finished.

   The newly created database appears in the tree as a subitem of **All Databases**.
6. To create the tables for the Content Data Store database in DB2, select the newly created database node in the tree.

   A number of options appears at bottom right.
7. From these options, select **Query**.

   A new **Command Editor** tab opens in the Control Center, with **Target** set to your new database.
8. At the bottom of your screen, enter an exclamation mark ('!') next to **Statement termination character**.
9. From the toolbar, click **Open**, browse to the following folder on the SDL Tridion installation media: `Database\Scripts\db2\create\`
10. Open and run each of the following scripts in the order listed:

    a. `CD_BROKER_TABLES.db2`

    b. `CD_BROKER_CONSTRAINTS.db2`

    c. `CD_BROKER_INDEXES.db2`

    d. `CD_WAI_TABLES.db2`

    e. `CD_WAI_CONSTRAINTS.db2`

f. `CD_WAI_INDEXES.db2`

11. Navigate to the folder `Database\Scripts\db2\shared\70\` and in that location, run the script `GENERAL_SET_VERSION.db2`.

Result

You have created the Experience Manager database.

Next steps

When installing the Experience Manager Server Role, you configure the Experience Manager database in the Storage Layer configuration file.

7.3.2 Creating a Microsoft SQL Server Experience Manager database manually

Create your Microsoft SQL Server Experience Manager database manually by running database scripts that create the database and its tables.

Steps to execute

1. Close all connections to your Microsoft SQL Server database server.

2. Access your SDL Tridion installation media.

3. Navigate to the folder `Database\Scripts\MSSQL\Create\` and run the following scripts:

   - `CD_BROKER_CREATE.sql`
   - `CD_BROKER_TABLES.sql`
   - `CD_BROKER_CONSTRAINTS.sql`
   - `CD_BROKER_INDEXES.sql`
   - `CD_BROKER_DATA.sql`
   - `CD_WAI_TABLES.sql`
   - `CD_WAI_CONSTRAINTS.sql`
   - `CD_WAI_INDEXES.sql`

4. Navigate to the folder `Database\Scripts\MSSQL\Shared\70\` and run the following script\:

   - `GENERAL_SET_VERSION.sql`

Result

You have created the Experience Manager database.

Next steps

When installing the Experience Manager Server Role, you configure the Experience Manager database in the Storage Layer configuration file.

7.3.3 Creating an Oracle Experience Manager database manually

Create your Oracle Experience Manager database manually by running database scripts that create the database and its tables.

Steps to execute

1. Access your SDL Tridion installation media.
2. In the `Database\Scripts\Oracle\Create\` folder, run the following script as the SQL SDL Tridion administrator (defaults to `TRIDION_SYS`):

   - `CD_BROKER_CREATE_USER.sql`

   The script creates a new user (schema owner).
3. In the `Database\Scripts\Oracle\Shared\70\` folder, run the following script as the newly created user (schema owner):

   - `TCM_SYSTEM.sql`

   The script creates a new user (schema owner).
4. In the `Database\Scripts\Oracle\Create\` folder, run the following scripts as the newly created user (schema owner):

   - `CD_BROKER_TABLES.sql`
   - `CD_BROKER_CONSTRAINTS.sql`
   - `CD_BROKER_INDEXES.sql`
   - `CD_WAI_TABLES.sql`
   - `CD_WAI_SEQUENCES.sql`
   - `CD_WAI_CONSTRAINTS.sql`
   - `CD_WAI_INDEXES.sql`
5. In the folder `Database\Scripts\Oracle\Shared\70\`, run the following script as the newly created user (schema owner):

   - `GENERAL_SET_VERSION.sql`
6. In the `Database\Scripts\Oracle\Create\` folder, run the following script as the SQL SDL Tridion administrator (defaults to `TRIDION_SYS`):

   - `CD_FINALIZE_RIGHTS.sql`

   The script removes rights that the new user no longer needs.
7. In the folder `Database\Scripts\Oracle\Shared\70\`, run the following script as the SQL SDL Tridion administrator (defaults to `TRIDION_SYS`):

   - `TCM_FINALIZE.sql`

Result

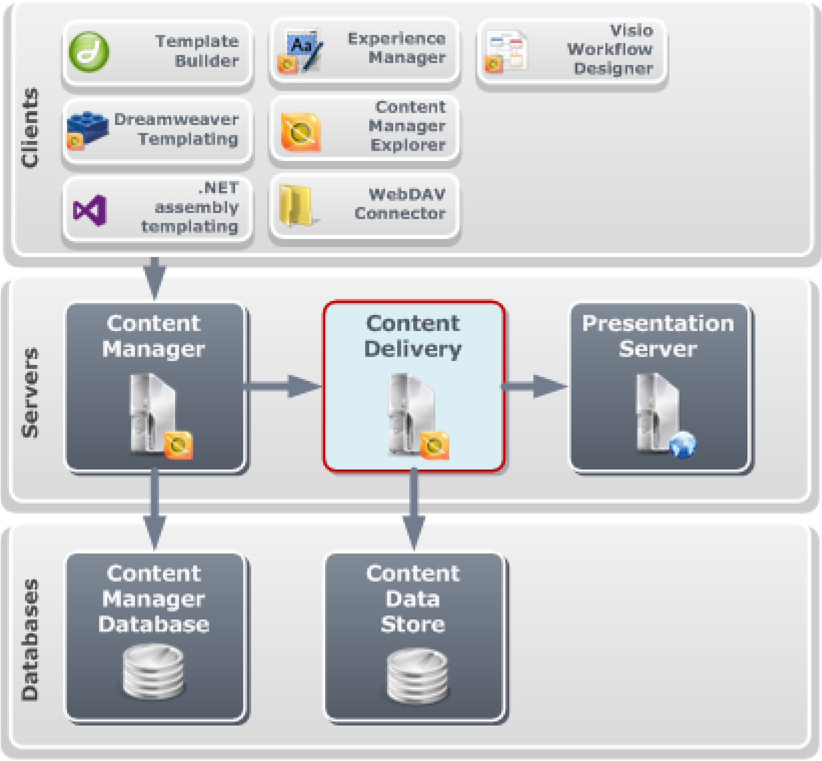
You have created the Experience Manager database.

Next steps

When installing the Experience Manager Server Role, you configure the Experience Manager database in the Storage Layer configuration file.

Chapter 8 Installing core Content Delivery Server Roles

Content Delivery consists of Server Roles: features that work with content published from Content Manager. This section explains how to install each Server Role separately, but you can also combine Server Roles with each other, or with existing Web applications. A basic installation only requires some of these Server Roles to be installed.



8.1 Content Delivery Server Roles

Content Delivery Server Roles correspond to a specific feature or piece of functionality that deals with content published from Content Manager. The basic installation of the core software modules requires the installation of some of these Server Roles.

The following Content Delivery Server Roles are needed for a basic installation:

Content Deployer (HTTP or HTTPS) or Content Deployer (other protocols)

Each of these Server Roles receives and processes incoming content published from the Content Manager, and passes it on to the Storage Layer for storage in the Content Data Store. Install Content Deployer (HTTP or HTTPS) if you intend to publish content over an HTTP or HTTPS transport protocol, or Content Deployer (other protocols) otherwise.

Experience Manager

Install this Server Role to enable users to work with content directly on the staging Web site. This Server Role consists of multiple software components:

- Experience Manager Web service - a Web service used for enabling in-context editing
- Experience Manager Web site extension - software to add to the staging Web site

Note that this product feature also involves installation of software on the Content Manager side. Note that when you set up this product feature in combination with a classic ASP Web site, the following features will not work:

- Session Preview
- Footprint Sets

API

Install this Server Role (typically on an SDL Tridion-enabled Web site) if you want to code against the Content Delivery APIs, for example to dynamically query the Content Data Store or to use Dynamic Linking, Profiling & Personalization or the Ambient Data Framework.

The following Content Delivery Server Roles can be installed separately at a later time:

Cache Channel Service

Install this Server Role if you have a scaled-out scenario and you need to synchronize caches on several machines.

Monitoring

Install and configure this Server Role to monitor whether Server Roles are up and running.

User Generated Content

Install this Server Role to allow visitors to your Web site to rate your content and leave comments. This Server Role consists of multiple software components:

- User Generated Content database - a SQL Server or Oracle database for storing ratings and comments
- Community Web service - a Web service used for submitting ratings and comments
- Moderation Web service - a Web service used for moderating ratings and comments
- User Generated Content Web site extension - software to add to the live Web site
- User Generated Content Content Deployer extensions - software to add to the Content Deployer Server Role

Note that this product feature also involves installation of software on the Content Manager side. Note also that this product feature does not work in combination with a classic ASP Web site.

Content Delivery Web service

Install this Server Role to enable external parties to interact with your Content Data Store from other Web sites.

8.2 Core Server Role installation options

Depending on the Server Role, you can install Content Delivery server roles as Windows services, standalone Java processes, or in a JSP or .NET Web application.

The following core Server Roles are installable as a 64-bit Windows service or as a Java process:

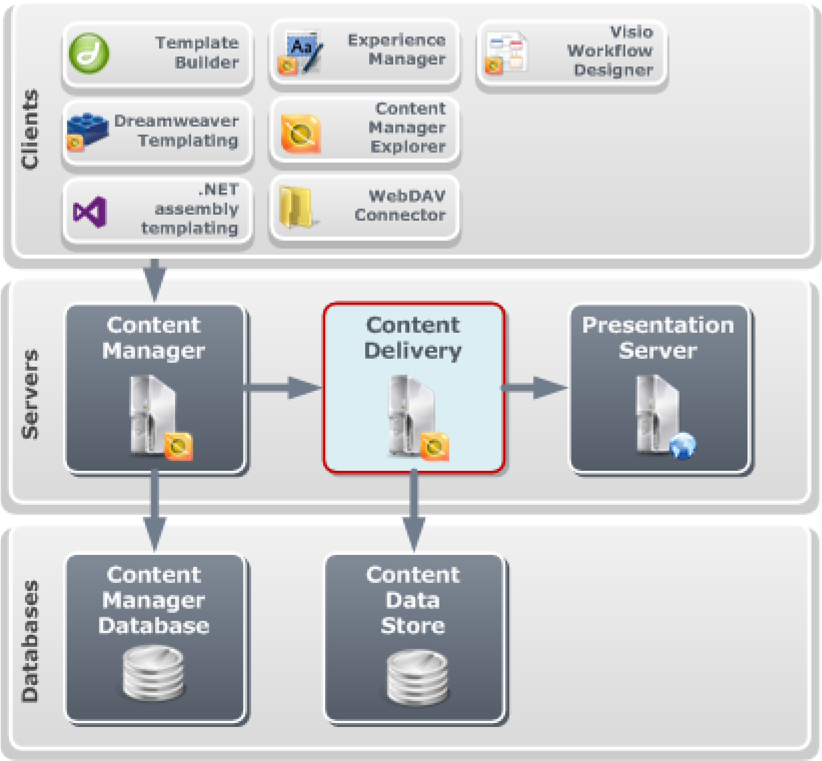
- Content Deployer (other protocols)

The following (parts of) core Server Roles are installable as (part of) a Web application (.NET or JSP):

- Content Deployer (HTTP or HTTPS)
- Content Deployer (other protocols): only install this Server Role as a Web application if you have a pressing reason to do so.
- Experience Manager - Web service
- Experience Manager - Web site extension
- API

8.3 Content Delivery prerequisites

This topic lists the required and supported software for installing Content Delivery Server Roles. The prerequisites are the same for all Server Roles.



Recommended prerequisites

SDL recommends the following set of Content Delivery prerequisites:

Java/JSP implementation

If you publish to Java/JSP, SDL recommends the following setup:

Prerequisite	Recommended product release
Operating system	The latest version of any of the supported operating systems
Web and Application server	The latest version of any of the supported Web and Application servers
Java	Java SE 7.0

ASP.NET implementation

If you publish to ASP.NET, SDL recommends the following setup:

Prerequisite	Recommended product release
Operating system	Microsoft Windows Server 2012 (x64)
Web and Application server	Microsoft IIS 8
Java	Java SE 7.0
.NET Framework	Microsoft .NET Framework 4.5

Microsoft Windows operating systems

Content Delivery Server Roles run on the following Windows operating systems:

Supported:

- Microsoft Windows Server 2012 (x64)
- Microsoft Windows 2008 R2 SP1 (x64)

Deprecated: None

Note: The SDL Tridion installation media contains a Web page, `\Redistributables\Readme.htm`, with links to the Microsoft download pages for the Service Packs listed.

Your Windows release must be in one of the following languages:

- US English
- French
- German
- Spanish
- Dutch
- Japanese

The language of your Windows version must correspond to the language of related software. For example, using a French version of Windows requires a French database.

Non-Windows operating systems

Content Delivery Server Roles run on the following non-Windows operating systems:

Supported:

Vendor	Operating system version
Sun	Solaris 9
Sun	Solaris 10
IBM	IBM AIX 6.1
IBM	IBM AIX 7.1
Redhat	Redhat Enterprise Linux 5.9
Redhat	Redhat Enterprise Linux 6.4
HP	HP-UX 11i V2
HP	HP-UX 11i V3

Deprecated: None

Database client

If you intend to store published content in a database, your Content Delivery Server Role requires a database client (JDBC driver) to connect to the Content Data Store. The database client software you need depends on your database vendor:

Vendor	JDBC Driver	Required JAR
Microsoft SQL Server	Microsoft SQL Server JDBC Driver 4.0	`sqljdbc4.jar`
Oracle	Oracle JDBC driver 11.2.0.3	`ojdbc6.jar`
IBM DB2	IBM Data Server Driver for JDBC and SQLJ (9.7 GA) *(deprecated)*	`db2jcc.jar`

To download these files, access the SDL Tridion installation media and in the `Redistributables\` folder, open the Web page `Readme.htm` to see a list of links to the JDBC drivers.

Java

Content Delivery Server Roles, including those running on a .NET Web site, require a Java Runtime Environment (JRE). The JRE can be any Java SE 7.0 or J2SE 6.0 compliant runtime environment that ships with your Web Application Server. If your Web Application Server does not ship with a JRE, install one of the Java software products listed below:

Supported:

- Java 7.0, bitsize 32 or 64
- Java 6.0, bitsize 32 or 64 (deprecated)

To download one of these Java releases, access the SDL Tridion installation media and in the `Redistributables\` folder, open the Web page `Readme.htm` to see a list of links to the Java releases.

To run Java on Windows, you also need the Microsoft Visual C++ 2010 SP1 Redistributable (with the same bitsize as the Java instance you intend to use), which is also accessible from that Web page.

To install a Server Role as a Windows service, you require a 64-bit JVM.

.NET requirement

If you intend to publish ASP.NET pages on your Web site or to install any Server Role as a .NET Web application, you require one of the following .NET environments:

Supported:

- Microsoft .NET Framework version 4.5, CLR 4.5
- Microsoft .NET Framework version 4.0, CLR 4.0

Deprecated:

.NET Framework version	Dropped in next version release?
Microsoft .NET Framework version 3.5 SP1, CLR 2.0	Probably

To download one of these .NET Frameworks, access the SDL Tridion installation media and in the `Redistributables\` folder, open the Web page `Readme.htm` to see a list of links to the .NET Frameworks.

Web Servers and Application Servers

Content Delivery Server Role that you install as a Web application require a Web server or a Web Application Server. The following table lists the Web Servers and Web Application Servers that Content Delivery supports (your choice of Web and Application server restricts the operating systems you can use for your Content Delivery Server Roles):

Supported:

Vendor	Name	Version
Apache Software Foundation	Tomcat	6.0
Apache Software Foundation	Tomcat	7.0
IBM	WebSphere	7.0
IBM	WebSphere	8.5
Microsoft	IIS (Internet Information Services)	7.5
Microsoft	IIS (Internet Information Services)	8.0
Oracle	WebLogic Server	11*g* R1
Oracle	WebLogic Server	12c
Oracle	GlassFish Server	v3
RedHat	JBoss Enterprise Application Platform	5.1
RedHat	JBoss Enterprise Application Platform	6.0

If you use IBM WebSphere 7.0, you must also have the Feature Pack for OSGi Applications and Java Persistence API (JPA) 2.0 installed.

Deprecated: None

8.4 Installing Server Roles as Windows services

To install a Server Role as a Windows service, choose a Server Role to install and follow the instructions. All Windows services are 64-bit processes and require a 64-bit JVM.

8.4.1 Installing the Content Deployer (other protocols) as a Windows service

To install the Content Deployer as a 64-bit Windows service, use the Content Delivery installer. You can only do this if your transport protocol is not HTTP or HTTPS. You can also install the Content Deployer as a .NET Web application.

Steps to execute

1. Open a command prompt.

2. Access the SDL Tridion installation media, navigate to the `Content Delivery\installer\` folder and run the file `startCDinstaller.bat`.

3. Follow the instructions on the screen and select to install Content Deployer in a location you specify.

4. After the installer has finished, add the license file you received from Customer Support to the location you selected, giving it the name `cd_licenses.xml`. Your license must enable Content Deployer as a feature.

5. Access the `config\` subfolder of the location you specified.

6. Open the file `cd_deployer_conf.xml` for editing.

7. In the `<Queue>` section, add a `Location` element for each protocol over which you publish content to this Content Deployer, for example:

```
<Location Path="MYPATH" WindowSize="20" Workers="10" Cleanup="true" Interval="2s" />
```

For each `Location` element, specify the `Path` and optionally other attributes to optimize the publishing process:

Path

The absolute path to a location specified in a Publication Target as a relative path:

Protocol	Publication Target Property (relative path)
FTP	**Location**
SFTP (FTP over SSH)	**Path on the FTP server**
SSHFTP (SSH2 using an FTP subsystem)	**Location**
Local file copy	**Location**

For example, if the **Location** property specifies `/dest/` for an FTP publishing destination, set `Path` to the absolute path to this location: `c:\inetpub\ftproot\dest`.

WindowSize

Sets the maximum number of Transport Package that can be "in flight" (publishing or deploying) at any one time. It represents the number of publish actions the Content Deployer should be able to handle. The default value for this attribute is 20.

Workers

Sets the maximum number of deployment threads that can be running at any time. If `WindowSize` is bigger than `Workers`, items to publish may enter a deployment queue, which in turn may cause a publish action to time out and fail. The default value for this attribute is `10`.

Note: If you store metadata on the file system rather than in a database, you cannot configure multiple workers.

Interval

The amount of time, in milliseconds (`ms`), seconds (`s`), hours (`h`) and so on, that the Content Deployer waits before checking for new content in this location. The default value for this attribute is `2s`; that is, two seconds.

8. If you intend to use SDL Tridion .NET Server Controls, ensure that your `<TCDLEngine>` contains a `<Properties>` section containing the following property:

   ```
   <Property Name="tcdl.dotnet.style" Value="controls" />
   ```

9. Save and close `cd_deployer_conf.xml`.

10. Access the `bin\` subdirectory of the installation location and enter the following command:

    ```
    cd_deployer.exe -install
    ```

Result

Content Deployer (other protocols) is now installed as Windows service. Note that this Windows service requires a 64-bit JVM.

8.5 Installing Server Roles as .NET Web applications

To install a Server Role as a .NET Web application, choose a Server Role to install and follow the instructions.

8.5.1 Installing the Content Deployer (HTTP or HTTPS) as a .NET Web application

Install the Content Deployer (HTTP or HTTPS) Server Role, to enable transport over the HTTP(s) protocol.

Requirements

The license file you use when installing this Server Role must include a Content Deployer license.

Steps to execute

1. Ensure that all the prerequisites for a Content Delivery Server Role are met on your target system.

2. In IIS Manager, create or select a Web site that will contain your Content Deployer (HTTP or HTTPS) .NET Web application.

3. Select the Web site and in the **IIS** area on the right, select **Authentication**. In the list of properties that appears, set **Anonymous Authentication** to **Disabled** and **Basic Authentication** to **Enabled**.

4. If this is a 32-bit .NET Web application running in combination with Java 6, then in the advanced system properties of your machine, select **Environment Variables** to see and edit your environment variables, and ensure that your `PATH` variable contains the `bin\` subfolder of the Java instance you are using. This location contains the file `MSVCR71.DLL`.

5. Access the SDL Tridion installation media and navigate to the `Content Delivery\roles\` folder.

6. Access the `upload\dotNET` subfolder.

7. From this location, copy the file `HTTPUpload.aspx` to the root of the Web site.

8. Configure the IIS user who will be executing the `HTTPUpload.aspx` Web page to have modify access to the Content Deployer location for incoming content. By default, the IIS user is the identity running the SDL Tridion Application Pool (which typically is the Network Service user).

9. Access the `webapp\` subfolder.

10. Do one of the following:
    - To install a 32-bit .NET Web application, access the `x86\` subfolder.
    - To install a 64-bit .NET Web application, access the `x86_64\` subfolder.

11. Unzip the Zip file in this location to the location on your target system where you want to place the .NET Web application.

12. If you store published content in a database, download the JDBC driver (JAR file) required for your specific database vendor and copy it into the `bin\lib` folder of your Web application:

Database vendor	JDBC Driver	File to download
Microsoft SQL Server	Microsoft JDBC Driver 4.0 for SQL Server	`sqljdbc4.jar`
Oracle	Oracle JDBC driver 11.2.0.3	`ojdbc6.jar`

13. Navigate to `\Content Delivery\resources\configurations\`.

14. Copy `logback.xml` from that folder to the `bin\config` folder of your Web application.

15. Copy and rename the following configuration files to the `bin\config` folder of your Web application:

Corresponding configuration file	Renamed configuration file
`cd_deployer_conf_sample.xml`	`cd_deployer_conf.xml`
`cd_storage_conf_sample.xml`	`cd_storage_conf.xml`
`cd_link_conf_sample.xml`	`cd_link_conf.xml`

16. Open the file `cd_deployer_conf.xml` for editing.

17. In the `<Queue>` section, add a `Location` element, for example:

    ```
    <Location Path="MYPATH" WindowSize="20" Workers="10" Cleanup="true" Interval="2s" />
    ```

    Specify the `Path` and optionally other attributes to optimize the publishing process:

    `Path`

    The absolute path to the pickup location.

    `WindowSize`

    The maximum number of Transport Package that can be "in flight" (publishing or deploying) at any one time. It represents the number of publish actions the Content Deployer should be able to handle. The default value for this attribute is 20.

    `Workers`

    The maximum number of deployment threads that can be running at any time. If `WindowSize` is bigger than `Workers`, items to publish may enter a deployment queue, which in turn may cause a publish action to time out and fail. The default value for this attribute is `10`.

    Note: If you store metadata on the file system rather than in a database, you cannot configure multiple workers.

    `Interval`

    The amount of time, in milliseconds (`ms`), seconds (`s`), hours (`h`) and so on, that the Content Deployer waits before checking for new content in this location. The default value for this attribute is `2s`; that is, two seconds.

18. Directly below the `<Queue>` section, add the following element:

    ```
    <HTTPSReceiver MaxSize="10000000" Location="MYPATH" InProcessDeploy="true" />
    ```

    Configure the attributes as follows:

    `MaxSize`

    The maximum size (in bytes) of a Transport Package sent to the location.

    `Location`

    The same value as the *MYPATH* you specified in the `Location` element earlier.

    `InProcessDeploy`

    If you set this attribute to `true` (this is the case by default), publishing and deployment is contained within the Web application. If you set it to `false`, you must have your Content Manager publish content over HTTP or HTTPS to a location outside the Web application, and run the Content Deployer as a standalone Windows service or Java process. In this setup, the entire publishing transaction bypasses your Web application, which can improve your publishing performance. Refer to for a more detailed explanation of this implementation plan.

19. Save and close `cd_deployer_conf.xml`.

20. Open `cd_ambient_conf.xml` for editing (if using).

21. Find the `<Security>` section and enclose it in comments. This element is only needed if you intend to set up OAuth authentication.

22. Save and close `cd_ambient_conf.xml`.

23. Copy `cd_licenses.xml`, your Content Delivery license file available from SDL Tridion Customer Support, to the `bin\config` directory of your Web application.

    If you want to name the file differently or put it in a different location, you need to update the `License` element in your configuration files.

24. Configure logging.

25. Configure the Storage Layer and optionally encrypt sensitive strings.

26. Disable recycling of the Application Pool of this Web application by doing the following:

    a. Open IIS Manager.

    b. From the tree on the left, select the current machine and navigate to the Web application you just created.

    c. Click **Advanced settings** to open a dialog with the application's properties, and note down the value of **Application Pool**.

    d. Close the dialog.

    e. From the tree on the left, under the current machine, now select the **Application Pools** node.

    f. From the list on the right, select the Application Pool you just noted down.

    g. Select **Recycling** in the **Edit Application Pool** area on the right.

    h. In the dialog that opes, deselect **Regular time intervals**, then click **Next** and **Finish** to commit your change.

    i. Restart IIS.

27. Restart the Web application.

28. Test the installation by accessing the ASPX page `HTTPUpload.aspx` from a Web browser. The browser should display the following:

    ```
    SDL Tridion ASPX Upload Page
    ```

8.5.2 Installing the Content Deployer (other protocols) as a .NET Web application

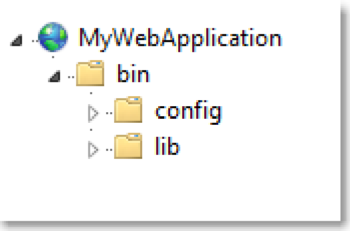
Typically, if your protocol is not HTTP or HTTPS and your operating system is Windows, you would install the Content Deployer (other protocols) as a Windows service. Only if you have a pressing reason to do so, install Content Deployer as a .NET Web application.

Requirements

The license file you use when installing this Server Role must include a Content Deployer license.

Steps to execute

1. Ensure that all the prerequisites for a Content Delivery Server Role are met on your target system.
2. If this is a 32-bit .NET Web application running in combination with Java 6, then in the advanced system properties of your machine, select **Environment Variables** to see and edit your environment variables, and ensure that your `PATH` variable contains the `bin\` subfolder of the Java instance you are using. This location contains the file `MSVCR71.DLL`.
3. Create the following Web application directory structure:
   a. In the root folder, create a `bin\` subfolder.
   b. In the `bin\` folder, create subfolders `config\` and `lib\`.



4. Access the SDL Tridion installation media and navigate to the `Content Delivery\roles\` folder.
5. Access the `deployer\DotNET` subfolder.
6. Copy all DLLs in this folder to the `bin\` folder of your Web application.
7. Do one of the following:
   - To install a 32-bit .NET Web application, access the `x86\` subfolder.
   - To install a 64-bit .NET Web application, access the `x86_64\` subfolder.
8. Copy all DLLs in this folder to the `bin\` folder of your Web application.
9. On the installation media, navigate to the `Content Delivery\roles\deployer\java\` folder.

10. Copy all the JAR files from the `lib\` subfolder, and all the JAR files from the `third-party-lib\` subfolder, to the `bin\lib\` folder of your Web application.

11. If you store published content in a database, download the JDBC driver (JAR file) required for your specific database vendor and copy it into the `bin\lib` folder of your Web application:

Database vendor	JDBC Driver	File to download
Microsoft SQL Server	Microsoft JDBC Driver 4.0 for SQL Server	`sqljdbc4.jar`
Oracle	Oracle JDBC driver 11.2.0.3	`ojdbc6.jar`

12. Navigate to `\Content Delivery\resources\configurations\`.

13. Copy `logback.xml` from that folder to the `bin\config` folder of your Web application.

14. Copy and rename the following configuration files from that folder to the `bin\config` folder of your Web application:

Original configuration file	Renamed configuration file
`cd_deployer_conf_sample.xml`	`cd_deployer_conf.xml`
`cd_storage_conf_sample.xml`	`cd_storage_conf.xml`
`cd_link_conf_sample.xml`	`cd_link_conf.xml`

15. Open the file `cd_deployer_conf.xml` for editing.

16. In the `<Queue>` section, add a `Location` element for each protocol over which you publish content to this Content Deployer, for example:

```
<Location Path="MYPATH" WindowSize="20" Workers="10" Cleanup="true" Interval="2s" />
```

For each `Location` element, specify the `Path` and optionally other attributes to optimize the publishing process:

Path

The absolute path to a location specified in a Publication Target as a relative path:

Protocol	Publication Target Property (relative path)
FTP	**Location**
SFTP (FTP over SSH)	**Path on the FTP server**
SSHFTP (SSH2 using an FTP subsystem)	**Location**
Local file copy	**Location**

For example, if the **Location** property specifies `/dest/` for an FTP publishing destination, set `Path` to the absolute path to this location: `c:\inetpub\ftproot\dest`.

`WindowSize`

Sets the maximum number of Transport Package that can be "in flight" (publishing or deploying) at any one time. It represents the number of publish actions the Content Deployer should be able to handle. The default value for this attribute is 20.

`Workers`

Sets the maximum number of deployment threads that can be running at any time. If `WindowSize` is bigger than `Workers`, items to publish may enter a deployment queue, which in turn may cause a publish action to time out and fail. The default value for this attribute is `10`.

Note: If you store metadata on the file system rather than in a database, you cannot configure multiple workers.

`Interval`

The amount of time, in milliseconds (`ms`), seconds (`s`), hours (`h`) and so on, that the Content Deployer waits before checking for new content in this location. The default value for this attribute is `2s`; that is, two seconds.

17. If you intend to use SDL Tridion .NET Server Controls, ensure that your `<TCDLEngine>` contains a `<Properties>` section containing the following property:

    ```
    <Property Name="tcdl.dotnet.style" Value="controls" />
    ```

18. Save and close `cd_deployer_conf.xml`.

19. Copy `cd_licenses.xml`, your Content Delivery license file available from SDL Tridion Customer Support, to the `bin\config` directory of your Web application.

    If you want to name the file differently or put it in a different location, you need to update the `License` element in your configuration files.

20. Configure logging.

21. Configure the Storage Layer and optionally encrypt sensitive strings.

22. Disable recycling of the Application Pool of this Web application by doing the following:

    a. Open IIS Manager.

    b. From the tree on the left, select the current machine and navigate to the Web application you just created.

    c. Click **Advanced settings** to open a dialog with the application's properties, and note down the value of **Application Pool**.

    d. Close the dialog.

e. From the tree on the left, under the current machine, now select the **Application Pools** node.

f. From the list on the right, select the Application Pool you just noted down.

g. Select **Recycling** in the **Edit Application Pool** area on the right.

h. In the dialog that opes, deselect **Regular time intervals**, then click **Next** and **Finish** to commit your change.

i. Restart IIS.

23. Restart the Web application.

8.5.3 Installing the Experience Manager Web service as a .NET Web application

As part of installing Experience Manager , install the Experience Manager Web service Server Role. Note that this Web application also contains the Content Delivery Web service Server Role. Note also that if you have a classic ASP Web site, you can skip this task.

Steps to execute

1. Ensure that all the prerequisites for a Content Delivery Server Role are met on your target system.

2. If this is a 32-bit .NET Web application running in combination with Java 6, then in the advanced system properties of your machine, select **Environment Variables** to see and edit your environment variables, and ensure that your `PATH` variable contains the `bin\` subfolder of the Java instance you are using. This location contains the file `MSVCR71.DLL`.

3. On your Windows machine, start up the Server Manager and do one of the following:

   - If this Web application runs under Windows 2012, start up the **Add Roles and Features Wizard**. Step through the wizard until you reach the **Features** screen. Now do one of the following:
     - If you use .NET 4.5, expand the item **.NET Framework 4.5 Features** and the subitem **WCF Services**.
     - Alternatively, if you use .NET 3.5, expand the item **.NET Framework 3.5 Features**.
   - Alternatively, if this Web application runs under Windows 2008 R2 SP1, under **Features Summary**, select **Add Features**. In the **Add Features** dialog that opens, expand the item **.NET Framework 3.5.1 Features** and the subitem **WCF Activation**.

   If the option **HTTP Activation** is not listed as installed, select it. Then click **Next** and **Install** to install the feature.

4. Access the SDL Tridion installation media and navigate to the `Content Delivery\roles\` folder.

5. Access the `preview\webservice\dotNET\webapp\` subfolder.

6. Unzip one of the following Zip files in this location to the location on your target system where you want to place the .NET Web application:

   - To install a 32-bit .NET Web application, unzip `x86.zip`.
   - To install a 64-bit .NET Web application, unzip `x86_64.zip`.

7. If you store published content in a database, download the JDBC driver (JAR file) required for your specific database vendor and copy it into the `bin\lib` folder of your Web application:

Database vendor	JDBC Driver	File to download
Microsoft SQL Server	Microsoft JDBC Driver 4.0 for SQL Server	`sqljdbc4.jar`
Oracle	Oracle JDBC driver 11.2.0.3	`ojdbc6.jar`

8. Navigate to the folder `\Content Delivery\roles\preview\webservice\configuration\samples\`.

9. Copy `logback.xml` from that folder to the `bin\config` folder of your Web application.

10. Copy each of the files that end in `_sample.xml` to the `bin\config` folder of your Web application, and rename them by removing the string `_sample` from each filename (for example, rename `cd_storage_conf_sample.xml` to `cd_storage_conf.xml`).

11. From the location in which you installed the Content Deployer, copy `cd_deployer_conf.xml` from that Server Role into the `bin\config` subfolder of your new Web application.

12. If you have SDL SmartTarget set up in one of your Content Delivery Server Roles, copy the following JAR files from that Server Role into the `bin\lib` subfolder of your new Web application:

    - `smarttarget_core.jar`
    - `smarttarget_entitymodel.jar`
    - All third-party library JAR files required by SmartTarget. You can find out what these files are by checking your SmartTarget installation media.

13. Copy `cd_licenses.xml`, your Content Delivery license file available from SDL Tridion Customer Support, to the `bin\config` directory of your Web application.

    If you want to name the file differently or put it in a different location, you need to update the `License` element in your configuration files.

14. Configure logging.

15. Configure the Storage Layer and optionally encrypt sensitive strings.

16. Open `Web.config` for editing in the root folder of this Web application.

17. To enable the Ambient Data Framework, do the following:

Application Pool in Classic Mode

Add the following inside the `<httpModules>` inside the `<system.web>` section (create an `<httpModules>` element if it does not yet exist):

```
<add type="Tridion.ContentDelivery.AmbientData.HttpModule" name="AmbientFrameworkModule" />
```

Application Pool in Integrated Mode

Add the following inside the `<modules>` inside the `<system.webServer>` section (create a `<modules>` element if it does not yet exist):

```
<add type="Tridion.ContentDelivery.AmbientData.HttpModule" name="AmbientFrameworkModule"
 preCondition="managedHandler" />
```

18. Save and close `Web.config`.

19. Disable recycling of the Application Pool of this Web application by doing the following:

    a. Open IIS Manager.

    b. From the tree on the left, select the current machine and navigate to the Web application you just created.

    c. Click **Advanced settings** to open a dialog with the application's properties, and note down the value of **Application Pool**.

    d. Close the dialog.

    e. From the tree on the left, under the current machine, now select the **Application Pools** node.

    f. From the list on the right, select the Application Pool you just noted down.

    g. Select **Recycling** in the **Edit Application Pool** area on the right.

    h. In the dialog that opes, deselect **Regular time intervals**, then click **Next** and **Finish** to commit your change.

    i. Restart IIS.

20. Implement security for the new Web application by following the instructions about securing the Web service in the SDL Tridion platform documentation. If you do not wish to secure the Web application (it is strongly recommended that you do), remove the `<Security>` section from `cd_ambient_conf.xml`.

21. Restart the Web application.

8.5.4 Adding the Experience Manager Web site extension to a .NET Web site

To add Experience Manager to your existing staging Web site in IIS, install the Experience Manager Web site extension as a .NET Web application in the staging Web site. Note also that if you have a classic ASP Web site, you can skip this task.

Steps to execute

1. Ensure that all the prerequisites for a Content Delivery Server Role are met on your target system.
2. If this is a 32-bit .NET Web application running in combination with Java 6, then in the advanced system properties of your machine, select **Environment Variables** to see and edit your environment variables, and ensure that your `PATH` variable contains the `bin\` subfolder of the Java instance you are using. This location contains the file `MSVCR71.DLL`.
3. Access the SDL Tridion installation media and navigate to the `Content Delivery\roles\` folder.
4. Access the `preview\web\dotNET\webapp\` subfolder.
5. Do one of the following:
   - To install a 32-bit .NET Web application, unzip the Zip file `x86.zip` in this location to the location on your target system where you want to place the .NET Web application.
   - To install a 64-bit .NET Web application, unzip the Zip file `x86_64.zip` in this location to the location on your target system where you want to place the .NET Web application.

   Typically, the place to install will be your existing staging Web site, but especially in a testing scenario, you may want to run this Web application by itself.
6. If you store published content in a database, download the JDBC driver (JAR file) required for your specific database vendor and copy it into the `WEB-INF/lib` directory of your Web application:

Database vendor	JDBC Driver	File to download
Microsoft SQL Server	Microsoft JDBC Driver 4.0 for SQL Server	`sqljdbc4.jar`
Oracle	Oracle JDBC driver 11.2.0.3	`ojdbc6.jar`

7. Navigate to the folder `\Content Delivery\roles\preview\web\configuration\samples\`.
8. Copy `logback.xml` from that folder to the `bin\config` folder of your Web application.

9. Copy each of the files that end in `_sample.xml` to the `bin\config` folder of your Web application, and rename them by removing the string `_sample` from each filename (for example, rename `cd_storage_conf_sample.xml` to `cd_storage_conf.xml`).

10. Open `cd_ambient_conf.xml` for editing.

11. Add the following lines inside the `<Cartridges>` section:

    ```
    <Cartridge File="cd_webservice_preview_cartridge.xml"/>
    <Cartridge File="footprint_cartridge_conf.xml"/>
    ```

12. Save and close `cd_ambient_conf.xml`.

13. Open `cd_dynamic_conf.xml` in a plain-text or XML editor, and insert the following structure as the first child element inside the root element. This section may already exist, in which case you can use the mappings you see, and add new ones if needed.

    ```
    <URLMappings>
      <StaticMappings>
        <Publications>
          <Publication Id="23">
            <Host Domain="example.com" Port="91" Protocol="http" Path="/local-fr" />
          </Publication>
        </Publications>
      </StaticMappings>
      <StorageMapping IdentifyPublicationByProperty="publicationUrl" />
    </URLMappings>
    ```

    This example makes it possible to retrieve metadata for a Web page located in the `http://example.com:91/local-fr/` by checking the corresponding Page in the Publication with ID 23 in the Content Manager.

14. Save and close `cd_dynamic_conf.xml`.

15. Copy `cd_licenses.xml`, your Content Delivery license file available from SDL Tridion Customer Support, to the `bin\config` directory of your Web application.

    If you want to name the file differently or put it in a different location, you need to update the `License` element in your configuration files.

16. Configure logging.

17. Configure the Storage Layer and optionally encrypt sensitive strings.

18. Open `Web.config` for editing in the root folder of this Web application.

19. To enable the Ambient Data Framework, do the following:

    Application Pool in Classic Mode

    Add the following inside the `<httpModules>` inside the `<system.web>` section (create an `<httpModules>` element if it does not yet exist):

```
<add type="Tridion.ContentDelivery.AmbientData.HttpModule" name="AmbientFrameworkModule" />
```

    Application Pool in Integrated Mode

    Add the following inside the `<modules>` inside the `<system.webServer>` section (create a `<modules>` element if it does not yet exist):

```
<add type="Tridion.ContentDelivery.AmbientData.HttpModule" name="AmbientFrameworkModule"
 preCondition="managedHandler" />
```

20. Find the location of your Ambient Data Framework module (in the `modules` section inside the `system.webServer` section of the file) and insert the following directly under it, if it is not already there:

```
<add name="PreviewContentModule" type="Tridion.ContentDelivery.Preview.Web.PreviewContentModule" />
```

21. Save and close `Web.config`.

22. Disable recycling of the Application Pool of this Web application by doing the following:

    a. Open IIS Manager.

    b. From the tree on the left, select the current machine and navigate to the Web application you just created.

    c. Click **Advanced settings** to open a dialog with the application's properties, and note down the value of **Application Pool**.

    d. Close the dialog.

    e. From the tree on the left, under the current machine, now select the **Application Pools** node.

    f. From the list on the right, select the Application Pool you just noted down.

    g. Select **Recycling** in the **Edit Application Pool** area on the right.

    h. In the dialog that opes, deselect **Regular time intervals**, then click **Next** and **Finish** to commit your change.

    i. Restart IIS.

23. Still in IIS Manager, select the staging Web site. Double-click **HTTP Reponse Headers** on the right. In the **Actions** panel that appears on the far right, click **Set Common Headers**. In the dialog that opens, check **Expire Web content** and set the values for **After** to **36500** and **Day(s)** (that is, 100 years from now). Click **OK** to commit your change, then close IIS Manager.

24. Restart the Web application.

8.5.5 Installing the API Server Role as a .NET Web application

Installing the API server role involves registering .NET Server Controls, configuring Profiling & Personalization, and setting up the Ambient Data Framework. The API Server Role is typically installed on the staging or live Web site, and takes care of resolving dynamic links between content items.

Steps to execute

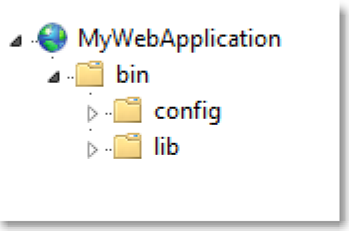
1. Ensure that all the prerequisites for a Content Delivery Server Role are met on your target system.

2. If this is a 32-bit .NET Web application running in combination with Java 6, then in the advanced system properties of your machine, select **Environment Variables** to see and edit your environment variables, and ensure that your `PATH` variable contains the `bin\` subfolder of the Java instance you are using. This location contains the file `MSVCR71.DLL`.

3. Create the following Web application directory structure:

   a. In the root folder, create a `bin\` subfolder.

   b. In the `bin\` folder, create subfolders `config\` and `lib\`.



4. Access the SDL Tridion installation media and navigate to the `Content Delivery\roles\` folder.

5. Access the `api\DotNET` subfolder.

6. Copy all DLLs in this folder to the `bin\` folder of your Web application.

7. Do one of the following:
   - To install a 32-bit .NET Web application, access the `x86\` subfolder.
   - To install a 64-bit .NET Web application, access the `x86_64\` subfolder.

8. Copy all DLLs in this folder to the `bin\` folder of your Web application.

9. On the installation media, navigate to the `Content Delivery\roles \api\java\` folder.

10. Copy all the JAR files from the `lib\` subfolder, and all the JAR files from the `third-party-lib\` subfolder, to the `bin\lib\` folder of your Web application.

11. If you store published content in a database, download the JDBC driver (JAR file) required for your specific database vendor and copy it into the `bin\lib` folder of your Web application:

Database vendor	JDBC Driver	File to download
Microsoft SQL Server	Microsoft JDBC Driver 4.0 for SQL Server	`sqljdbc4.jar`
Oracle	Oracle JDBC driver 11.2.0.3	`ojdbc6.jar`

12. Navigate to `\Content Delivery\resources\configurations\`.

13. Copy `logback.xml` from that folder to the `bin\config` folder of your Web application.

14. Copy and rename the following configuration files to the `bin\config` folder of your Web application:

Corresponding configuration file	Renamed configuration file
`cd_dynamic_conf_sample.xml`	`cd_dynamic_conf.xml`
`cd_storage_conf_sample.xml`	`cd_storage_conf.xml`
`cd_link_conf_sample.xml`	`cd_link_conf.xml`
`cd_wai_conf_sample.xml`	`cd_wai_conf.xml`

15. Copy `cd_licenses.xml`, your Content Delivery license file available from SDL Tridion Customer Support, to the `bin\config` directory of your Web application.

    If you want to name the file differently or put it in a different location, you need to update the `License` element in your configuration files.

16. Configure logging.

17. Configure the Storage Layer and optionally encrypt sensitive strings.

18. Disable recycling of the Application Pool of this Web application by doing the following:

    a. Open IIS Manager.

    b. From the tree on the left, select the current machine and navigate to the Web application you just created.

    c. Click **Advanced settings** to open a dialog with the application's properties, and note down the value of **Application Pool**.

    d. Close the dialog.

    e. From the tree on the left, under the current machine, now select the **Application Pools** node.

    f. From the list on the right, select the Application Pool you just noted down.

g. Select **Recycling** in the **Edit Application Pool** area on the right.

h. In the dialog that opes, deselect **Regular time intervals**, then click **Next** and **Finish** to commit your change.

i. Restart IIS.

19. Access the SDL Tridion installation media and from the folder `Content Delivery\resources\web\`, copy the file `Web.config` from that location to the root location of your Web application, and open it for editing.

20. Add a page control as follows:

```
<configuration>
 <system.web>
  <pages>
   <controls>
    <add tagPrefix="tridion"
         namespace="Tridion.ContentDelivery.Web.UI"
         assembly="Tridion.ContentDelivery" />
   </controls>
  </pages>
 </system.web>
</configuration>
```

21. If you intend to use Profiling & Personalization functionality, do one of the following:

Application Pool in Classic Mode

Add the following inside the `<httpModules>` inside the `<system.web>` section (create an `<httpModules>` element if it does not yet exist):

```
<add type="Tridion.ContentDelivery.Web.WAI.WAIModule, Tridion.ContentDelivery" name="TridionWAIHttpModule" />
```

Application Pool in Integrated Mode

- Add the following inside the `<system.webServer>` section:

```
<directoryBrowse enabled="true">
```

- Add the following inside the `<modules>` inside the `<system.webServer>` section (create a `<modules>` element if it does not yet exist):

```
<add type="Tridion.ContentDelivery.Web.WAI.WAIModule, Tridion.ContentDelivery" name="TridionWAIHttpModule" />
```

22. Save and close `Web.config`.

23. Restart the Web application.

8.6 Installing Server Roles as Java processes

To install a Server Role as a Java process, choose a Server Role to install and follow the instructions.

8.6.1 Installing the Content Deployer (other protocols) as a standalone Java process

To install the Content Deployer (other protocols) as a standalone Java process (on a Windows or non-Windows machine), copy resources and configuration files. You can also install the Content Deployer (other protocols) as a Web application.

Steps to execute

1. Open a command prompt.
2. Create a directory on your machine for the Content Deployer.
3. Create subdirectories `lib/` and `config/` in this directory.
4. Access the SDL Tridion installation media and navigate to the `Content Delivery\roles\` folder.
5. Navigate to the `deployer\java\` subfolder.
6. Access the `lib\` subfolder and copy all the JAR files into the `lib/` directory.
7. Access the `third-party-lib\` subfolder and copy all the JAR files into the `lib/` directory.
8. If you store published content in a database, download the JDBC driver (JAR file) required for your specific database vendor and copy it into the `lib/` directory:

Database vendor	JDBC Driver	File to download
Microsoft SQL Server	Microsoft JDBC Driver 4.0 for SQL Server	`sqljdbc4.jar`
Oracle	Oracle JDBC driver 11.2.0.3	`ojdbc6.jar`

9. Navigate to `\Content Delivery\resources\configurations\`.
10. Copy `logback.xml` from that folder to the `config/` directory.
11. Copy the Content Delivery license file you received from SDL Tridion Customer Support in the `config/` subdirectory as `cd_licenses.xml`. This license must include a Content Deployer license.
12. Copy the following configuration files to the `config/` subdirectory you created, and rename them:

Original configuration file	Renamed configuration file
cd_ambient_conf_sample.xml (if using the Ambient Data Framework)	cd_ambient_conf.xml
cd_deployer_conf_sample.xml	cd_deployer_conf.xml
cd_storage_conf_sample.xml	cd_storage_conf.xml
cd_link_conf_sample.xml	cd_link_conf.xml

13. Access the config/ subfolder and open the file cd_deployer_conf.xml in a plain-text or XML editor.

14. In the <Queue> section, add a Location element for each protocol over which you publish content to this Content Deployer, for example:

```
<Location Path="MYPATH" WindowSize="20" Workers="10" Cleanup="true" Interval="2s" />
```

For each Location element, specify the Path and optionally other attributes to optimize the publishing process:

Path

The absolute path to a location specified in a Publication Target as a relative path:

Protocol	Publication Target Property (relative path)
FTP	**Location**
SFTP (FTP over SSH)	**Path on the FTP server**
SSHFTP (SSH2 using an FTP subsystem)	**Location**
Local file copy	**Location**

For example, if the **Location** property specifies /dest/ for an FTP publishing destination, set Path to the absolute path to this location: c:\inetpub\ftproot\dest.

WindowSize

Sets the maximum number of Transport Package that can be "in flight" (publishing or deploying) at any one time. It represents the number of publish actions the Content Deployer should be able to handle. The default value for this attribute is 20.

Workers

Sets the maximum number of deployment threads that can be running at any time. If WindowSize is bigger than Workers, items to publish may enter a deployment queue, which in turn may cause a publish action to time out and fail. The default value for this attribute is 10.

Note: If you store metadata on the file system rather than in a database, you cannot configure multiple workers.

Interval

The amount of time, in milliseconds (ms), seconds (s), hours (h) and so on, that the Content Deployer waits before checking for new content in this location. The default value for this attribute is 2s; that is, two seconds.

15. If you use SDL Tridion JSP Custom Tags on your Web site:

    a. Open cd_deployer_conf.xml.

    b. Ensure that your <TCDLEngine> contains a <Properties> section containing the following property:

    ```
    <Property Name="tcdl.jsp.style" Value="tags" />
    ```

    c. Save and close cd_deployer_conf.xml.

16. Configure the Storage Layer and optionally encrypt sensitive strings.

17. In the directory you created, create a shell script file that places the config/ subdirectory, as well as all the JAR files, in the classpath and then runs the Java Content Deployer class:

    Linux/UNIX operating systems:

    ```
    #!/bin/sh
    TRIDION_CLASSPATH=./config
    for i in ./lib/*.jar ; do
      TRIDION_CLASSPATH=${TRIDION_CLASSPATH}:$i
    done
    java -cp $TRIDION_CLASSPATH com.tridion.deployer.Deployer
    ```

    Windows operating system:

    ```
    java -cp "./config;./lib/*" com.tridion.deployer.Deployer
    ```

18. Run this shell script to run the Content Deployer.

8.7 Installing Server Roles as Java Web applications

To install a Server Role as a Java Web application, choose a Server Role to install and follow the instructions.

8.7.1 Installing the Content Deployer (HTTP or HTTPS) as a Java Web application

Instead of installing the Content Deployer (HTTP or HTTPS) Server Role by deploying its WAR file and performing some manual steps.

Steps to execute

1. Ensure that all the prerequisites for a Content Delivery Server Role are met on your target system.

2. Access the SDL Tridion installation media and navigate to the `Content Delivery\roles\` folder.

3. Access the `upload\java\` subfolder.

4. Deploy the file `cd_upload.war` in this location as a new Java Web application.

5. If you store published content in a database, download the JDBC driver (JAR file) required for your specific database vendor and copy it into the `WEB-INF/lib` directory of your Web application:

Database vendor	JDBC Driver	File to download
Microsoft SQL Server	Microsoft JDBC Driver 4.0 for SQL Server	`sqljdbc4.jar`
Oracle	Oracle JDBC driver 11.2.0.3	`ojdbc6.jar`

6. Navigate to `\Content Delivery\resources\configurations\`.

7. Copy `logback.xml` from that folder to the `WEB-INF/classes/` subdirectory of your Web application.

8. Copy and rename the following configuration files to the `WEB-INF/classes` folder of your Web application:

Corresponding configuration file	Renamed configuration file
`cd_deployer_conf_sample.xml`	`cd_deployer_conf.xml`
`cd_storage_conf_sample.xml`	`cd_storage_conf.xml`
`cd_link_conf_sample.xml`	`cd_link_conf.xml`

9. Open the file `cd_deployer_conf.xml` for editing.

10. In the `<Queue>` section, add a `Location` element, for example:

```
<Location Path="MYPATH" WindowSize="20" Workers="10" Cleanup="true" Interval="2s" />
```

Specify the `Path` and optionally other attributes to optimize the publishing process:

`Path`

The absolute path to the pickup location.

`WindowSize`

The maximum number of Transport Package that can be "in flight" (publishing or deploying) at any one time. It represents the number of publish actions the Content Deployer should be able to handle. The default value for this attribute is 20.

`Workers`

The maximum number of deployment threads that can be running at any time. If `WindowSize` is bigger than `Workers`, items to publish may enter a deployment queue, which in turn may cause a publish action to time out and fail. The default value for this attribute is `10`.

Note: If you store metadata on the file system rather than in a database, you cannot configure multiple workers.

`Interval`

The amount of time, in milliseconds (`ms`), seconds (`s`), hours (`h`) and so on, that the Content Deployer waits before checking for new content in this location. The default value for this attribute is `2s`; that is, two seconds.

11. Directly below the `<Queue>` section, add the following element:

```
<HTTPSReceiver MaxSize="10000000" Location="MYPATH" InProcessDeploy="true" />
```

Configure the attributes as follows:

`MaxSize`

The maximum size (in bytes) of a Transport Package sent to the location.

`Location`

The same value as the *MYPATH* you specified in the `Location` element earlier.

`InProcessDeploy`

If you set this attribute to `true` (this is the case by default), publishing and deployment is contained within the Web application. If you set it to `false`, you must have your Content Manager publish content over HTTP or HTTPS to a location outside the Web application, and run the Content Deployer as a standalone Windows service or Java process. In this setup, the entire publishing transaction bypasses your Web application, which can improve your publishing performance. Refer to for a more detailed explanation of this implementation plan.

12. Save and close `cd_deployer_conf.xml`.

13. Copy `cd_licenses.xml`, your Content Delivery license file available from SDL Tridion Customer Support, to the `WEB-INF/classes` directory of your Web application.

    If you want to name the file differently or put it in a different location, you need to update the `License` element in your configuration files.

14. Configure logging.

15. Configure the Storage Layer and optionally encrypt sensitive strings.

16. If your Web application server is JBoss, do the following:

    a. Locate the following files in the `WEB-INF/lib/` subdirectory and remove them:

       - `stax-api.jar`

- `serializer.jar`
- `xalan.jar`

b. If you use JBoss 5.1, add the following JVM property-value pair to the `JAVA_OPTS` variable: `-Dorg.jboss.logging.provider=slf4j`

c. If you use JBoss 6, create a file with the following contents called `jboss-deployment-structure.xml` and save the file in the `WEB-INF/` directory:

```
<jboss-deployment-structure>
  <deployment>
    <exclusions>
      <module name="org.slf4j" />
      <module name="org.slf4j.impl" />
      <module name="com.sun.jersey" />
      <module name="com.sun.jersey.json" />
      <module name="com.sun.jersey.core" />
    </exclusions>
  </deployment>
</jboss-deployment-structure>
```

d. If you use JBoss 6, open `WEB-INF/web.xml` file, add the following section and save the file:

```
<context-param>
    <param-name>resteasy.scan</param-name>
    <param-value>false</param-value>
</context-param>
<context-param>
     <param-name>resteasy.scan.resources</param-name>
     <param-value>false</param-value>
</context-param>
<context-param>
      <param-name>resteasy.scan.providers</param-name>
      <param-value>false</param-value>
</context-param>
```

17. If your Web application server is WebLogic, do the following:

a. Remove the file `stax-api.jar` from your `WEB-INF/lib/` subdirectory.

b. Add a `weblogic.xml` file to the `WEB-INF` folder of your Web application that forces WebLogic to use the libraries contained in the Web application, rather than its own libraries. The contents of this `weblogic.xml` file must be as follows:

```
<weblogic-web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.bea.com/ns/weblogic/weblogic-web-app http://www.bea.com/ns/weblogic/weblogic-
web-app/1.0/weblogic-web-app.xsd"
  xmlns="http://www.bea.com/ns/weblogic/weblogic-web-app">
  <container-descriptor>
    <prefer-application-packages>
      <package-name>javax.persistence.**</package-name>
      <package-name>antlr.*</package-name>
      <package-name>org.slf4j.**</package-name>
      <package-name>org.apache.log4j.**</package-name>
      <package-name>org.apache.xml.serializer.**</package-name>
    </prefer-application-packages>
  </container-descriptor>
</weblogic-web-app>
```

18. Restart the Web application.

19. In an internet browser of your choice, navigate to `http://HOST:PORT/cd_upload/httpupload` where *HOST* and *PORT* refer to the location of the HTTP Upload Web application.

    The following text appears in your browser:

```
SDL Tridion HTTP Upload and Responder Servlet
```

8.7.2 Installing the Content Deployer (other protocols) as a Java Web application

Typically, if your protocol is not HTTP or HTTPS, you would install the Content Deployer (other protocols) as a Windows service (on Windows) or Java process (on non-Windows operating systems). Only if you have a pressing reason to do so, install Content Deployer as a Java Web application.

Requirements

The license file you use when installing this Server Role must include a Content Deployer license.

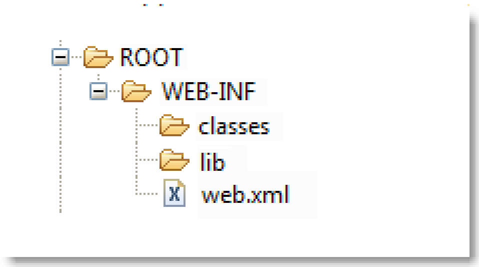
Steps to execute

1. Ensure that all the prerequisites for a Content Delivery Server Role are met on your target system.

2. Create the following Web application directory structure:

   a. In the root folder, create a `WEB-INF/` directory.

   b. In the `WEB-INF/` directory, create subdirectories `classes/` and `lib/`.



```
ROOT
  WEB-INF
    classes
    lib
    web.xml
```

3. On the installation media, navigate to the `Content Delivery\roles\deployer\java\` folder.

4. Copy all the JAR files from the `lib\` subfolder, and all the JAR files from the `third-party-lib\` subfolder, to the `WEB-INF/lib/` folder of your Web application.

5. If you store published content in a database, download the JDBC driver (JAR file) required for your specific database vendor and copy it into the `WEB-INF/lib` directory of your Web application:

Database vendor	JDBC Driver	File to download
Microsoft SQL Server	Microsoft JDBC Driver 4.0 for SQL Server	`sqljdbc4.jar`

Database vendor	JDBC Driver	File to download
Oracle	Oracle JDBC driver 11.2.0.3	`ojdbc6.jar`

6. Navigate to `\Content Delivery\resources\configurations\`.

7. Copy `logback.xml` from that folder to the `WEB-INF/classes/` subdirectory of your Web application.

8. Copy and rename the following configuration files from that folder to the `WEB-INF/classes/` subdirectory of your Web application:

Original configuration file	Renamed configuration file
`cd_deployer_conf_sample.xml`	`cd_deployer_conf.xml`
`cd_storage_conf_sample.xml`	`cd_storage_conf.xml`
`cd_link_conf_sample.xml`	`cd_link_conf.xml`

9. Open the file `cd_deployer_conf.xml` for editing.

10. In the `<Queue>` section, add a `Location` element for each protocol over which you publish content to this Content Deployer, for example:

```
<Location Path="MYPATH" WindowSize="20" Workers="10" Cleanup="true" Interval="2s" />
```

For each `Location` element, specify the `Path` and optionally other attributes to optimize the publishing process:

`Path`

The absolute path to a location specified in a Publication Target as a relative path:

Protocol	Publication Target Property (relative path)
FTP	**Location**
SFTP (FTP over SSH)	**Path on the FTP server**
SSHFTP (SSH2 using an FTP subsystem)	**Location**
Local file copy	**Location**

For example, if the **Location** property specifies `/dest/` for an FTP publishing destination, set `Path` to the absolute path to this location: `c:\inetpub\ftproot\dest`.

`WindowSize`

Sets the maximum number of Transport Package that can be "in flight" (publishing or deploying) at any one time. It represents the number of publish actions the Content Deployer should be able to handle. The default value for this attribute is 20.

`Workers`

Sets the maximum number of deployment threads that can be running at any time. If `WindowSize` is bigger than `Workers`, items to publish may enter a deployment queue, which in turn may cause a publish action to time out and fail. The default value for this attribute is `10`.

Note: If you store metadata on the file system rather than in a database, you cannot configure multiple workers.

`Interval`

The amount of time, in milliseconds (`ms`), seconds (`s`), hours (`h`) and so on, that the Content Deployer waits before checking for new content in this location. The default value for this attribute is `2s`; that is, two seconds.

11. If you use SDL Tridion JSP Custom Tags on your Web site, ensure that your `<TCDLEngine>` contains a `<Properties>` section containing the following property:

```
<Property Name="tcdl.jsp.style" Value="tags" />
```

12. Save and close `cd_deployer_conf.xml`.

13. Copy `cd_licenses.xml`, your Content Delivery license file available from SDL Tridion Customer Support, to the `WEB-INF/classes` directory of your Web application.

    If you want to name the file differently or put it in a different location, you need to update the `License` element in your configuration files.

14. Configure logging.

15. Configure the Storage Layer and optionally encrypt sensitive strings.

16. If your Web application server is JBoss, do the following:

    a. Locate the following files in the `WEB-INF/lib/` subdirectory and remove them:

    - `stax-api.jar`
    - `serializer.jar`
    - `xalan.jar`

    b. If you use JBoss 5.1, add the following JVM property-value pair to the `JAVA_OPTS` variable: `-Dorg.jboss.logging.provider=slf4j`

    c. If you use JBoss 6, create a file with the following contents called `jboss-deployment-structure.xml` and save the file in the `WEB-INF/` directory:

```
<jboss-deployment-structure>
  <deployment>
    <exclusions>
      <module name="org.slf4j" />
      <module name="org.slf4j.impl" />
      <module name="com.sun.jersey" />
      <module name="com.sun.jersey.json" />
      <module name="com.sun.jersey.core" />
    </exclusions>
  </deployment>
</jboss-deployment-structure>
```

d. If you use JBoss 6, open `WEB-INF/web.xml` file, add the following section and save the file:

```
<context-param>
    <param-name>resteasy.scan</param-name>
    <param-value>false</param-value>
</context-param>
<context-param>
     <param-name>resteasy.scan.resources</param-name>
     <param-value>false</param-value>
</context-param>
<context-param>
      <param-name>resteasy.scan.providers</param-name>
      <param-value>false</param-value>
</context-param>
```

17. If your Web application server is WebLogic, do the following:

a. Remove the file `stax-api.jar` from your `WEB-INF/lib/` subdirectory.

b. Add a `weblogic.xml` file to the `WEB-INF` folder of your Web application that forces WebLogic to use the libraries contained in the Web application, rather than its own libraries. The contents of this `weblogic.xml` file must be as follows:

```
<weblogic-web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.bea.com/ns/weblogic/weblogic-web-app http://www.bea.com/ns/weblogic/weblogic-
web-app/1.0/weblogic-web-app.xsd"
  xmlns="http://www.bea.com/ns/weblogic/weblogic-web-app">
  <container-descriptor>
    <prefer-application-packages>
      <package-name>javax.persistence.**</package-name>
      <package-name>antlr.*</package-name>
      <package-name>org.slf4j.**</package-name>
      <package-name>org.apache.log4j.**</package-name>
      <package-name>org.apache.xml.serializer.**</package-name>
    </prefer-application-packages>
  </container-descriptor>
</weblogic-web-app>
```

18. Restart the Web application.

8.7.3 Installing the Experience Manager Web service as a Java Web application

As part of installing Experience Manager , install the Experience Manager Web service. Note that this Web application also contains the Content Delivery Web service Server Role. Note that if you have a classic ASP Web site, you can skip this task.

Steps to execute

1. Ensure that all the prerequisites for a Content Delivery Server Role are met on your target system.
2. Access the SDL Tridion installation media and navigate to the `Content Delivery\roles\` folder.
3. Access the `preview\webservice\java\webapp\` subfolder.

4. Deploy the file `cd_preview_webservice.war` in this location as a new Java Web application.

5. If you store published content in a database, download the JDBC driver (JAR file) required for your specific database vendor and copy it into the `WEB-INF/lib` directory of your Web application:

Database vendor	JDBC Driver	File to download
Microsoft SQL Server	Microsoft JDBC Driver 4.0 for SQL Server	`sqljdbc4.jar`
Oracle	Oracle JDBC driver 11.2.0.3	`ojdbc6.jar`

6. Navigate to the folder `\Content Delivery\roles\preview\webservice\configuration\samples\`.

7. Copy `logback.xml` from that folder to the `WEB-INF/classes/` subdirectory of your Web application.

8. Copy each of the files that end in `_sample.xml` to the `WEB-INF/classes` folder of your Web application, and rename them by removing the string `_sample` from each filename (for example, rename `cd_storage_conf_sample.xml` to `cd_storage_conf.xml`).

9. From the location in which you installed the Content Deployer, copy `cd_deployer_conf.xml` from that Server Role into the `WEB-INF/classes/` subfolder of your new Web application.

10. If you have SDL SmartTarget set up in one of your Content Delivery Server Roles, copy the following JAR files from that Server Role into the `WEB-INF/lib` subfolder of your new Web application:
    - `smarttarget_core.jar`
    - `smarttarget_entitymodel.jar`
    - All third-party library JAR files required by SmartTarget. You can find out what these files are by checking your SmartTarget installation media.

11. Copy `cd_licenses.xml`, your Content Delivery license file available from SDL Tridion Customer Support, to the `WEB-INF/classes` directory of your Web application.

    If you want to name the file differently or put it in a different location, you need to update the `License` element in your configuration files.

12. Configure logging.

13. Configure the Storage Layer and optionally encrypt sensitive strings.

14. Implement security for the new Web application by following the instructions about securing the Web service in the SDL Tridion platform documentation. If you do not wish to secure the Web application (it is strongly recommended that you do), remove the `<Security>` section from `cd_ambient_conf.xml`.

15. Enable the Ambient Data Framework by adding the following to your `web.xml`:

```
<filter>
  <filter-name>Ambient Data Framework</filter-name>
  <filter-class>com.tridion.ambientdata.web.AmbientDataServletFilter</filter-class>
</filter>
<filter-mapping>
  <filter-name>Ambient Data Framework</filter-name>
  <servlet-name>Content Delivery Web service</servlet-name>
</filter-mapping>
```

This enables the Ambient Data Framework for the Content Delivery Web service only. If you want to enable the Ambient Data Framework in the entire Web application, replace the `servlet-name` element with the following:

```
<url-pattern>/*</url-pattern>
```

16. If your Web application server is JBoss, do the following:

    a. Locate the following files in the `WEB-INF/lib/` subdirectory and remove them:

    - `stax-api.jar`
    - `serializer.jar`
    - `xalan.jar`

    b. If you use JBoss 5.1, add the following JVM property-value pair to the `JAVA_OPTS` variable: `-Dorg.jboss.logging.provider=slf4j`

    c. If you use JBoss 6, create a file with the following contents called `jboss-deployment-structure.xml` and save the file in the `WEB-INF/` directory:

```
<jboss-deployment-structure>
  <deployment>
    <exclusions>
      <module name="org.slf4j" />
      <module name="org.slf4j.impl" />
      <module name="com.sun.jersey" />
      <module name="com.sun.jersey.json" />
      <module name="com.sun.jersey.core" />
    </exclusions>
  </deployment>
</jboss-deployment-structure>
```

    d. If you use JBoss 6, open `WEB-INF/web.xml` file, add the following section and save the file:

```
<context-param>
    <param-name>resteasy.scan</param-name>
    <param-value>false</param-value>
</context-param>
<context-param>
     <param-name>resteasy.scan.resources</param-name>
     <param-value>false</param-value>
</context-param>
<context-param>
      <param-name>resteasy.scan.providers</param-name>
      <param-value>false</param-value>
</context-param>
```

17. If your Web application server is WebLogic, do the following:

    a. Remove the file `stax-api.jar` from your `WEB-INF/lib/` subdirectory.

b. Add a `weblogic.xml` file to the `WEB-INF` folder of your Web application that forces WebLogic to use the libraries contained in the Web application, rather than its own libraries. The contents of this `weblogic.xml` file must be as follows:

```
<weblogic-web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.bea.com/ns/weblogic/weblogic-web-app http://www.bea.com/ns/weblogic/weblogic-
web-app/1.0/weblogic-web-app.xsd"
  xmlns="http://www.bea.com/ns/weblogic/weblogic-web-app">
  <container-descriptor>
    <prefer-application-packages>
      <package-name>javax.persistence.**</package-name>
      <package-name>antlr.*</package-name>
      <package-name>org.slf4j.**</package-name>
      <package-name>org.apache.log4j.**</package-name>
      <package-name>org.apache.xml.serializer.**</package-name>
    </prefer-application-packages>
  </container-descriptor>
</weblogic-web-app>
```

18. Restart the Web application.

8.7.4 Adding the Experience Manager Web site extension to a JSP Web site

You can add the Experience Manager Web site extension to your existing staging Web site, or you can create a standalone Web site that contains the Experience Manager Web site extension. You would typically do the latter for testing purposes.

Steps to execute

1. Ensure that all the prerequisites for a Content Delivery Server Role are met on your target system.

2. Access the SDL Tridion installation media and navigate to the `Content Delivery\roles\` folder.

3. Access the `preview\web\java\webapp\` subfolder.

4. Do one of the following with the file `cd_preview_web.war` in this location:

   - In a test scenario, deploy the WAR file on its own. Note that by default, this Web site is not secure because it is considered a staging Web site, and so it is expected to be behind a firewall or otherwise protected.
   - To add the Experience Manager Web site extension to your staging Web site, add its contents to the Web site.

5. If you store published content in a database, download the JDBC driver (JAR file) required for your specific database vendor and copy it into the `WEB-INF/lib` directory of your Web application:

Database vendor	JDBC Driver	File to download
Microsoft SQL Server	Microsoft JDBC Driver 4.0 for SQL Server	`sqljdbc4.jar`
Oracle	Oracle JDBC driver 11.2.0.3	`ojdbc6.jar`

6. Navigate to the folder `\Content Delivery\roles\preview\web\configuration\samples\`.

7. Copy `logback.xml` from that folder to the `WEB-INF/classes/` subdirectory of your Web application.

8. Copy each of the files that end in `_sample.xml` to the `WEB-INF/classes` folder of your Web application, and rename them by removing the string `_sample` from each filename (for example, rename `cd_storage_conf_sample.xml` to `cd_storage_conf.xml`).

9. Open `cd_ambient_conf.xml` for editing.

10. Find the `<Security>` section and enclose it in comments. This element is only needed if you intend to set up OAuth authentication.

11. Add the following lines inside the `<Cartridges>` section:

```
<Cartridge File="cd_webservice_preview_cartridge.xml"/>
<Cartridge File="footprint_cartridge_conf.xml"/>
```

12. Save and close `cd_ambient_conf.xml`.

13. Open `cd_dynamic_conf.xml` in a plain-text or XML editor, and insert the following structure as the first child element inside the root element. This section may already exist, in which case you can use the mappings you see, and add new ones if needed.

```
<URLMappings>
  <StaticMappings>
    <Publications>
      <Publication Id="23">
        <Host Domain="example.com" Port="91" Protocol="http" Path="/local-fr" />
      </Publication>
    </Publications>
  </StaticMappings>
  <StorageMapping IdentifyPublicationByProperty="publicationUrl" />
</URLMappings>
```

This example makes it possible to retrieve metadata for a Web page located in the `http://example.com:91/local-fr/` by checking the corresponding Page in the Publication with ID 23 in the Content Manager.

14. Save and close `cd_dynamic_conf.xml`.

15. Copy `cd_licenses.xml`, your Content Delivery license file available from SDL Tridion Customer Support, to the `WEB-INF/classes` directory of your Web application.

    If you want to name the file differently or put it in a different location, you need to update the `License` element in your configuration files.

16. Configure logging.

17. Configure the Storage Layer and optionally encrypt sensitive strings.

18. If your Web application server is JBoss, do the following:

    a. Locate the following files in the `WEB-INF/lib/` subdirectory and remove them:

    - `stax-api.jar`
    - `serializer.jar`
    - `xalan.jar`

    b. If you use JBoss 5.1, add the following JVM property-value pair to the `JAVA_OPTS` variable: `-Dorg.jboss.logging.provider=slf4j`

    c. If you use JBoss 6, create a file with the following contents called `jboss-deployment-structure.xml` and save the file in the `WEB-INF/` directory:

```
<jboss-deployment-structure>
  <deployment>
    <exclusions>
      <module name="org.slf4j" />
      <module name="org.slf4j.impl" />
      <module name="com.sun.jersey" />
      <module name="com.sun.jersey.json" />
      <module name="com.sun.jersey.core" />
    </exclusions>
  </deployment>
</jboss-deployment-structure>
```

    d. If you use JBoss 6, open `WEB-INF/web.xml` file, add the following section and save the file:

```
<context-param>
    <param-name>resteasy.scan</param-name>
    <param-value>false</param-value>
</context-param>
<context-param>
     <param-name>resteasy.scan.resources</param-name>
     <param-value>false</param-value>
</context-param>
<context-param>
      <param-name>resteasy.scan.providers</param-name>
      <param-value>false</param-value>
</context-param>
```

19. If your Web application server is WebLogic, do the following:

    a. Remove the file `stax-api.jar` from your `WEB-INF/lib/` subdirectory.

    b. Add a `weblogic.xml` file to the `WEB-INF` folder of your Web application that forces WebLogic to use the libraries contained in the Web application, rather than its own libraries. The contents of this `weblogic.xml` file must be as follows:

```
<weblogic-web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.bea.com/ns/weblogic/weblogic-web-app http://www.bea.com/ns/weblogic/weblogic-web-app/1.0/weblogic-web-app.xsd"
  xmlns="http://www.bea.com/ns/weblogic/weblogic-web-app">
  <container-descriptor>
    <prefer-application-packages>
      <package-name>javax.persistence.**</package-name>
      <package-name>antlr.*</package-name>
      <package-name>org.slf4j.**</package-name>
      <package-name>org.apache.log4j.**</package-name>
      <package-name>org.apache.xml.serializer.**</package-name>
    </prefer-application-packages>
  </container-descriptor>
</weblogic-web-app>
```

20. Access the root of your staging Web site and open `web.xml` in a plain-text or XML editor.

21. Enable the Ambient Data Framework by adding the following to your `web.xml`:

```
<filter>
  <filter-name>Ambient Data Framework</filter-name>
  <filter-class>com.tridion.ambientdata.web.AmbientDataServletFilter</filter-class>
</filter>
<filter-mapping>
  <filter-name>Ambient Data Framework</filter-name>
  <servlet-name>Content Delivery Web service</servlet-name>
</filter-mapping>
```

    This enables the Ambient Data Framework for the Content Delivery Web service only. If you want to enable the Ambient Data Framework in the entire Web application, replace the `servlet-name` element with the following:

```
<url-pattern>/*</url-pattern>
```

22. Insert the following directly under the Ambient Data Framework filter:

```
<listener>
  <listener-class>
    com.tridion.storage.persistence.session.SessionManagementContextListener
  </listener-class>
</listener>

<filter>
  <filter-name>Page Content Filter</filter-name>
  <filter-class>com.tridion.preview.web.PageContentFilter</filter-class>
</filter>

<filter-mapping>
  <filter-name>Page Content Filter</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>

<filter>
 <filter-name>Binary Content Filter</filter-name>
 <filter-class>com.tridion.preview.web.BinaryContentFilter</filter-class>
</filter>

<filter-mapping>
 <filter-name>Binary Content Filter</filter-name>
 <url-pattern>/*</url-pattern>
</filter-mapping>
```

    This applies session preview filtering to your entire Web site. If your entire staging Web is located in a subdirectory, change the value of `url-pattern`.

23. Save and close `web.xml` and restart your JSP Web site.

24. Restart the Web application.

8.7.5 Installing the API Server Role for a Java Web application

The API Server Role is typically installed on the staging or live Web site, and takes care of resolving dynamic links between content items.

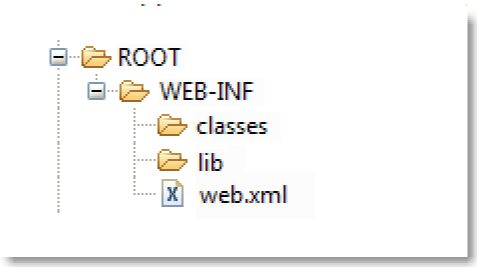
Steps to execute

1. Ensure that all the prerequisites for a Content Delivery Server Role are met on your target system.

2. Create the following Web application directory structure:

   a. In the root folder, create a `WEB-INF/` directory.

   b. In the `WEB-INF/` directory, create subdirectories `classes/` and `lib/`.



3. Access the SDL Tridion installation media and navigate to the `Content Delivery\roles\` folder.

4. Access the `api\java\` subfolder.

5. Copy all the JAR files from the `lib\` subfolder, and all the JAR files from the `third-party-lib\` subfolder, to the `WEB-INF/lib/` folder of your Web application.

6. If you store published content in a database, download the JDBC driver (JAR file) required for your specific database vendor and copy it into the `WEB-INF/lib` directory of your Web application:

Database vendor	JDBC Driver	File to download
Microsoft SQL Server	Microsoft JDBC Driver 4.0 for SQL Server	`sqljdbc4.jar`
Oracle	Oracle JDBC driver 11.2.0.3	`ojdbc6.jar`

7. Navigate to `\Content Delivery\resources\configurations\`.

8. Copy `logback.xml` from that folder to the `bin\config` folder of your Web application.

9. Copy and rename the following configuration files to the `WEB-INF/classes/` folder of your Web application:

Corresponding configuration file	Renamed configuration file
`cd_dynamic_conf_sample.xml`	`cd_dynamic_conf.xml`
`cd_storage_conf_sample.xml`	`cd_storage_conf.xml`
`cd_link_conf_sample.xml`	`cd_link_conf.xml`
`cd_wai_conf_sample.xml`	`cd_wai_conf.xml`

10. Copy `cd_licenses.xml`, your Content Delivery license file available from SDL Tridion Customer Support, to the `WEB-INF/classes` directory of your Web application.

    If you want to name the file differently or put it in a different location, you need to update the `License` element in your configuration files.

11. Configure logging.

12. Configure the Storage Layer and optionally encrypt sensitive strings.

13. If your Web application server is JBoss, do the following:

    a. Locate the following files in the `WEB-INF/lib/` subdirectory and remove them:

       - `stax-api.jar`
       - `serializer.jar`
       - `xalan.jar`

    b. If you use JBoss 5.1, add the following JVM property-value pair to the `JAVA_OPTS` variable: `-Dorg.jboss.logging.provider=slf4j`

    c. If you use JBoss 6, create a file with the following contents called `jboss-deployment-structure.xml` and save the file in the `WEB-INF/` directory:

```
<jboss-deployment-structure>
  <deployment>
    <exclusions>
      <module name="org.slf4j" />
      <module name="org.slf4j.impl" />
      <module name="com.sun.jersey" />
      <module name="com.sun.jersey.json" />
      <module name="com.sun.jersey.core" />
    </exclusions>
  </deployment>
</jboss-deployment-structure>
```

    d. If you use JBoss 6, open `WEB-INF/web.xml` file, add the following section and save the file:

```
<context-param>
    <param-name>resteasy.scan</param-name>
    <param-value>false</param-value>
</context-param>
<context-param>
     <param-name>resteasy.scan.resources</param-name>
     <param-value>false</param-value>
</context-param>
<context-param>
      <param-name>resteasy.scan.providers</param-name>
      <param-value>false</param-value>
</context-param>
```

14. If your Web application server is WebLogic, do the following:

   a. Remove the file `stax-api.jar` from your `WEB-INF/lib/` subdirectory.

   b. Add a `weblogic.xml` file to the `WEB-INF` folder of your Web application that forces WebLogic to use the libraries contained in the Web application, rather than its own libraries. The contents of this `weblogic.xml` file must be as follows:

```
<weblogic-web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.bea.com/ns/weblogic/weblogic-web-app http://www.bea.com/ns/weblogic/weblogic-
web-app/1.0/weblogic-web-app.xsd"
  xmlns="http://www.bea.com/ns/weblogic/weblogic-web-app">
  <container-descriptor>
    <prefer-application-packages>
      <package-name>javax.persistence.**</package-name>
      <package-name>antlr.*</package-name>
      <package-name>org.slf4j.**</package-name>
      <package-name>org.apache.log4j.**</package-name>
      <package-name>org.apache.xml.serializer.**</package-name>
    </prefer-application-packages>
  </container-descriptor>
</weblogic-web-app>
```

15. Access the root of your Web application and copy the file `cd_tags.tld` (the tag library) from the `Content Delivery\resources\tld\` folder on the SDL Tridion installation media to the `WEB-INF/lib/` subdirectory of your Web application.

16. Add the following XML fragment to the `web.xml` file of the Web application:

    ```
    <jsp-config>
      <taglib>
        <taglib-uri>cd_tags</taglib-uri>
        <taglib-location>/WEB-INF/lib/cd_tags.tld</taglib-location>
     </taglib>
    </jsp-config>
    ```

17. Save and close `web.xml`.

18. Restart the Web application.

8.8 Configuring logging on the Content Delivery server

The `logback.xml` file is the configuration file for the logback logging framework used by Content Delivery.

Context

Note: For detailed information about the Logback framework and its configuration, refer to http://logback.qos.ch/documentation.html.

Steps to execute

1. Open the file `logback.xml` in a plain-text editor or XML editor:

- On Content Delivery, the logback.xml file is located in the config/ (.NET) or /classes (Java) subfolder if Content Delivery is running as a Web application, otherwise in the root of your Content Delivery installation.

2. Set the following properties:

```
<property name="log.pattern" value="%date %-5level %logger{0} - %message%n"/>
<property name="log.history" value="7"/>
<property name="log.folder" value="c:/tridion/log"/>
<property name="log.level" value="WARN"/>
```

log.pattern (logging filename format)

The log.pattern property specifies the name of the log files using a format string. Refer to the online Logback documentation for more information.

log.history (number of log files)

The log.history property specifies the number of log files to back up (the default is 7 files).

log.folder (logging location)

The log.folder property specifies the location where log files are saved (defaults to c:/Tridion/log).

For example, on the Content Manager Server set the log location to: c:/Program Files (x86)/Tridion/log.

log.level (logging level)

To change the default logging level, change the value of the log.level property to one of the values describes in the following table. Note that logging levels are incremental and inclusive; each level logs data for the previous levels. For example, DEBUG also logs and events covered by INFO, WARN, and ERROR.

Logging Level	Description
OFF	Turn off logging.
ERROR (default)	Log events that cause a specific feature of the application to fail.
WARN	Log events that are potentially harmful.
INFO	Log neutral, informational messages about high-level functionality being executed.
DEBUG	Log fine-grained informational messages about low-level functionality being executed, intended to discover the source of a problem.
TRACE	Log highly detailed informational messages about the application's lowest-level activity,
ALL	Maximum logging level that logs everything.

Note: The `DEBUG` level is intended for hunting down the origin of a particular problem only and will result in a significant performance drop while this level is set. Therefore, it is recommended to set the logging level back to as soon as the problem has been resolved.

3. If you want to use property substitutions, that is, variables resolved by context such as environment variables, the Logback framework allows you to do so. The simplest way to do this is to define an environment variable (say `Foo`) wherever you set up Content Delivery, and insert the string `${Foo}` in your logging configuration file. Logback now substitutes the string with the value of the environment variable. This approach enables you to reuse the same configuration file (and, say, deploy a WAR file without having to edit this file). For more information, refer to the heading "Property substitution" on this Web page: http://logback.qos.ch/manual/configuration.html.

4. Save and close `logback.xml`.

8.9 Setting up storage

You need to specify where published content is stored or retrieved in the Storage Layer configuration file `cd_storage_conf.xml` file.

Context

Note: For information on setting up caching, see the implementation topics for Content Delivery.

8.9.1 Configuring content storage

The `cd_storage_conf.xml` file is the Content Delivery Storage Layer configuration file. This task explains how to configure standard storage scenarios.

Context

When the Content Deployer receives incoming content it deploys that content to the Storage Layer which then stores the content in the Content Data Store. The Content Data Store is the local file system and/or a relational database.

Steps to execute

1. Open `cd_storage_conf.xml` located in the `config\` subdirectory of your Content Delivery root location in a plain-text editor or XML editor.

2. In the `Storages` element, configure one or more of the following storage mechanisms:

Note: By default, the sample Storage Layer configuration file provided with SDL Tridion is configured to store all content to the local file system in the folder `c:\temp`.

- To enable storage of content in a **Microsoft SQL Server database**, ensure the presence of the following `<Storage>` element, uncommented:

```
<Storage Type="persistence" Id="MSSQL_01" dialect="MSSQL"
  Class="com.tridion.storage.persistence.JPADAOFactory">
  <Pool Type="jdbc" Size="5" MonitorInterval="60" IdleTimeout="120" CheckoutTimeout="120" />
  <DataSource Class="com.microsoft.sqlserver.jdbc.SQLServerDataSource">
    <Property Name="serverName" Value="SERVER_NAME" />
    <Property Name="portNumber" Value="1433" />
    <Property Name="databaseName" Value="DATABASE_NAME" />
    <Property Name="user" Value="USER_NAME" />
    <Property Name="password" Value="PASSWORD" />
  </DataSource>
</Storage>
```

- To enable storage of content in an **Oracle database**, ensure the presence of the following `<Storage>` element, uncommented:

```
<Storage Type="persistence" Id="Oracle_01" dialect="ORACLESQL"
  Class="com.tridion.storage.persistence.JPADAOFactory">
  <Pool Type="jdbc" Size="5" MonitorInterval="60" IdleTimeout="120" CheckoutTimeout="120" />
  <DataSource class="oracle.jdbc.pool.OracleDataSource">
    <Property Name="driverType" Value="thin" />
    <Property Name="networkProtocol" Value="tcp" />
    <Property Name="serverName" Value="SERVER_NAME" />
    <Property Name="portNumber" Value="1521" />
    <Property Name="databaseName" Value="DATABASE_NAME" />
    <Property Name="user" Value="USER_NAME" />
    <Property Name="password" Value="PASSWORD" />
  </DataSource>
</Storage>
```

- To enable storage of content on the **local file system**, ensure the presence of the following `<Storage>` element, uncommented:

```
<Storage Type="filesystem" Id="LocalFileSystem_01" defaultFilesystem="true" defaultStorage="true"
  Class="com.tridion.storage.filesystem.FSDAOFactory">
  <Root Path="STORAGE_PATH" />
</Storage>
```

  where `STORAGE_PATH` is the fully qualified path to the root folder of your storage location.
- To enable storage through **JNDI** (that is, storage in an Oracle or Microsoft SQL Server database via your Web application server), ensure the presence of the following `<Storage>` element, uncommented:

  ```
  <Storage Type="persistence" Id="JNDI_01" dialect="DIALECT_TYPE"
    Class="com.tridion.storage.persistence.JPADAOFactory" JNDIName="JNDI_NAME" />
  ```

  where `DIALECT_TYPE` is `ORACLESQL` (for storage in an Oracle database)

  or `MSSQL` (for storage in a SQL Server database), and `JNDI_NAME` is the identifier of your data source (the exact inner structure of this string depends on the Web application server you use).

Note: The values for the `Id` attributes of the `<Storage>` elements (`MSSQL_01`, `Oracle_01` and `JNDI_01`) are arbitrary. They only need to be unique strings.

Note: If you are running a multi-threaded Content Deployer, be aware that the value of the `Size` attribute in the `Pool` element can impair the effect of setting up a multi-threaded Content Deployer. You may need to increase value of the `Size` attribute.

3. Provide values for the properties by replacing the following strings in the Value attributes of the Property elements:

   For database storage, specify the following:

   SERVER_NAME

   Replace this string with the name of the database server host.

   DATABASE_NAME

   Replace this string with the name of the database to access.

   USER_NAME

   Replace this string with the name of a user who can access the database.

   PASSWORD

   Replace this string with a valid password for the user you specified.

   Note: You can add as many <Storage> elements as you like.

4. Select one of the storage media you configured (a database or a file system location) as your default storage destination in the ItemTypes element, located directly after the Global section:

   ```
   <ItemTypes defaultStorageId="STORAGE_ID" cached="CACHE_BEHAVIOR">
   </ItemTypes>
   ```

   Replace STORAGE_ID with a value of the Id attribute of one of your Storage elements (for example, MSSQL_01). Also, set cached to true or false depending on your preference. Typically, in a development environment you would set cached to false and in a live environment to true.

   Content Delivery will now store all content in the storage medium identified by the storage ID you specified, and cache or not cache it (depending on your value for cached).

5. You now have a number of ways in which you can override this default behavior:

   - You can change whether or not items of a certain type are cached. To do this, insert an Item element inside the ItemTypes element and set the value of cached to the reverse of the value specified in ItemTypes:

     ```
     <Item typeMapping="TYPE_NAME" cached="CACHE_BEHAVIOR" storageId="STORAGE_ID"/>
     ```

   - You can store items of a certain type in a different storage medium. To do this, insert an Item element inside the ItemTypes element and set the value of storageId to another value than the one specified in ItemTypes:

     ```
     <Item typeMapping="TYPE_NAME" cached="CACHE_BEHAVIOR" storageId="STORAGE_ID"/>
     ```

   - You can store specific types of binary items (that is, binary files with specific file extensions) in a different storage medium. To do this, insert an Item element inside the ItemTypes element:

```
<Item typeMapping="Binary" itemExtension=".EXT" cached="CACHE_BEHAVIOR" storageId="STORAGE_ID" />
```

- You can store specific types of Pages (that is, Web pages with specific file extensions) in a different storage medium. To do this, insert an `Item` element inside the `ItemTypes` element:

```
<Item typeMapping="Page" itemExtension=".Jsp" cached="CACHE_BEHAVIOR" storageId="STORAGE_ID" />
```

  Other possible values for `itemExtension` are `.Aspx` and `.html`
- You can store specific types of Component Presentations (that is, Component Presentations with specific file extensions) in a different storage medium. To do this, insert an `Item` element inside the `ItemTypes` element:

```
<Item typeMapping="ComponentPresentation" itemExtension=".Jsp" cached="CACHE_BEHAVIOR"
 storageId="STORAGE_ID" />
```

  Possible values for `itemExtension` are:

  - `.Jsp`
  - `.Ascx`
  - `.Xml`
  - `.Txt`
  - `.rel`
- You can specify a different storage medium and/or caching behavior for items that were published from a specific Publication in the Content Manager, by inserting a `Publication` element inside the `ItemTypes` element, directly after any `Item` elements:

  ```
  <Publication Id="PUB_ID" defaultStorageId="STORAGE_ID" cached="CACHE_BEHAVIOR">
  </Publication>
  ```
- You can specify a different storage medium and/or caching behavior for items of a certain type that were published from a specific Publication, by inserting an `Item` element inside a `Publication` element. These elements are formatted the same way as the `Item` elements directly under the `ItemTypes` element.

Replace the strings as follows:

STORAGE_ID

The ID of one of the storage mediums you specified in the `<Storages>` element.

CACHE_BEHAVIOR

Either `true` if you want the items to be cached, or `false` if you do not.

TYPE_NAME

The name of an item type you wish to configure storage for. `TYPE_NAME` must have one of the following values:

Value	Description
`Binary`	A binary resource
`ComponentLinkClick`	Information about visitors clicking on a Component Link, used in the Communication Statistics product
`ComponentPresentation`	A Component Presentation (the result of rendering a Component with a Component Template)

Value	Description
`ComponentVisit`	Information about visitors visiting a Component, used in the Communication Statistics product
`DynamicLinkInfo`	The properties of a dynamic link
`ExtensionData`	Rendering instructions related to custom content you add during publishing (if `TYPE_NAME` has this value, `STORAGE_ID` must refer to a database, not a file system)
`Metadata`	The metadata of items, including binary variants and link information
`Page`	A Page
`Personalization`	Data related to Personalization & Profiling functionality
`Publication`	A Publication
`Query`	A Query used for Intelligent Navigation
`Taxonomy`	A Taxonomy
`Timeframe`	A Timeframe, as used in Personalization & Profiling
`TrackedPage`	A Tracked Page, as used in Personalization & Profiling
`XSLT`	An Dynamic Template containing an XSLT stylesheet

You now override the default storage and caching behavior configured in the `ItemTypes` behavior, as specified.

6. Change the transaction timeout value by adding a `<Transaction>` element directly after the `Storages` element and before the `</Global>` closing tag. The following example sets the timeout value to 240000 ms (4 minutes):

```
...
</Storages>
  <Transaction Timeout="240000" MonitorInterval="5000" />
</Global>
```

The Storage Layer has a default timeout value of 2 minutes for transactions to the storage system. If you intend to store larger data sets, such as a large taxonomy containing 150,000 Keywords or more, this default timeout value is insufficient (4 minutes would be a better value in this example.)

Note: The `MonitorInterval` specifies how often to check for pending transactions. The default `5000` is usually a good value.

7. Save and close the `cd_storage_conf.xml` file.

8. Assuming Content Broker is running as a Web application, restart the Web application according to the instructions of your Web application server. In IIS, for example, perform an `iisreset`.

Next steps

Performing this configuration may involve entering sensitive data such as passwords in the configuration file. For that reason, you may want to encrypt these strings.

Configuring storage for the Experience Manager Web service

You need to edit the Storage Layer configuration file used by the Content Delivery Experience Manager Web service to configure Experience Manager to store session data in a dedicated database. If you have two or more staging environments, you can also configure load balancing.

Steps to execute

1. In the Web application that contains the Experience Manager Web service, open `cd_storage_conf.xml` (the Storage Layer configuration file) in a plain-text or XML editor.

2. If the Web application contains only the Experience Manager Web service, remove all references to other databases than the Content Data Store, such as the Outbound E-mail Subscription and Tracking databases, the Archive Manager database, and so on.

3. In the `<Storages>` element, add the following `<StorageBindings>`:

   ```
   <StorageBindings>
     <Bundle src="preview_dao_bundle.xml"/>
   </StorageBindings>
   ```

   Note: `preview_dao_bundle.xml` is bundled in the `cd_session.jar` file.

4. Using the PowerShell script, create a new, empty Content Delivery database.

5. Inside the `<Storages>` section, add a `<Wrappers>` section, as follows:

   ```
   <Wrappers>
     <Wrapper Name="SessionWrapper">
       <Timeout>120000</Timeout>
       <Storage Type="persistence" Id="db-session-webservice" dialect="MSSQL"
                Class="com.tridion.storage.persistence.JPADAOFactory">
         <Pool Type="jdbc" Size="5" MonitorInterval="60"
               IdleTimeout="120" CheckoutTimeout="120" />
         <DataSource Class="com.microsoft.sqlserver.jdbc.SQLServerDataSource">
           <Property Name="serverName" Value="myserver" />
           <Property Name="portNumber" Value="1433" />
           <Property Name="databaseName" Value="mydb" />
           <Property Name="user" Value="admin" />
           <Property Name="password" Value="kumquat" />
         </DataSource>
       </Storage>
     </Wrapper>
   </Wrappers>
   ```

   The `Storage` element in this fragment must refer to the database you just created. Ensure that the storage identifier (here `db-session-webservice`) is different from the storage identifier you configure in the storage configuration of the staging Web site.

6. To configure load balancing, do the following:

a. In the <Wrapper> element, add a <MachineName> element specifying a logical name:

```
<Wrappers>
  <Wrapper Name="SessionWrapper">
     <MachineName>MachineName</MachineName>
     <Storage>
       ...
     </Storage>
  </Wrapper>
</Wrappers>
```

The maximum length of the <MachineName> element is 100 characters and the value needs to be unique for each Web application in the load balanced environment.

b. Configure the same database in the cd_storage_conf.xml for all Web applications in the load balanced environment.

7. Save and close cd_storage_conf.xml.

8. In your Web application server, ensure that your Java Virtual Machine (JVM) is called with the correct file encoding by setting the file.encoding property. You can set this property in a jvm.xml file of your Web applications. On a Windows machine, you can also choose to set this property system-wide in your registry. Refer to your Web application server documentation for more information.

9. If you store content in a database, and your Content Delivery session preview Web service Web application does not already contain a license file, ensure the presence of a Content Delivery license file, typically called cd_licenses.xml, in the WEB-INF/classes/ (Java) or the bin\config\ (.NET) subfolder of the Web application.

Configuring storage for the Experience Manager Web site extension

You need to edit the Storage Layer configuration file used by the staging Web site to configure the Experience Manager Web site extension to store session data in the dedicated database you created earlier. If you have two or more staging environments, you can also configure load balancing, but that does mean that you need to use a dedicated database.

Steps to execute

1. Open cd_storage_conf.xml (the Storage Layer configuration file) on your staging Web site in a plain-text or XML editor.

2. In the <Storages> element, add the following <StorageBindings>:

```
<StorageBindings>
  <Bundle src="preview_dao_bundle.xml"/>
</StorageBindings>
```

Note: preview_dao_bundle.xml is bundled in the cd_session.jar file.

3. Inside the <Storages> section, add a <Wrappers> section, as follows:

```
<Wrappers>
  <Wrapper Name="SessionWrapper">
    <Timeout>120000</Timeout>
    <Storage Type="persistence" Id="db-session-staging" dialect="MSSQL"
```

```
             Class="com.tridion.storage.persistence.JPADAOFactory">
      <Pool Type="jdbc" Size="5" MonitorInterval="60"
            IdleTimeout="120" CheckoutTimeout="120" />
      <DataSource Class="com.microsoft.sqlserver.jdbc.SQLServerDataSource">
        <Property Name="serverName" Value="myserver" />
        <Property Name="portNumber" Value="1433" />
        <Property Name="databaseName" Value="mydb" />
        <Property Name="user" Value="admin" />
        <Property Name="password" Value="kumquat" />
      </DataSource>
    </Storage>
  </Wrapper>
</Wrappers>
```

The `Storage` element in this fragment must refer to the database you created when you set up the Experience Manager Web service Server Role. Ensure that the storage identifier (here `db-session-staging`) is different from the storage identifier you configure in the storage configuration of the Experience Manager Web service.

4. To configure load balancing, do the following:

   a. In the `<Wrapper>` element, add a `<MachineName>` element specifying a logical name:

   ```
   <Wrappers>
     <Wrapper Name="SessionWrapper">
        <MachineName>MachineName</MachineName>
        <Storage>
          ...
        </Storage>
     </Wrapper>
   </Wrappers>
   ```

   The maximum length of the `<MachineName>` element is 100 characters and the value needs to be unique for each Web application in the load balanced environment.

   b. Configure the same database in the `cd_storage_conf.xml` for all Web applications in the load balanced environment.

5. Save and close `cd_storage_conf.xml`.

6. In your Web application server, ensure that your Java Virtual Machine (JVM) is called with the correct file encoding by setting the `file.encoding` property. You can set this property in a `jvm.xml` file of your Web applications. On a Windows machine, you can also choose to set this property system-wide in your registry. Refer to your Web application server documentation for more information.

7. If you store content in a database, and your Content Delivery staging Web site does not already contain a license file, ensure the presence of a Content Delivery license file, typically called `cd_licenses.xml`, in the `WEB-INF/classes/` (Java) or the `bin\config\` (.NET) subfolder of the Web application.

8.9.2 Encrypting sensitive strings

SDL Tridion provides a command line tool for encrypting sensitive data such as passwords in your configuration files as you may not want to include sensitive data in the configuration files in cleartext.

Steps to execute

1. On the Content Delivery server, open a command prompt.

2. Navigate to the directory that contains the file `cd_core.jar`. By default, this file is located in your Content Delivery root location.

3. Enter the following command, where `mysensitivestring` is the string you want to encrypt:

   ```
   java -cp cd_core.jar com.tridion.crypto.Encrypt mysensitivestring
   ```

   (You may wish to redirect output to a file for easy copy-pasting of the tool's response).

   The encryption responds as follows:

   ```
   SDL Tridion configuration value = encrypted:9FUJ9CP81Oj63VhnJxcqx//
   pW3fP4bekeupIexctzcs=
   ```

4. In your configuration file where you would normally insert your sensitive data, now insert the string returned by the tool, including the `encrypted:` prefix. For example:

   ```
   <Property Name="password" Value="encrypted:fhS6whp/g==hjS2G2rG6aI6T"/>
   ```

Result

When Content Delivery loads the configuration file, it decrypts the encryption string back to its original value.

8.10 Configuring the Experience Manager Web site extension

This section explains how to configure the Experience Manager Web site extension.

8.10.1 Preventing 404 errors on your staging Web site

Place a blank HTML file in the root of the staging Web site to prevent 404 messages on your staging Web site while editing.

Steps to execute

1. Access the server on which your staging Web site runs.

2. Navigate to the root folder of your staging Web site.

3. Create a plain-text file. This file can be completely empty, or it can be a valid HTML page without actual content.

4. Save this file as `se_blank.html` on the root folder, and close it.

8.10.2 Updating frame references on your staging Web site

If your staging Web site contains any link that points to the `_top` frame (either through a `target` attribute set to the value `_top`, or through JavaScript), such a link will not work correctly while a user is editing the Web page. To fix this, change such target reference in one of several ways.

Steps to execute

1. If the target frame of the link is a direct parent of the source frame, explicitly point to the parent frame instead. For plain HTML links, this means setting the `target` attribute to the value `_parent`; for JavaScript, this means using `window.parent`.

2. Alternatively, in a more complex scenario, if you use JavaScript and you can find out the distance from the source frame to the target frame, use an expression such as `window.parent.parent.parent`, where the number of `parent` items corresponds to the distance to the target frame.

3. Alternatively, if you use HTML links, or if the frame structure is more complex or dynamic, give the target frame of the link a name using the following JavaScript assignment:

   ```
   window.name="my_named_topframe"
   ```

   where `my_name_topframe` is the name of the top frame on your Web page.

   You can now refer to the frame by name from plain HTML using `target="my_named_topframe"`, or from JavaScript using `top.frames["my_named_topframe"]`.

8.11 Configuring the Java Virtual Machine

If you use .NET as your primary Content Delivery technology, you can configure the JVM for all .NET services and .NET Web application. Regardless of your technology, you can also run the JVM in server mode.

8.11.1 Configuring the JVM for Windows services and .NET Web applications

Content Delivery Server Roles running as Windows services or .NET Web applications start a Java Virtual Machine (JVM). You can specify switches for these JVM calls. (To configure JVM calls from Java Web applications, refer to the documentation of your Java/JSP Web application server.)

Context

The following Server Roles (running as Windows services or .NET Web applications) call a JVM:

On the Content Manager server:

- Tridion Content Distributor Transport Service
- Tridion Monitoring Service

On a Content Delivery server that uses IIS and/or .NET:

- Tridion Cache Channel Service
- Tridion Content Deployer
- Any Server Role running as a .NET Web application
- Tridion Monitoring Service

You configure the Java home directory and JRE version for all of these Server Roles in the registry.

In addition, each of these Server Roles has its own JVM. On a Content Manager Server, you can configure the JVM call switches for the Transport Service running on that machine (note that this also affects all other .NET Web applications and Windows services that call a JVM on that machine).

On a Content Delivery server, you can do the same as on the Content Manager server, or you can configure separate JVM settings for each Windows service or .NET Web application.

Steps to execute

1. To configure the Java home directory or the JRE version, do the following:

    a. Start a registry editor, for example, `regedit`.

    b. Depending on whether you use a 32-bit or a 64-bit operating system, check for the existence of a registry subkey `Tridion` in one of the following locations:

    - On a 32-bit system, search for this subkey in `HKEY_LOCAL_MACHINE\SOFTWARE`
    - On a 64-bit system, search for this subkey in `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node`

    If you cannot find this key, create it.

    c. In the `HKEY_LOCAL_MACHINE\SOFTWARE\Tridion` subkey, create a subkey called `Content Delivery`.

    d. In the `HKEY_LOCAL_MACHINE\SOFTWARE\Tridion\Content Delivery` subkey, create a subkey called `General`.

2. You can now add one or both of the following entries to this subkey:

    `TRIDION_HOME` (the SDL Tridion home directory)

    Each Windows service needs to know where to find certain SDL Tridion files to include in its classpath. See for details.

If you specify a value for `TRIDION_HOME`, the Windows services will use that value as their starting point for finding files, rather than the value of the `TRIDION_HOME` environment variable. This is useful if you have Content Manager and Content Delivery installed on the same machine, and want to place the Content Delivery-related files in a different directory than the Content Manager.

`JREVersion` (the version of the Java Runtime Engine)

Setting the `JREVersion` is relevant if you have multiple versions of the Java Runtime Environment (JRE) installed.

By default, Content Delivery sets the JRE version to whatever it finds as the value of the following registry subkey:

```
HKEY_LOCAL_MACHINE\SOFTWARE\JavaSoft\Java Runtime
Environment\CurrentVersion
```

If you specify a registry value for the JRE for one or all services, then that version will be used instead of the default one. The version must, of course, be installed on the server.

To see which versions of the JRE are installed, look at the subkeys found in the following registry subkey:

```
HKEY_LOCAL_MACHINE\SOFTWARE\JavaSoft\Java Runtime
Environment\
```

Set the `JREVersion` entry to the name of one of the subkeys (that is, a valid version number such as "1.6" or "1.7"). If you fill in a version that is not installed, you cannot start the affected Windows services or .NET Web applications after you stop them.

3. To configure the same JVM settings for all Windows services and .NET Web applications (whether Content Delivery-related or not) on the machine, create the key structure described above (if you have not done so already) and add entries to the `HKEY_LOCAL_MACHINE\SOFTWARE\Tridion\Content Delivery\General` subkey called `jvmarg1`, `jvmarg2`, `jvmarg3` and so on, to represent JVM startup switches.

   By default, each Windows service or .NET Web application calls the JVM with its own default parameters. To add one parameter to your .NET services, create a registry entry `jvmarg1` in the `General` subkey. To add two parameters, create two registry entries, `jvmarg1` and `jvmarg2`, and so on. Give each registry entry the value of a JVM parameter, including an initial dash (-). For example, to set the maximum heap size to 256 MB, add a `jvmarg1` registry entry and give it the value `-Xmx256M`.

   To extract startup log files from the .NET services and write them to a file in a folder location, specify a `jvmargX` entry (where X is a number) entry and set it to the value `log=<fullpath>`, where `<fullpath>` is the full path to a folder location. For example, a valid value could be `log=D:\logfiles\services`.

   Note the following:

   - If you supply a `jvmargX` registry entry that sets the classpath (that is, with a value that starts with `-Djava.class.path` or with `-cp`), the service will not work, because your classpath will override the default one set by the Windows service or .NET Web application itself.
   - You can see in Windows Event Viewer which startup options are applied, in the Tridion Content Manager category.

4. Alternatively, if you are on a Content Delivery server, you can configure JVM startup settings for a specific Windows service or .NET Web application (except for the Tridion Content Distributor Transport Service or the Tridion Monitoring Service). To do this, ensure the presence of a file called `jvm.xml` in the `config\` subfolder of the Windows service or in the `bin\config\` subfolder of the .NET Web application. You can copy the sample file `jvm_sample.xml` found in the folder `Content Delivery\resources\configurations\` on the installation media and rename it to `jvm.xml`. The file has the following structure:

   ```
   <VirtualMachine>
     <Options>
       <Option Name="-NAME" Value="VALUE"/>
     </Options>
   </VirtualMachine>
   ```

   where *NAME* is the JVM switch you wish to set, and *VALUE* is the value you want to specify for that switch. You can specify as many `Option` elements as you like, then save and close the file.

   You have configured settings for one Windows service or .NET Web application, which override any corresponding settings in the registry. You can repeat this step to configure JVM calls from other Windows services or .NET Web applications.

5. If you used the registry editor, close it now.

6. Stop and start all Content Delivery .NET services and Web applications affected by your changes, and restart IIS.

Result

When you now start a Content Delivery .NET service or Web application, Content Delivery applies the settings you specified.

8.11.2 Running the JVM in server mode

The Sun Java Virtual Machine (JVM) used by Content Delivery services can run in Client or Server mode. You can run Java applications using the server virtual machine by modifying the `jvm.cfg` file

Context

By default, Java applications and applets run in Client Mode as the service will initially run faster. However, for long-term use it is recommended to run the JVM in Server Mode. Running the JVM in server mode will improve the performance of Windows Services, such as the Content Deployer and the Cache Channel Service.

Steps to execute

1. Open `jvm.cfg` for editing, which is in one of the following subdirectories of your Java home directory (typically found as the value of the `JAVA_HOME` environment variable on your machine):
   - On a 64-bit installation: `lib\amd64\`
   - On a 32-bit installation: `lib\i386\`

2. Change the following lines:

```
-client KNOWN
-server KNOWN
```

to:

```
-server KNOWN
-client KNOWN
```

3. Save and close `jvm.cfg`.

Result

As a result, the server Virtual Machine will be run for all applications with the exception of applications that are run with the `-client` argument.

Note that you can also configure these options in the registry, see Configuring the JVM for Windows services and .NET Web applications (see page 118).

Chapter 9 Setting up Publishing (Content Distribution)

To be able to publish content from the Content Manager server to the Content Delivery server, on the Content Manager server you need to create and configure Publication Targets, Protocols, Target Types, and the Transport Service. (To enable Content Distribution on the Content Delivery Server you need to configure (or customize) your Content Deployer and Storage Layer (and optionally Caching).

Note: You should set up Publishing after you have installed a Content Delivery server.

9.1 Setting up Target Types

A Target Type specifies a user-friendly name for one or more Publication Targets and specifies permission settings for the target(s).

Context

When Content Manager users publish content, they select one or more Target Types to publish to. The publish action then publishes content to those Publication Targets that the user has permission to publish to.

For example, your organization may have two different Target Types:

- A Target Type called **Staging** to which authors can publish content.
- A Target Type called **Live** to which only editors can publish content.

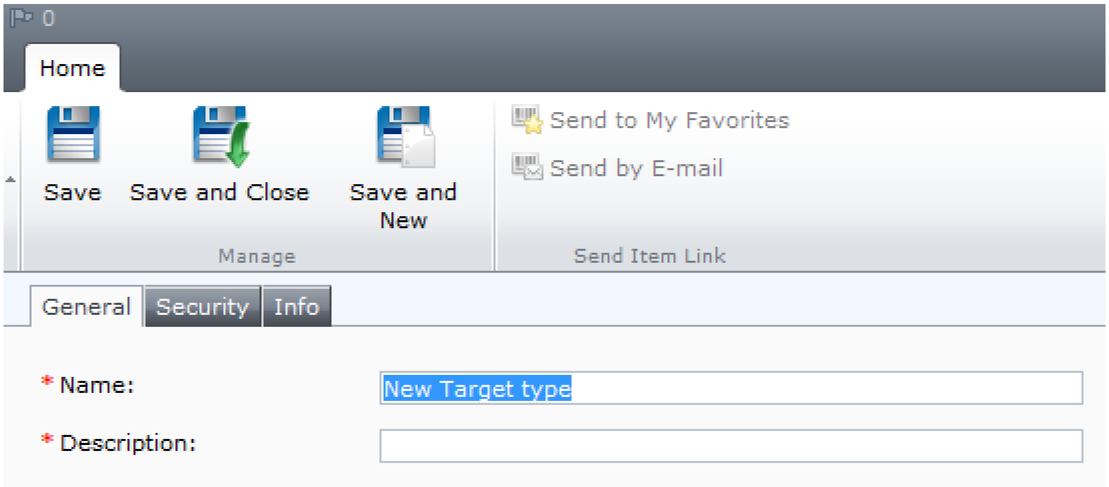
In this example, the Target Types represent types of Publication Targets. The Staging Target Type might publish to a local file system while the Live Target Type may publish to an HTTP server that external Web site visitors can access.

Steps to execute

1. Access the Content Manager Explorer Web site as an administrator user and select **Administration** at bottom left. In the **Administration** tree that appears on the left, navigate to **Publication Management** > **Target Types** .

2. Now do one of the following:

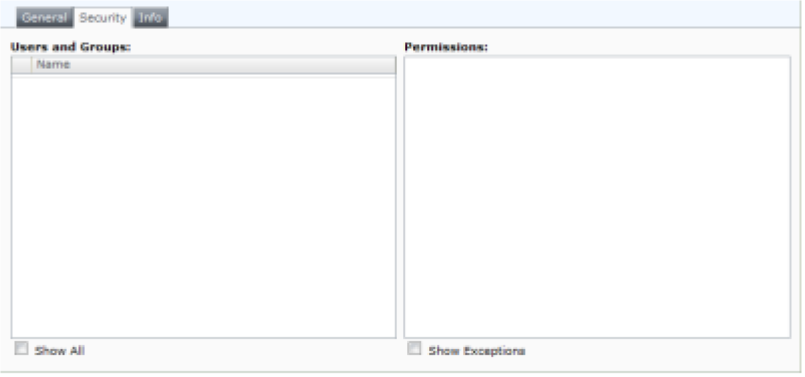
   - In the **Administration** tab of the Ribbon toolbar, click **Target Types** and select **New Target Type** from the dropdown menu.
   - Right-click an empty location in the content area and select **New** > **Target Type** from the context menu that opens.



* Name: New Target type

* Description:

3. Fill in a **Name** and a **Description** for this Target Type.
4. Select the **Security** tab.



5. Select **Show all** to see a list of all configured Content Manager Users and User Groups (if you have not configured any users, the list contains only the Content Manager Administrator user). To allow a User or User Group to use a Target Type, select that User or User Group and select Use Target Type.
6. Click **Save and Close** to create the Target Type. The New Target Type window closes, and the new Target Type appears in the list in the main content area.

Result

You have now create a Target Type that you can associate with one or more Publication Targets.

9.2 Setting up Publication Targets

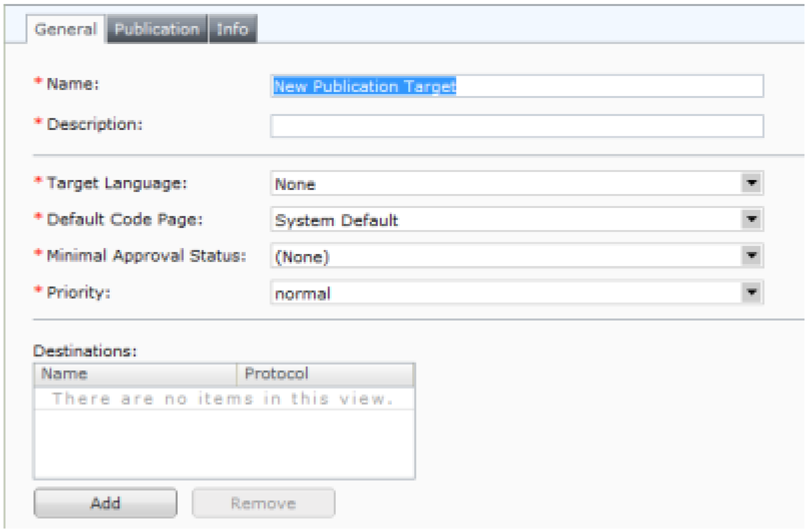
A Publication Target defines the information required to publish content. The Transport Service (the publishing engine) uses this information when it creates its Transport Package to send to the destination.

Steps to execute

1. Access the Content Manager Explorer Web site as an administrator user and select **Administration** at bottom left. In the **Administration** tree that appears on the left, navigate to **Publication Management** > **Publication Targets** .

2. Now do one of the following:

   - In the **Administration** tab of the Ribbon toolbar, click **Publication Targets** and select **New Publication Target** from the dropdown menu.
   - Right-click an empty location in the content area and select **New** > **Publication Target** from the context menu that opens.



* Name: New Publication Target

* Description:

* Target Language: None

* Default Code Page: System Default

* Minimal Approval Status: (None)

* Priority: normal

Destinations:

Name	Protocol
There are no items in this view.	

3. On the **General** tab, fill in the following fields:

   Name

   A unique name for this Publication Target

   Description

   A description of this Publication Target.

   Leave the other settings set to their default values.

4. In the **Destinations** area, click **Add** to add a Content Delivery server. In the area on the right, specify the following:

   Destination Name

   A unique name by which you can identify this Content Delivery server.

   Protocol

   A transport protocol to use for publishing content to this Content Delivery server. Selecting a transport protocol reveals the parameters for that protocol. Fill in the transport protocol parameters according to the following tables:

FTP protocol parameter	Description
Host	The IP address or host name of the FTP server
Port	The port number of the FTP server
UserName	A user name to authenticate against the FTP server
Password	A password to authenticate against the FTP server
Location	The path to which the FTP Sender copies its Transport Package. This path is relative to the FTP root and should include forward slashes '/' at its start and end. For example, to send content to `c:\inetpub\ftproot\dest\`, fill in `/dest/`. Note: The location you specify here must correspond to the value of the `Path` attribute of a `Location` element in the Content Deployer configuration file.
Connection mode	The way in which the Content Manager server connects to the FTP server (`Active` or `Passive`). If left blank, defaults to `Passive`.

HTTP or HTTPS protocol parameter	Description
Login	The user name used to access an HTTPS connection. Leave empty for HTTP.
Password	The password used to access an HTTPS connection. Leave empty for HTTP.
Fully specified URL to upload page	The Web address of the HTTP(S) server and its file upload location. Start this address with http:// for HTTP transport, with https:// for HTTPS transport.

Leave the other controls empty.

Local File System protocol parameter	Description
Location	A file system location to which the content will be published.

Local File System protocol parameter	Description
	Note: The location you specify here must correspond to the value of the `Path` attribute of a `Location` element in the Content Deployer configuration file.

SFTP protocol parameter (FTP over SSH):	Description
Hostname for the FTP server	The IP address or host name of the FTP server
FTP Port to use	The port number of the FTP server
FTPUserName	A user name to authenticate against the FTP server
Password for the FTP server	A password to authenticate against the FTP server
Path on the FTP server	The path to which the FTP Sender copies its Transport Package. This path is relative to the FTP root and should include forward slashes '/' at its start and end. For example, to send content to `c:\inetpub\ftproot\dest\`, fill in `/dest/`. Note: The location you specify here must correspond to the value of the `Path` attribute of a `Location` element in the Content Deployer configuration file.
Hostname for the SSH server	The IP address or host name of the SSH server
SSH Port to use	The port number of the SSH server
SSHUserName	A user name to authenticate against the SSH server
Password for the SSH server	A password to authenticate against the SSH server
Connection mode	The way in which the Content Manager server connects to the FTP server (Active or Passive). If left blank, defaults to Passive.

SSHFTP protocol parameter (SSH2 using an SFTP subsystem)	Description
Hostname	The IP address or host name of the SSH server
SSH Port	(optional) The port number of the SSH server (defaults to 22)
Location	The path to which the SSHFTP Sender copies its Transport Package. This path is relative to the SSH server root and should include forward slashes '/' at its start and end. For example, to send content to `c:\inetpub\sshroot\dest\`, fill in `/dest/`. Note: The location you specify here must correspond to the value of the `Path` attribute of a `Location` element in the Content Deployer configuration file.
UserName	A user name to authenticate against the SSH server
Password	(optional) A password to authenticate against the SSH server. Not needed if you use a private key.
Private key file	(optional) The full path and filename of a file containing the private key used to access the SSH server. This file must be located on the Content Manager machine. Not needed if you use a password.

5. Click **Add** again to add more Content Delivery servers.

6. Select the **Publication** tab and do the following:

   - To specify the Publications from which you want to allow content to be published to this Publication Target, add Publications from the **Available Publications** to the **Allowed Publications**.
   - To make this Publication Target part of a Target Type, add the Target Type from the **Available Target Types** to the **Allowed Target Types**.

   To move an item from the list on the left to the list on the right, double-click the item on the left or select it and click **Add**. For the reverse, double-click the item on the right or select it and click **Remove**.

7. Click **Save and Close** to create the Publication Target.

Result

The New Publication Target window closes and the new Publication Target appears in the list in the main content area.

9.3 Configuring the Transport Service

Configure the Transport Service to make actual publishing of content over one or more transport protocols possible. To configure the Transport Service protocol:

Steps to execute

1. Open `cd_transport_conf.xml`, by default located in the Content Manager `config` directory, in a plain-text or XML editor.

2. In the `WorkFolder` element, set the value of the `Location` attribute to a folder where the Transport Service can store temporary files.

3. If you want to configure multiple thread pools for the Transport Service, you can do so by insert an element called `Workers` directly under the `WorkFolder` element, and setting one or more of the following attributes:

   `TransportPriorityPoolSize`

   The maximum number of workers for normal transport. Defaults to 5 if omitted. This attribute can also be set to 0 (meaning unlimited threads).

   `NormalPriorityPoolSize`

   The maximum number of workers for transport operations such as unzipping and state synchronization. Defaults to 5 if omitted.

   `HighPriorityPoolSize`

   The maximum number of workers for high-priority transport. Defaults to 5 if omitted.

   Each of the attribute values must be equal to, or higher than, the number of threads configured for the Publisher in the MMC Snap-in.

   Note: If you store metadata on the file system rather than in a database, you cannot configure multiple workers.

4. Find the Senders and add the following Sender element(s) as needed:

   Local File System

```
<Sender Type="Local" Class="com.tridion.transport.connection.connectors.FileTransportConnector" />
```

   FTP

```
<Sender Type="FTP" Class="com.tridion.transport.connection.connectors.FTPTransportConnector" />
```

   SFTP

```
<Sender Type="FTP" Class="com.tridion.transport.connection.connectors.FTPTransportConnector" />
```

SSHFTP

```
<Sender Type="SSHFTP" Class="com.tridion.transport.connection.connectors.SSHFTPTransportConnector" />
```

HTTP or HTTPS

```
<Sender Type="HTTPS" Class="com.tridion.transport.connection.connectors.HTTPSTransportConnector" />
```

5. Save and close `cd_transport_conf.xml`.

6. Access your Windows **Services**.

7. Locate the Windows service called **Tridion Transport Service** and restart it.

9.4 Configuring Content Delivery logging on the Content Manager server

The `logback.xml` file is the configuration file for the logback logging framework used by Content Delivery.

Context

Note: For detailed information about the Logback framework and its configuration, refer to http://logback.qos.ch/documentation.html.

Steps to execute

1. Open the file `logback.xml` in a plain-text editor or XML editor:

   - On Content Manager, the `logback.xml` file is located in the Content Manager root location (defaults to `C:\Program Files (x86)\Tridion\`).

2. Set the following properties:

```
<property name="log.pattern" value="%date %-5level %logger{0} - %message%n"/>
<property name="log.history" value="7"/>
<property name="log.folder" value="c:/tridion/log"/>
<property name="log.level" value="WARN"/>
```

`log.pattern` (logging filename format)

The `log.pattern` property specifies the name of the log files using a format string. Refer to the online Logback documentation for more information.

`log.history` (number of log files)

The `log.history` property specifies the number of log files to back up (the default is 7 files).

`log.folder` (logging location)

The `log.folder` property specifies the location where log files are saved (defaults to `c:/Tridion/log`).

For example, on the Content Manager Server set the log location to: `c:/Program Files (x86)/Tridion/log`.

`log.level` (logging level)

To change the default logging level, change the value of the `log.level` property to one of the values describes in the following table. Note that logging levels are incremental and inclusive; each level logs data for the previous levels. For example, `DEBUG` also logs and events covered by `INFO`, `WARN`, and `ERROR`.

Logging Level	Description
`OFF`	Turn off logging.
`ERROR` (default)	Log events that cause a specific feature of the application to fail.
`WARN`	Log events that are potentially harmful.
`INFO`	Log neutral, informational messages about high-level functionality being executed.
`DEBUG`	Log fine-grained informational messages about low-level functionality being executed, intended to discover the source of a problem.
`TRACE`	Log highly detailed informational messages about the application's lowest-level activity,
`ALL`	Maximum logging level that logs everything.

Note: The `DEBUG` level is intended for hunting down the origin of a particular problem only and will result in a significant performance drop while this level is set. Therefore, it is recommended to set the logging level back to as soon as the problem has been resolved.

3. If you want to use property substitutions, that is, variables resolved by context such as environment variables, the Logback framework allows you to do so. The simplest way to do this is to define an environment variable (say `Foo`) wherever you set up Content Delivery, and insert the string `${Foo}` in your logging configuration file. Logback now substitutes the string with the value of the environment variable. This approach enables you to reuse the same configuration file (and, say, deploy a WAR file without having to edit this file). For more information, refer to the heading "Property substitution" on this Web page: http://logback.qos.ch/manual/configuration.html.

4. Save and close `logback.xml`.

9.5 Setting up publishing for Experience Manager

To enable publishing content to a staging site for your users, show and fill in the **Session Preview** tab in a Publication Target.

Steps to execute

1. Access Content Manager Explorer.

2. If you have not already done so, create a Publication Target that publishes to your staging Web site.

3. In the **General** tab of your Publication Target, select **Enable for inline editing**.

   A new **Session Preview** tab appears.

4. On the new **Session Preview** tab, set **Content Delivery Endpoint URL** to the URL of the Experience Manager Web service end point. This URL is one of the following:

   - For a JSP Web site, add the string `<baseURL>/cd_preview_webservice/odata.svc/`. If you have a `servlet-mapping` defined in your `web.xml`, then insert the path of the servlet mapping after `cd_preview_webservice/`. For example, if your servlet mapping maps the Web service to a subdirectory `ws/`, add the string `<baseURL>/cd_preview_webservice/ws/odata.svc/`.
   - For a .NET Web site, add the string `<baseURL>/odata.svc/`.

   where *<baseURL>* is the base URL (host name and port) of the machine on which you set up your Content Delivery session preview Web service, for example, `http://localhost:8080/`.

5. If you made your Experience Manager Web service secure, also set the following properties:

   OData Access Token URL

   The URL for obtaining an access token from the Experience Manager Web service. This value is one of the following:

   - For a JSP Web site, the URL is `<baseURL>/cd_preview_webservice/access_token/`
   - For a .NET Web site, the URL is `<baseURL>/access_token.svc/`

   where *<baseURL>* is the same base URL you used before.

   User Name

   The name of a user authorized to access the Web service. You configured this user name when you secured the Web service.

   Password

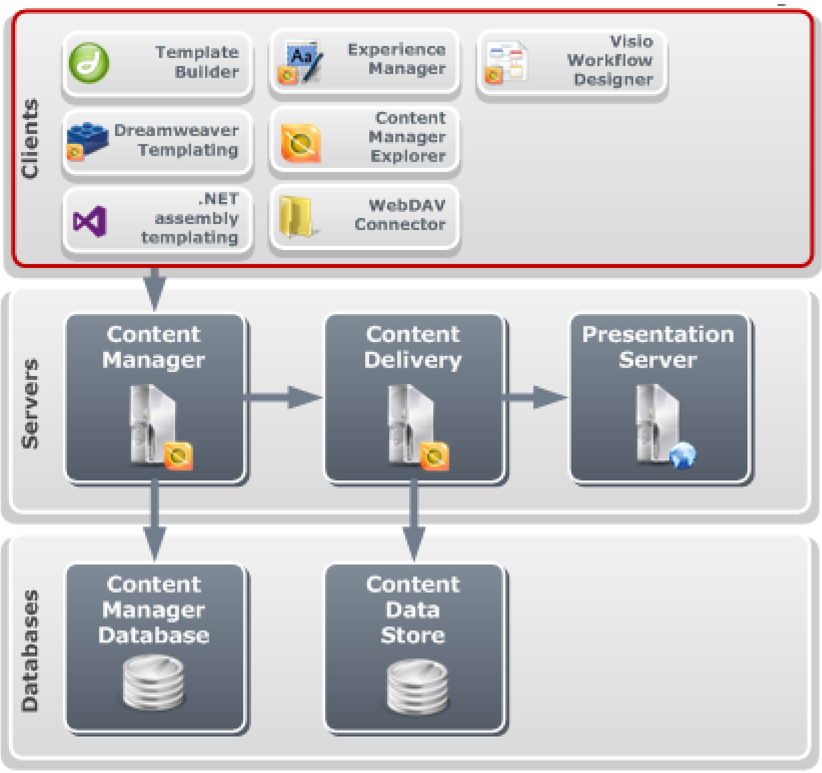
   The password that this user uses to authenticate against the Web service. You configured this password when you secured the Web service.

6. Use the **Add** button under **Web site URLs** to create a list of one or more URLs that represent the staging Web site, such as `http://localhost:82` or `http://myserver.mydomain:83/`.

Chapter 10 Installing core Content Manager clients

For a basic SDL Tridion installation, install the Content Manager Explorer to manage your content; Template Builder, .NET assembly templating, and Dreamweaver templating to develop templates; and the Experience Manager to edit content.

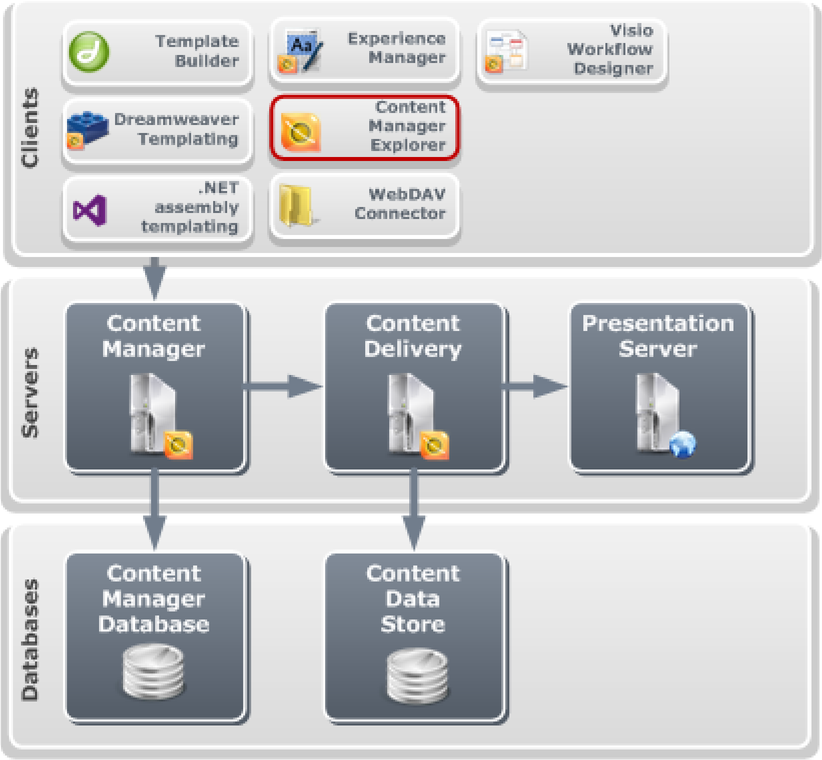
Context



10.1 Setting up the Content Manager Explorer client

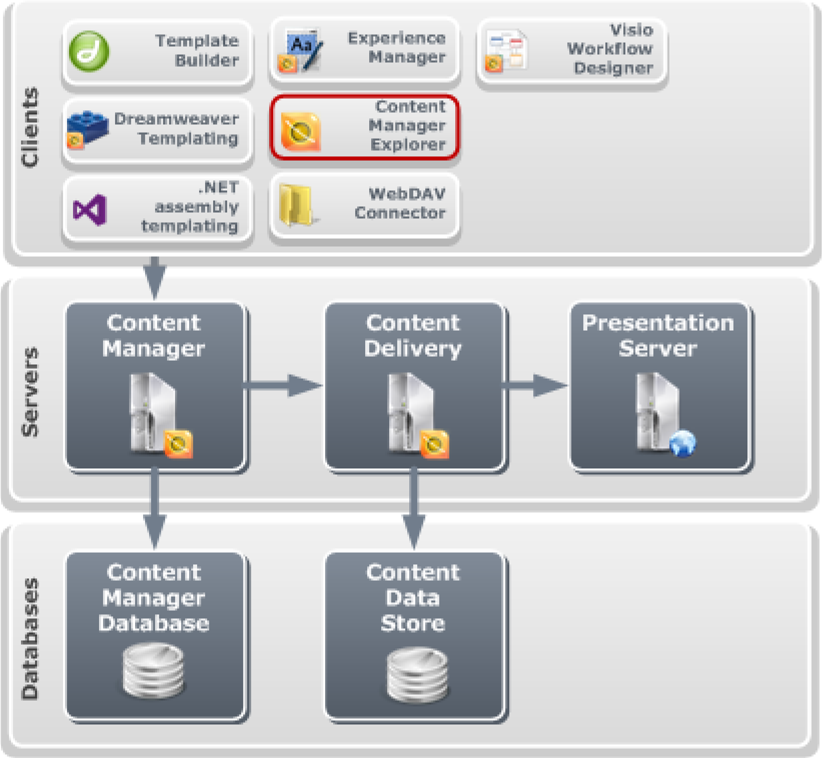
Content Manager Explorer is a Web-based Graphical User Interface to the Content Manager Server, used for managing your content. To get Content Manager Explorer up and running on a client machine, check the prerequisites, configure the browser, and configure Content Manager Explorer.

The Content Manager Explorer interface is available at `http://<rootlocation>/WebUI`, where `<rootlocation>` is the Web site you configured for the Content Manager Explorer.



10.1.1 Content Manager Explorer prerequisites

This topic lists the required and supported software for the Content Manager Explorer browser-based interface. The prerequisites are the same as for Experience Manager .



Recommended prerequisites

SDL recommends the following set of Content Manager Explorer prerequisites:

Prerequisite	Recommended product release
Operating system	Microsoft Windows 8 (x86 or x64), or Apple Mac OS X 10.8 Mountain Lion
Internet browser	Microsoft Internet Explorer 10.0, or Safari 6.0 for the Mac (if using Mac OS X)

Operating system languages

The operating system on which Content Manager Explorer runs must be in one of the following languages:

- US English
- French
- German
- Spanish
- Dutch
- Japanese

Operating system versions and editions

Supported:

OS Version	Service pack	Architecture
Windows 8	n/a	x86
Windows 8	n/a	x64
Windows 7	SP1	x86
Windows 7	SP1	x64
Windows Server 2012	n/a	x64
Windows Server 2008 R2	SP1	x64
Apple Mac OS X 10.8 Mountain Lion	n/a	n/a

Deprecated: None

Note: The SDL Tridion installation media contains a Web page, `\Redistributables\Readme.htm`, with links to the Microsoft download pages for the Service Packs listed in the table.

Web browser

Supported:

- Internet Explorer 10.0 (on Windows 8 and Windows Server 2012)
- Internet Explorer 9.0 (on other Windows versions)
- Safari 6.0 for the Mac
- Google Chrome, latest version
- Mozilla Firefox, latest version

Deprecated: None

For information about the supported operating systems for each browser, contact the vendor of the browser you intend to use.

Updating supported browser versions

The list of supported Web browser versions was known and tested at the time of release. However, new Web browser versions may have been released since that time that are also supported but have not been subjected to testing. As a result, the Content Manager Explorer may display a 'browser version not supported' warning contrary to this list, or it may display no warning even though the browser version you use is not listed.

If you have not yet already done so for the user interface for context editing, change SDL Tridion's response to a specific browser version by configuring the `system.config` file:

1. Access the subfolder `WebUI\WebRoot\Configuration` of the Content Manager root location (defaults to `C:\Program Files (x86)\Tridion\`).

2. Open `system.config` in a plain-text or XML editor. The `<supportedbrowsers>` section near the top of the file contains `browser` elements for supported browser versions.

3. To stop SDL Tridion from issuing a warning for an unsupported browser version, copy and paste an existing `browser` element for an earlier version and set the set the `name` and `regExp` attributes to values that represent the supported version.

4. To make SDL Tridion issue a warning for a browser version that is no longer listed as supported, remove the `browser` element that has its `name` attribute set to the browser version you want to issue a warning for.

5. Save and close `system.config`.

6. Restart IIS.

Licenses

To use Content Manager Explorer, you must have a license on the Content Manager server machine that explicitly licenses Content Manager Explorer.

10.1.2 Configuring your Web browser for Content Manager Explorer

This section explains how to configure your Web browser for use with Content Manager Explorer.

Context

Note: Depending on rights configured, not all users of the client system may be able to perform these tasks.

Steps to execute

1. To configure the browser, on your client machine, start Internet Explorer (even if you do not intend to use Internet Explorer as your browser).

2. In Internet Explorer, if you do not see the menu bar, press your left **Alt** key on the keyboard to display it.

3. Select **View** > **Encoding** > **Unicode** from the menu. This sets the browser encoding to Unicode.

4. Select **Tools** > **Internet Options** from the menu. The **Internet Options** dialog opens.

5. Access the **Security** tab, select the **Local intranet zone** and click **Sites**. The **Local intranet** dialog with generic site settings opens.

6. Select **Advanced** to open a **Local intranet** dialog listing sites associated with this zone.

7. Enter the URL of your Content Manager Explorer Web site and click **Add**. The site is added to the list. Click **Close** to close this dialog, and **OK** to close the generic sites dialog. The settings for the **Local intranet zone** reappear.

8. Click **Custom Level.** The **Security Settings - Local intranet Zone** dialog opens.

9. In the list of settings, find the following settings and set each to the value `Enabled`:

   - ActiveX:
     - **Run ActiveX controls and plug-ins**
     - **Script ActiveX controls marked safe for scripting**
   - Downloads:
     - **File downloads**
   - Miscellaneous:
     - **Submit non-encrypted form data**
   - Scripting:
     - **Active scripting**
     - **Allow Programmatic clipboard access**

   Click **OK** to apply these settings and close the dialog. The **Internet Options** dialog reappears, still showing the **Security** tab.

10. Select the **Privacy** tab and click **Advanced**. The **Advanced Privacy Settings** dialog opens.

11. Select **Override automatic cookie handling** and then select **Always allow session cookies**. Then click **OK** to close the **Advanced Privacy Settings** dialog. The **Internet Options** dialog reappears, still showing the **Privacy** tab.

12. Click **OK** to close the **Internet Options** dialog.

13. If you intend to use Mozilla Firefox as your browser, do the following:

    a. While logged in as an end user, launch Firefox and type `about:support` in the address bar.

    b. In the **Application Basics** area, check the **Profile Directory** property and click **Open Containing Folder** next to it.

       Windows Explorer (Windows) or Finder (Mac) opens, showing the Firefox profile folder.

    c. Close Firefox.

    d. In the folder, check for the presence of a file called `user.js`. If it exists, open it in a plain-text editor; if it does not exist, create it in a plain-text editor.

    e. Add the following lines anywhere in the `user.js` file:

```
user_pref("capability.policy.policynames", "allowclipboard");
user_pref("capability.policy.allowclipboard.sites", "http://<cmsname>");
user_pref("capability.policy.allowclipboard.Clipboard.cutcopy", "allAccess");
user_pref("capability.policy.allowclipboard.Clipboard.paste", "allAccess");
user_pref("browser.link.open_newwindow", 2);
```

       in which *<cmsname>* is the domain name of the Content Manager server machine.

Note: You can also add comments on separate lines. Start these lines with a double forward slash (`//`).

f. Save and close the `user.js` file.

g. If you are using Firefox on Mac OS X, also do the following:

- In the Apple menu, select **System Preferences**, then **Keyboard and Mouse**, then **Keyboard Shortcuts**
- Under **Full Keyboard Access**, select **All controls**.

Mozilla Firefox is now properly configured.

14. If you intend to use Google Chrome as your browser, SDL Tridion recommends to disable the auto-update functionality of Chrome.

    By default, Google Chrome is configured to auto-update to its latest version. Because this user interface cannot be guaranteed to work with a newer version of Google Chrome,disabling auto-updating is recommended. To disable auto-updating in Google Chrome, refer to this Web page: http://www.chromium.org/administrators/turning-off-auto-updates.

15. If you want to have additional information available about the current state of its window, enable the browser Status Bar (visible at the bottom of the screen):

    - In Mozilla Firefox, select **View** > **Status Bar**
    - In Microsoft Internet Explorer, select **View** > **Toolbars** > **Status Bar**
    - In Google Chrome, this feature is not available

16. For all browsers you intend to use, disable popup-blocking functionality.

10.1.3 Setting Content Manager Explorer preferences

When a Content Manager Explorer user logs on for the first time, they must set their preferred interface language and regional settings and optionally add the Content Manager Explorer Web address to Favorites.

10.1.4 Content Manager Explorer troubleshooting reference

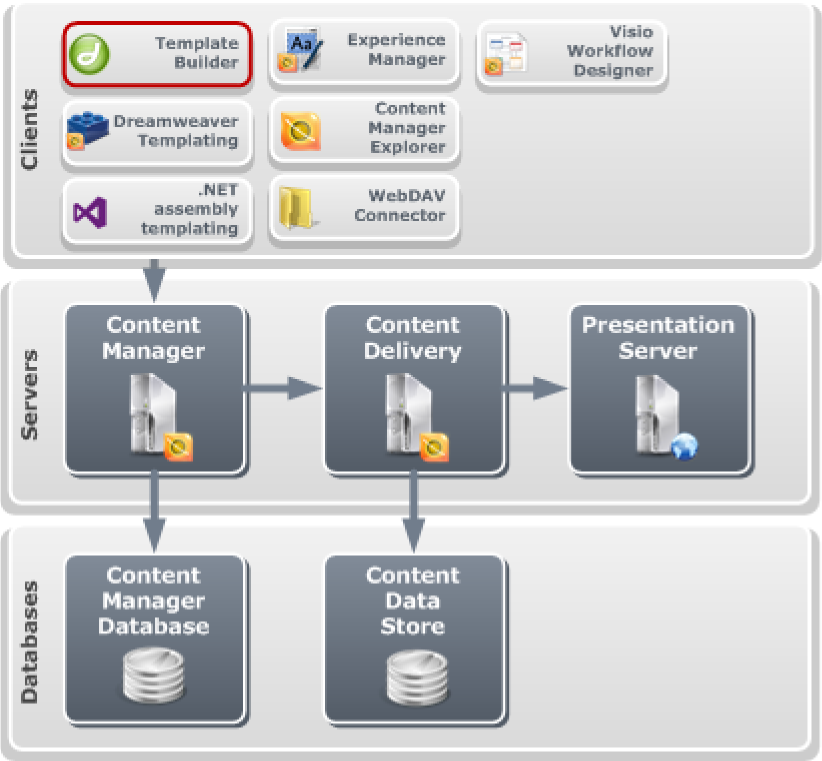
This troubleshooting section describes how to fix the most common problems with the Content Manager Explorer.

Problem	Diagnosis	Action
Accessing the GUI pops up an error with 'Object required' in the **Details** area.	Your browser is using a pop-up blocker such as the one included in the Google Toolbar.	Disable pop-up blocker(s) for the Content Manager Explorer Web site.

Problem	Diagnosis	Action
One of the following happens: • Right-clicking pops up an error with 'Object does not support this property or method' in the **Details** area. • Trying to open a dropdown list results in a series of JavaScript errors.	Your browser settings are incorrect.	Configure your browser as instructed.

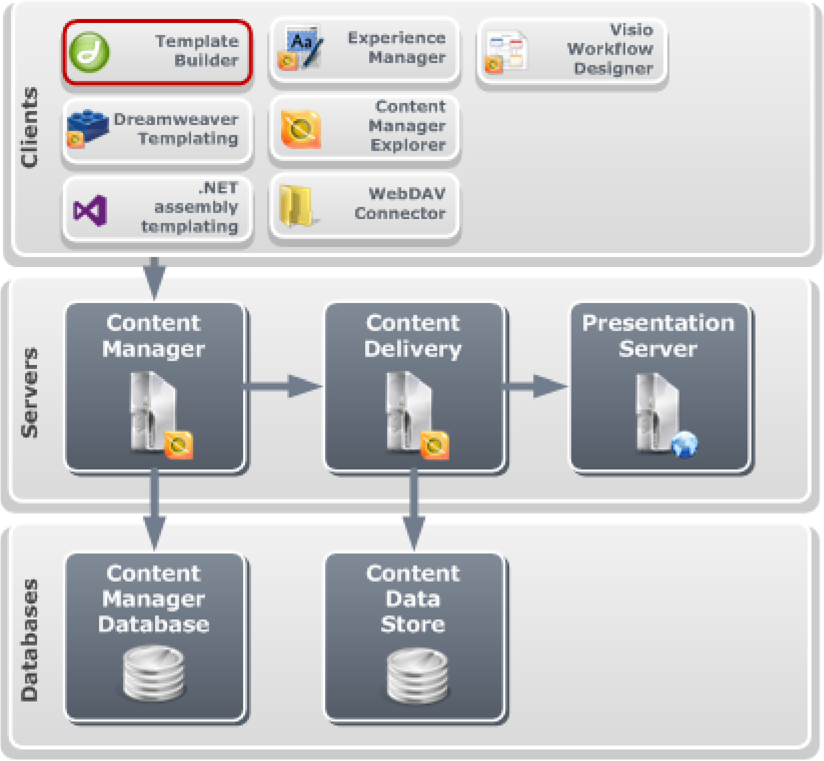
10.2 Setting up Template Builder

This section explains how to set up Template Builder on a client machine. Template developers use Template Builder to create, edit and debug Component Templates or Page Templates by combining Template Building Blocks.



10.2.1 Template Builder prerequisites

The prerequisites of the Template Builder client application. Template Builder enables template developers to combine Template Building Blocks into a Template. Template Builder requires no license at all to run, neither on the client machine nor on the Content Manager server machine.



Template Builder has the following requirements:

Operating system

Template Builder runs on any Microsoft Windows release supported for the Content Manager Explorer client.

.NET Framework

Template Builder requires Microsoft .NET Framework 4.0 or 4.5 (SDL recommends .NET 4.5).

Internet browser

Template Builder requires one of the following Web browsers:

- Microsoft Internet Explorer 10.0 (recommended)
- Microsoft Internet Explorer 9.0

10.2.2 Configuring internet options on the Template Builder client machine

You need to configure certain internet options on the client machine where you will be running Template Builder.

Steps to execute

1. Start Internet Explorer on the client machine where you will be running Template Builder.
2. Click the gear icon at top right and select **Internet Options** from the menu.

   The **Internet Options** dialog opens.
3. Access the **Security** tab and select the **Local Intranet** zone. Then click **Custom Level**.

   The dialog **Security Settings - Local Intranet** opens.
4. Find the settings group called **.NET Framework-reliant components** and under the item **Run components signed with Authenticode**, select **Enable**. Then click **OK**.

   The dialog closes.
5. Click **OK** again.

   The **Internet Options** dialog closes.
6. Close Internet Explorer.

10.2.3 Installing Template Builder

Install Template Builder by running the installer available in the Content Manager Explorer.

Steps to execute

1. To install Template Builder, set up a Microsoft Windows client machine so that it can run Content Manager Explorer.
2. On the client machine, use Internet Explorer (not Mozilla Firefox or Google Chrome) to access the Content Manager Explorer Web site.
3. Select the **Tools** tab from the Ribbon toolbar.
4. Click **Template Builder**. In the dialog that pops up, confirm that you want to install Template Builder.
5. Follow the instructions to install Template Builder.

10.2.4 Creating Experience Manager Template Building Blocks in Template Builder

To make published content editable in Experience Manager , you need to create Templates that include certain predefined Template Building Blocks. To be able to use these default Template Building Blocks, you must make them available in Template Builder first.

Steps to execute

1. To start Template Builder, access Content Manager Explorer, select the **Tools** tab of the ribbon toolbar, and select **Template Builder**. (If Template Builder is not yet installed on your client computer, perform this step in Internet Explorer.)

2. Select **Tools** > **Create Default Building Blocks** to create default Template Building Blocks in Template Builder.

3. Select a location for the default Template Building Blocks and click **OK**.

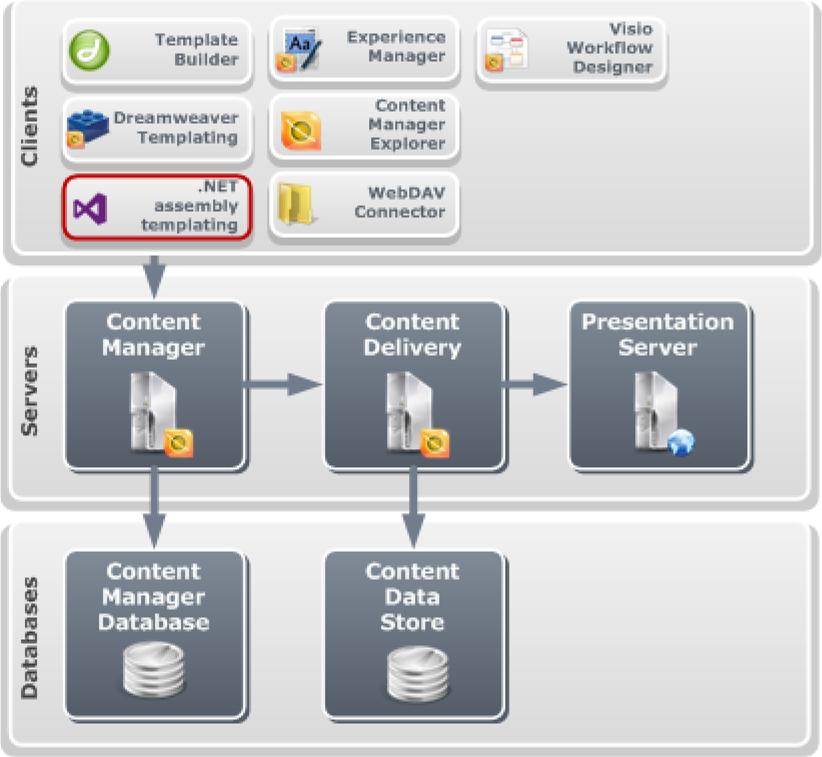
   A dialog appears confirming the default templates were successfully installed. When you create a Template, these default templates appear as **Building Blocks**.

Result

A dialog confirms that Template Builder successfully loaded the default Template Building Blocks for use in Templates.

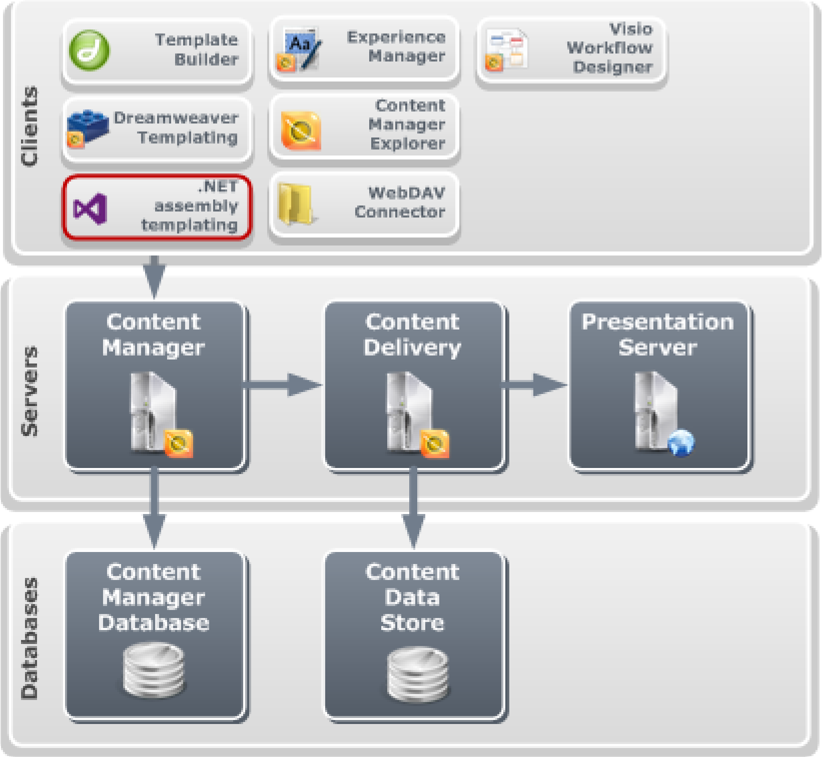
10.3 Setting up .NET assembly templating

This section explains how to set up .NET assembly templating on a client machine. Use .NET assembly templating to create and upload .NET assemblies that can be used as Template Building Blocks in a Compound Template.



10.3.1 .NET assembly templating client prerequisites

This topic lists the required and supported software for creating and uploading .NET assemblies into Template Building Blocks.



You create .NET assemblies and upload them into Template Building Blocks to create parts of a Compound Template. You then use Template Builder to assemble a Compound Template from these and other Template Building Blocks.

To create the .NET assembly, the client machine requires Microsoft .NET Framework 4.0 or 4.5 (SDL recommends using version 4.5). In addition, you will probably want to use a .NET development IDE. SDL recommends using the Microsoft Visual Studio 11 environment, but any IDE that produces .NET assemblies should work.

Uploading the .NET assembly requires the executable file `TcmUploadAssembly.exe` which is placed on the Content Manager server during installation. If you want to run this upload tool client-side from Content Manager Explorer, make sure that you have that client also installed on your .NET assembly client machine.

.NET assembly templating requires no license.

10.3.2 Setting up .NET assembly templating on a client machine

Use .NET assembly templating to create and upload .NET assemblies that can be used as Template Building Blocks in a Compound Template. You can use the same client machine for .NET assembly templating and for running the Template Builder.

Steps to execute

1. On the Content Manager server machine, navigate to the subfolder `bin\client\` of the Content Manager root location (defaults to `C:\Program Files (x86)\Tridion\`).

2. Copy the following files to any folder on your client machine:

   - `Tridion.Common.dll`
   - `Tridion.Common.xml`
   - `Tridion.ContentManager.Common.dll`
   - `Tridion.ContentManager.Common.xml`
   - `Tridion.ContentManager.dll`
   - `Tridion.ContentManager.xml`
   - `Tridion.ContentManager.Publishing.dll`
   - `Tridion.ContentManager.Publishing.xml`
   - `Tridion.ContentManager.Queuing.dll`
   - `Tridion.ContentManager.Queuing.xml`
   - `Tridion.ContentManager.TemplateTypes.dll`
   - `Tridion.ContentManager.TemplateTypes.xml`
   - `Tridion.ContentManager.Templating.dll`
   - `Tridion.ContentManager.Templating.xml`
   - `Tridion.ContentManager.TypeRegistration.dll`
   - `Tridion.ContentManager.TypeRegistration.xml`
   - `Tridion.Logging.dll`
   - `Tridion.Logging.xml`
   - `TcmUploadAssembly.exe`
   - `TcmUploadAssembly.exe.config`

3. When you create a .NET assembly to use in a Compound Template:

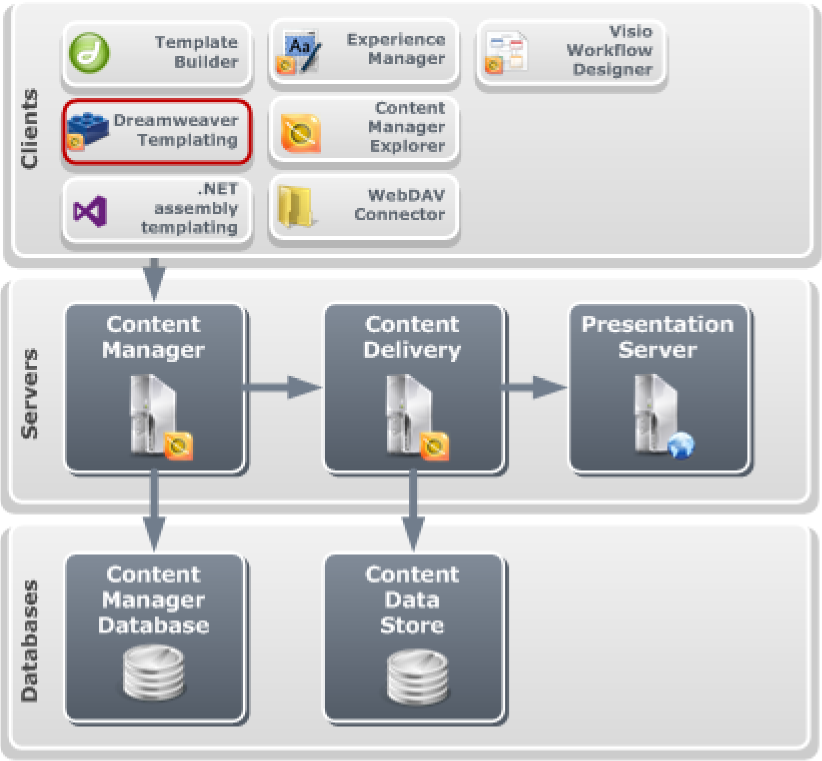
    a. Add these .NET assemblies (DLL files) to your project.

    b. Upload your compiled .NET assembly using the `TcmUploadAssembly.exe` tool.

Note: Do not add these .NET assembly to the Global Assembly Cache. This is not necessary.

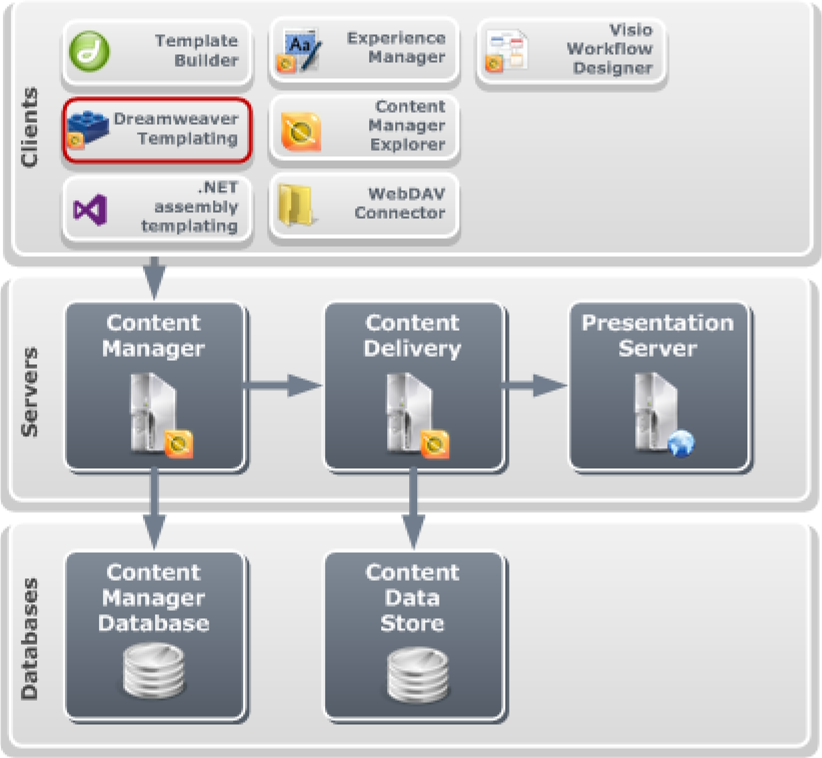
10.4 Setting up Dreamweaver templating

In Adobe Dreamweaver, you can create and edit Adobe Dreamweaver Templates (`.dwt` files) for use in Compound Templates using WebDAV. This section explains how to set up Dreamweaver templating on a client machine by setting up a site that connects to the Content Manager using WebDAV.



10.4.1 Dreamweaver templating prerequisites

This topic lists the required and supported software for creating and uploading Dreamweaver Template Building Blocks into Content Manager.



Dreamweaver templating requires one of the following versions of Dreamweaver in order to run:

- Adobe Dreamweaver CS6
- Adobe Dreamweaver CS5

Dreamweaver requires WebDAV Connector to be configured on the Content Manager server, but requires no license.

10.4.2 Connecting from Adobe Dreamweaver to the Content Manager Server

Set up a set up a site in Adobe Dreamweaver hat connects to the Content Manager using WebDAV allowing you to create and edit Adobe Dreamweaver Templates (`.dwt` files) for use in Compound Templates using WebDAV.

Steps to execute

1. Ensure that WebDAV is configured for Basic authentication only.
2. To enable LDAP users to connect to the Content Manager server from WebDAV, set the `webdav` virtual folder on the Content Manager to accept only Anonymous authentication.
3. Start Adobe Dreamweaver on the client machine.
4. In the **FILES** window, select **Manage Sites** from the drop-down list.
5. In the **Manage Sites** menu, select**New**.

The **Site Setup for <dummy name>** wizard (where <dummy name> is a name proposed by Dreamweaver) opens in order to guide you through the process of creating a new site.

6. In the initial screen, specify a **Site Name** and a **Local Site Folder** for your site.

7. Select **Servers** on the left.

   A (possibly empty) list of servers appears on the right.

8. Click the **Add** button (marked '+') under the list.

   A popup opens for configuring a new server.

9. Enter a **Server Name** to your liking and select **WebDAV** from the **Connect using** drop-down menu.

   The input fields in the popup change.

10. Fill in the fields as follows:

    URL

    Set to the Web address of the WebDAV server for Content Manager. This URL is the URL of the Content Manager Explorer Web site with the folder name `/webdav/` added. For example, `http://cms.myorg.com/webdav/`.

    Entering this URL adds all of the Publications in this installation of the Content Manager, plus all of their contents, to the site. This can slow down Adobe Dreamweaver. If your Dreamweaver Templates will be stored in only one Publication, or better yet, in one specific Folder, enter the WebDAV URL of that specific location here. For example, if your Dreamweaver Templates are all located in the Publication called `EngPub` in the Folder `Building Blocks\Templates\Dreamweaver` (or its subfolders), fill in the following URL:

    ```
    http://cms.myorg.com/webdav/EngPub/Building%20Blocks/
    Templates/Dreamweaver/
    ```

    Username

    Set to the name of a user authenticated to access the WebDAV server (Basic authentication)

    Password

    Set to the password of a user authenticated to access the WebDAV server (Basic authentication)

    Check the **Save** check box to allow the user to interact with WebDAV without having to reenter their credentials.

11. Click **Test** to test your connection. If the connection can be made, a popup appears reporting success.

    Click **OK** to close the popup.

    Note: If you cannot connect to the Content Manager, ensure that WebDAV is set up and properly configured on the Content Manager server.

12. Click **Advanced** at the top.

The advanced options appear.

13. Select **Enable file check-out** and specify a **Check-out Name** (username) and **Email Address**.
14. Click **Save** to create your new site.

    The **Manage Sites** dialog reappears.
15. Click **Done** to close the **Manage Sites** dialog.
16. Press **Ctrl+U** or select **Edit** > **Preferences** from the Dreamweaver main menu to open your Dreamweaver preferences. From the list on the left, select the **New Document** category, which contains settings for any new document created in this site.
17. In this screen, do the following:

    a. Under **Default encoding**, clear the option **Use when opening existing files that don’t specify an encoding**

    b. Under **Unicode normalization form**, select **C (Canonical Decomposition, followed by Canonical Composition)**.

    c. Click **OK** to close the dialog.
18. In the **FILES** tab, select the site you just created and click **Get File(s)**.

Result

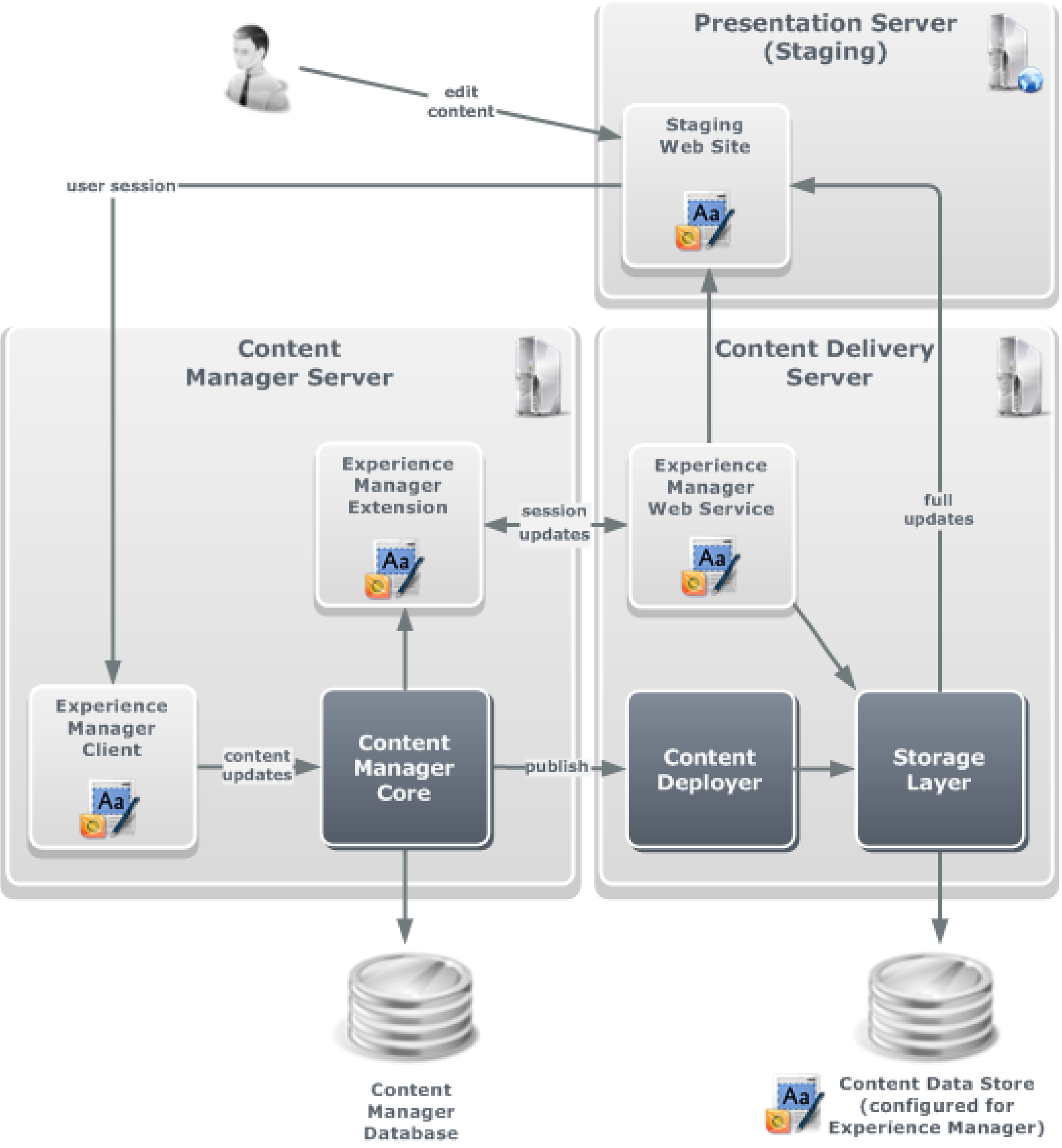
You have successfully connected to the Content Manager server and you can now start creating Dreamweaver templates by checking out individual files.

10.5 Setting up the Experience Manager client

This section explains how to install the Experience Manager client on the Content Manager side. To do so, perform installation and configuration tasks on the Content Manager server. Note that to make Experience Manager work, Experience Manager Server Roles must be installed on the Content Delivery side.

10.5.1 Experience Manager system diagram

Experience Manager consists of a client (editor) and an extension on the Content Manager server, and a Web service, Web site extension and database on the Content Delivery side.



System components installed on Content Manager:

Experience Manager client

The client is delivered as a GUI extension of the Content Manager of SDL Tridion, in the form of an Editor that implements Model-View-Controller software architecture. The client allows users to see a user-specific version of the staging Web site, showing each Web page with changes made by the current user (if any).

Content Manager Core, Content Manager Database

Any change that the user makes on a Web page, and then saves, leads to an update to the corresponding items in the Content Manager Database. The Content Manager Core also makes these changes available immediately to the user, through the Experience Manager extension; and eventually to all visitors of the staging Web site, through the normal publish mechanism.

Experience Manager Extension

The Experience Manager Content Manager Extension calls the Experience Manager Web service with any updates that the user made to items on the Web page, rendered into publishable form. This bypasses the publish cycle.

System components installed on Content Delivery:

Experience Manager Web service

The Experience Manager Web service receives updates from the Content Manager server and uses the **Storage Layer** to store the user-specific rendered content in a Content Data Store. Only the user who is making updates sees this content while editing; other visitors of the staging Web site see the latest content that is checked in on the Content Manager side.

Content Data Store (database configured for Experience Manager)

A Content Data Store is used for storage and retrieval of content within an editing session. The Content Data Store must be a relational database; a file system is not a supported storage medium for Experience Manager .

Content Deployer

The Content Deployer runs independently of the Experience Manager Web service to ensure normal deployment of content through publishing. Users of the user interface use the Content Deployer when they finish editing and make their changes available to all visitors of the staging Web site, or when they publish explicitly to the staging or live Web site.

Staging Presentation Server:

Staging Web site

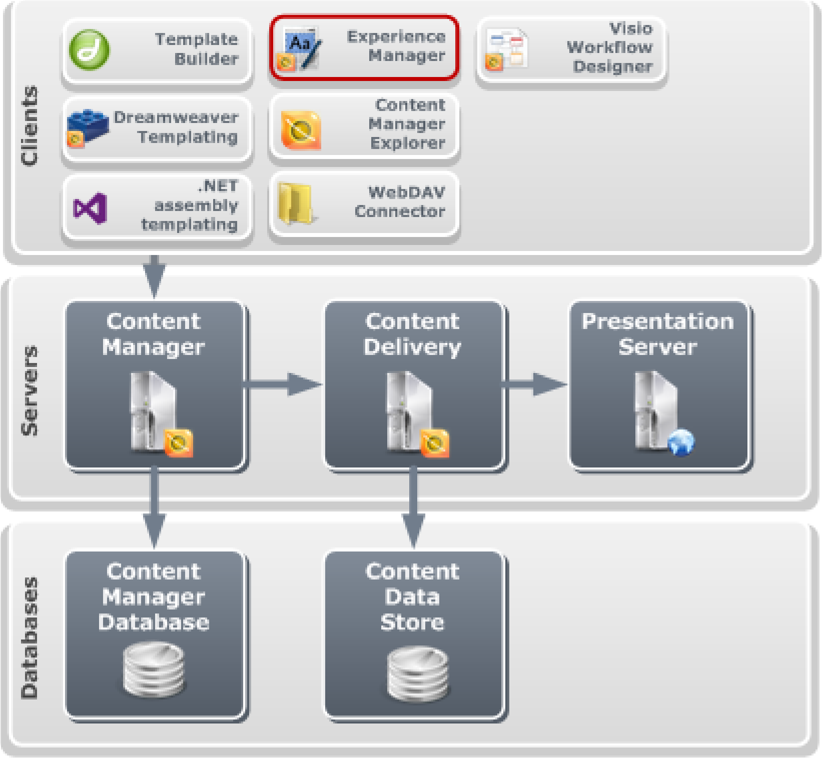
The staging Web site is where users edit content in Experience Manager . The staging Web site always shows the latest published content.

Experience Manager Web site extension

The Experience Manager Web site extension ensures that when a user is editing the Web page, it shows the latest content as updated by the specific user who is logged in, and as checked in by other users, and stores the user's modifications in the Content Data Store (database configured for Experience Manager).

10.5.2 Experience Manager client prerequisites

This topic lists the required and supported software for the Experience Manager client. The prerequisites are the same as for Content Manager Explorer.



Recommended prerequisites

SDL recommends the following set of prerequisites for Experience Manager :

Prerequisite	Recommended product release
Operating system	Microsoft Windows 8 (x86 or x64), or Apple Mac OS X 10.8 Mountain Lion
Internet browser	Microsoft Internet Explorer 10.0, or Safari 6.0 for the Mac (if using Mac OS X)

Operating system languages

The operating system on which the Experience Manager runs must be in one of the following languages:

- US English
- French
- German
- Spanish
- Dutch
- Japanese

Operating system versions and editions

Supported:

OS Version	Service pack	Architecture
Windows 8	n/a	x86
Windows 8	n/a	x64
Windows 7	SP1	x86
Windows 7	SP1	x64
Windows Server 2012	n/a	x64
Windows Server 2008 R2	SP1	x64
Apple Mac OS X 10.8 Mountain Lion	n/a	n/a

Deprecated: None

Note: The SDL Tridion installation media contains a Web page, `\Redistributables\Readme.htm`, with links to the Microsoft download pages for the Service Packs listed in the table.

Web browser

Supported:

- Internet Explorer 10.0 (on Windows 8 and Windows Server 2012)
- Internet Explorer 9.0 (on other Windows versions)
- Safari 6.0 for the Mac
- Google Chrome, latest version
- Mozilla Firefox, latest version

Deprecated: None

For information about the supported operating systems for each browser, contact the vendor of the browser you intend to use.

Updating supported browser versions

The list of supported Web browser versions was known and tested at the time of release. However, new Web browser versions may have been released since that time that are also supported but have not been subjected to testing. As a result, the Content Manager Explorer may display a 'browser version not supported' warning contrary to this list, or it may display no warning even though the browser version you use is not listed.

To change the response to a specific browser version, configure the `system.config` file:

1. Access the subfolder `WebUI\WebRoot\Configuration` of the Content Manager root location (defaults to `C:\Program Files (x86)\Tridion\`).

2. Open `system.config` in a plain-text or XML editor. The `<supportedbrowsers>` section near the top of the file contains `browser` elements for supported browser versions.

3. To stop SDL Tridion from issuing a warning for an unsupported browser version, copy and paste an existing `browser` element for an earlier version and set the set the `name` and `regExp` attributes to values that represent the supported version.

4. To make SDL Tridion issue a warning for a browser version that is no longer listed as supported, remove the `browser` element that has its `name` attribute set to the browser version you want to issue a warning for.

5. Save and close `system.config`.

6. Restart IIS.

10.5.3 Configuring your Web browser for Experience Manager

Experience Manager runs in the same browsers as Content Manager Explorer, and those browsers require the same configuration as for Content Manager Explorer. In addition, under certain conditions, you may also need to add a certificate exception to the browser.

Steps to execute

1. To configure the browser, on your client machine, start Internet Explorer (even if you do not intend to use Internet Explorer as your browser).

2. In Internet Explorer, if you do not see the menu bar, press your left **Alt** key on the keyboard to display it.

3. Select **View** > **Encoding** > **Unicode** from the menu. This sets the browser encoding to Unicode.

4. Select **Tools** > **Internet Options** from the menu. The **Internet Options** dialog opens.

5. Access the **Security** tab, select the **Local intranet zone** and click **Sites**. The **Local intranet** dialog with generic site settings opens.

6. Select **Advanced** to open a **Local intranet** dialog listing sites associated with this zone.

7. Enter the URL of your Experience Manager Web site and click **Add**. The site is added to the list. Click **Close** to close this dialog, and **OK** to close the generic sites dialog. The settings for the **Local intranet zone** reappear.

8. Click **Custom Level.** The **Security Settings - Local intranet Zone** dialog opens.

9. In the list of settings, find the following settings and set each to the value `Enabled`:
   - ActiveX:
     - **Run ActiveX controls and plug-ins**
     - **Script ActiveX controls marked safe for scripting**
   - Downloads:

    - **File downloads**
  - Miscellaneous:
    - **Submit non-encrypted form data**
  - Scripting:
    - **Active scripting**
    - **Allow Programmatic clipboard access**

  Click **OK** to apply these settings and close the dialog. The **Internet Options** dialog reappears, still showing the **Security** tab.

10. Select the **Privacy** tab and click **Advanced**. The **Advanced Privacy Settings** dialog opens.

11. Select **Override automatic cookie handling** and then select **Always allow session cookies**. Then click **OK** to close the **Advanced Privacy Settings** dialog. The **Internet Options** dialog reappears, still showing the **Privacy** tab.

12. Click **OK** to close the **Internet Options** dialog.

13. If you intend to use Mozilla Firefox as your browser, do the following:

    a. While logged in as an end user, launch Firefox and type `about:support` in the address bar.

    b. In the **Application Basics** area, check the **Profile Directory** property and click **Open Containing Folder** next to it.

    Windows Explorer (Windows) or Finder (Mac) opens, showing the Firefox profile folder.

    c. Close Firefox.

    d. In the folder, check for the presence of a file called `user.js`. If it exists, open it in a plain-text editor; if it does not exist, create it in a plain-text editor.

    e. Add the following lines anywhere in the `user.js` file:

```
user_pref("capability.policy.policynames", "allowclipboard");
user_pref("capability.policy.allowclipboard.sites", "http://<cmsname>");
user_pref("capability.policy.allowclipboard.Clipboard.cutcopy", "allAccess");
user_pref("capability.policy.allowclipboard.Clipboard.paste", "allAccess");
user_pref("browser.link.open_newwindow", 2);
```

    in which *<cmsname>* is the domain name of the Content Manager server machine.

    Note: You can also add comments on separate lines. Start these lines with a double forward slash (`//`).

    f. Save and close the `user.js` file.

    g. If you are using Firefox on Mac OS X, also do the following:

    - In the Apple menu, select **System Preferences**, then **Keyboard and Mouse**, then **Keyboard Shortcuts**
    - Under **Full Keyboard Access**, select **All controls**.

Mozilla Firefox is now properly configured.

14. If you intend to use Google Chrome as your browser, SDL Tridion recommends to disable the auto-update functionality of Chrome.

    By default, Google Chrome is configured to auto-update to its latest version. Because this user interface cannot be guaranteed to work with a newer version of Google Chrome,disabling auto-updating is recommended. To disable auto-updating in Google Chrome, refer to this Web page: http://www.chromium.org/administrators/turning-off-auto-updates.

15. If you want to have additional information available about the current state of its window, enable the browser Status Bar (visible at the bottom of the screen):

    - In Mozilla Firefox, select **View** > **Status Bar**
    - In Microsoft Internet Explorer, select **View** > **Toolbars** > **Status Bar**
    - In Google Chrome, this feature is not available

16. For all browsers you intend to use, disable popup-blocking functionality.

17. Check if the following is the case:

    - The user accesses Content Manager Explorer through an HTTPS connection.
    - No trusted certificate is in place on that Web site.

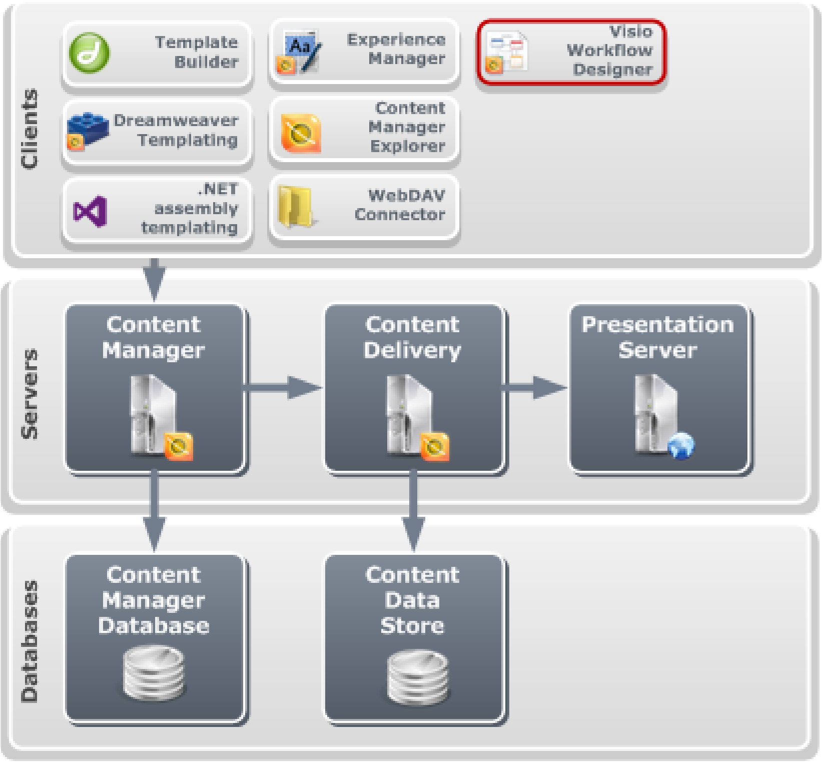
18. If these conditions hold, a user logging on to the user interface from a Web page on the staging Web site may be confronted with a message that the certificate is not trusted. Ensure that the user's browser trusts the certificate and does not ask this question again, by adding a certificate exception. (Alternatively, you can of course also add a trusted certificate, or allow HTTP access.)

Chapter 11 Installing Visio Workflow Designer

To implement Workflow, install Visio Workflow Designer, a plug-in for Microsoft Visio. Visio Workflow Designer runs on a client machine and communicates with Visio Workflow Server on the Content Manager server.

11.1 Visio Workflow Designer client prerequisites

Required and supported software for the Visio Workflow Designer client application. Visio Workflow Designer requires no licenses at all, neither on the client side nor on the Content Manager Server.



Recommended prerequisites

SDL recommends using version 4.5 of the .NET Framework.

Recommended prerequisites

SDL recommends the following set of Visio Workflow Designer prerequisites:

Prerequisite	Recommended product release
Operating system	Microsoft Windows 8 (x86 or x64), or Apple Mac OS X 10.8 Mountain Lion
Internet browser	Microsoft Internet Explorer 10.0, or Safari 6.0 for the Mac (if using Mac OS X)

Operating system languages

The operating system on which Visio Workflow Designer runs must be in one of the following languages:

- US English
- French
- German
- Spanish
- Dutch
- Japanese

Operating system versions and editions

Supported:

OS Version	Service pack	Architecture
Windows 8	n/a	x86
Windows 8	n/a	x64
Windows 7	SP1	x86
Windows 7	SP1	x64
Windows Server 2012	n/a	x64
Windows Server 2008 R2	SP1	x64
Apple Mac OS X 10.8 Mountain Lion	n/a	n/a

Deprecated: None

Note: The SDL Tridion installation media contains a Web page, `\Redistributables\Readme.htm`, with links to the Microsoft download pages for the Service Packs listed in the table.

Third-party software components

The Workflow Designer client application requires the following product:

Microsoft Visio 2010 SP1 (32-bit), English version

In addition, Workflow Designer requires:

- MS XML 6.0
- Version 4.0 or 4.5 of the Microsoft .NET Framework.

11.2 Running the Visio Workflow Designer installer

To install the Visio Workflow Designer client application, run the Visio Workflow Designer installer.

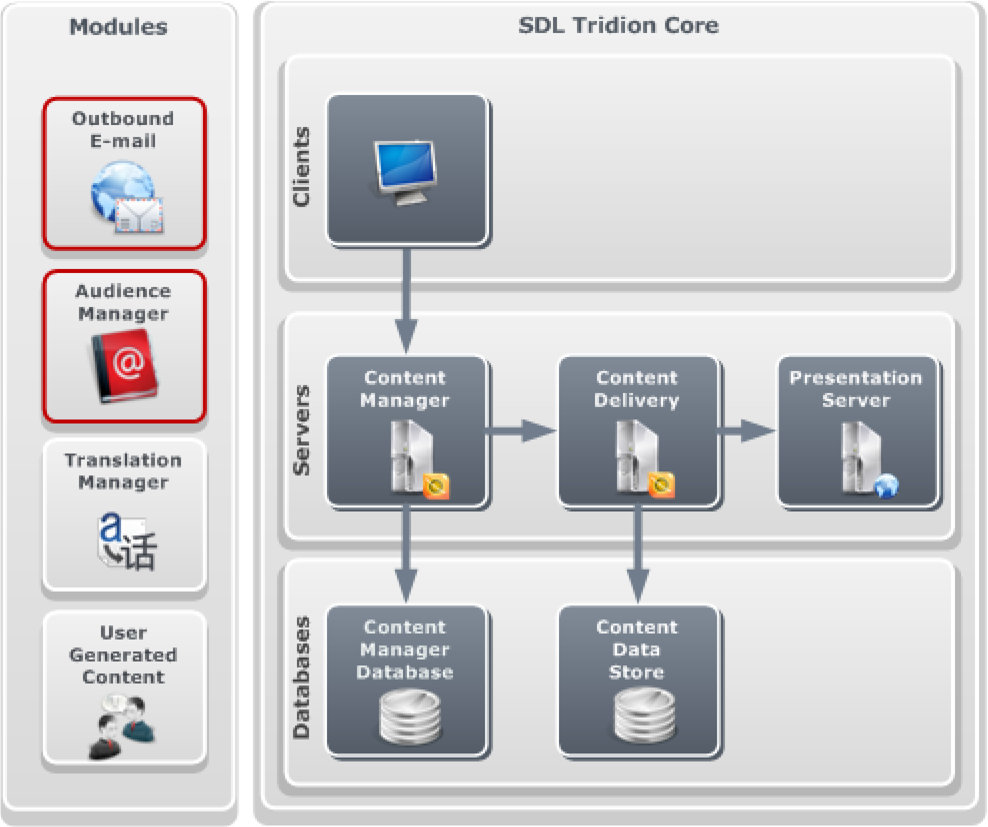
Steps to execute

1. Access the SDL Tridion installation media and navigate to the `Content Delivery\roles\` folder.
2. Navigate to the `Content Manager\Additional Products\Visio Workflow Designer` folder.
3. Double-click the shortcut **Install Visio Workflow Designer Client**.

   The Visio Workflow Designer client application installer starts up.
4. Follow the instructions to install Visio Workflow Designer.

Chapter 12 Installing Audience Manager and Outbound E-mail

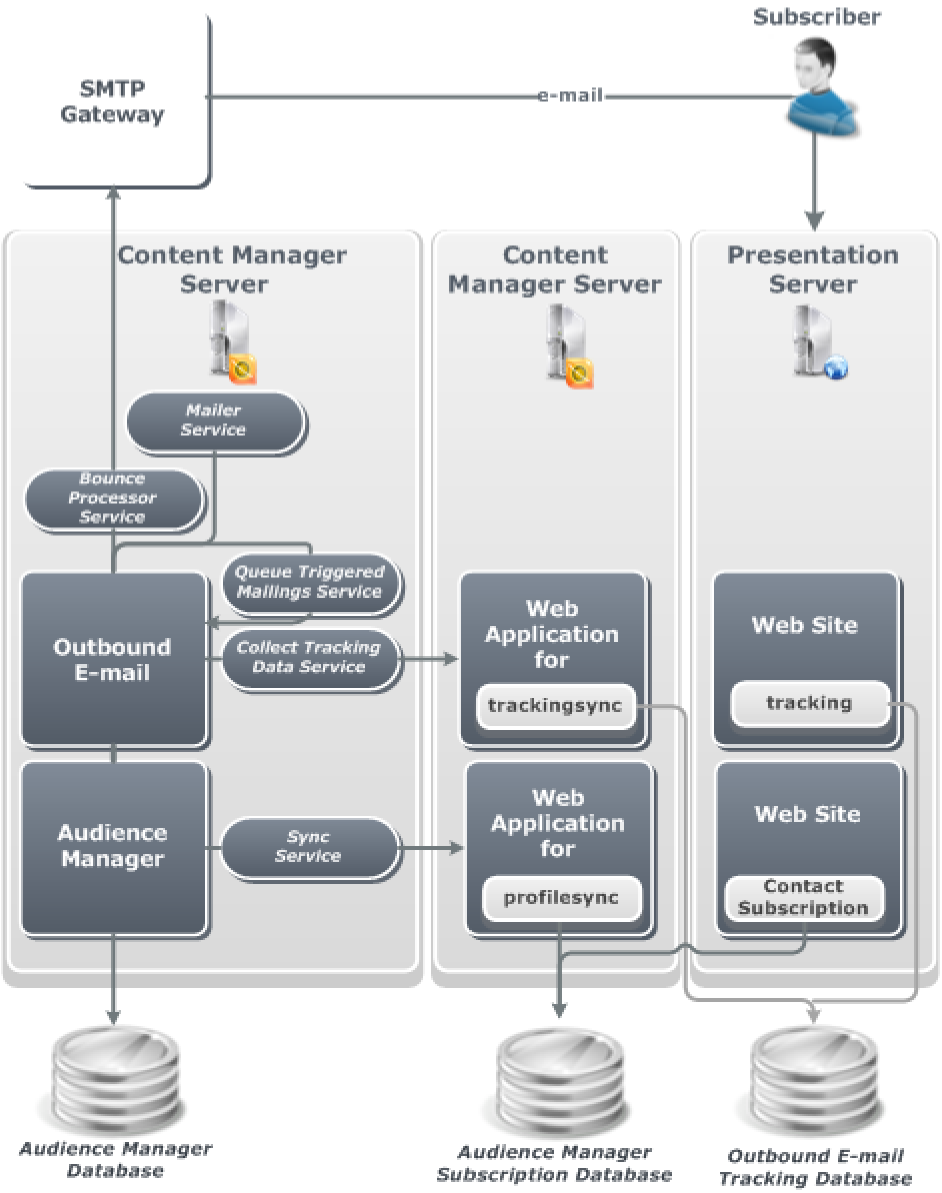
Audience Manager is for gathering information about Contacts and audiences such as interests, preferences and characteristic. Outbound E-mail is for managing e-mail campaigns (Outbound E-mail requires Audience Manager). Audience Manager and Outbound E-mail require you to install software components on the Content Manager side and presentation server side of your SDL Tridion installation.



12.1 Audience Manager and Outbound E-mail system diagram

The system diagram shows the software components and services that send and track Mailings for Outbound E-mail and collect Contact data for Audience Manager.

The following diagram shows the system architecture of Audience Manager and Outbound E-mail:



Audience Manager

Audience Manager consists of software components installed on the Content Manager that allow you to manage Contacts and Segments, Subscription pages installed on a Presentation Server to allow users to subscribe, confirm subscription, change their profile and unsubscribe (to Mailings), and a Subscription Synchronization page installed on a Synchronization Server to synchronize data between the two systems.

On the Content Manager Server

Audience Manager runs on a machine that is running the Content Manager of SDL Tridion. When you install Audience Manager, the installer installs GUI extensions for managing Address Books, Contacts and Segments in the Content Manager Explorer, and the Synchronization service.

- **Synchronization service**—synchronizes Contact information and Segments between the Audience Manager database and the Subscription database on the presentation side.

Audience Manager database

The Audience Manager database is the database for storing Contacts (Oracle or Microsoft SQL Server) on the Content Manager side.

Synchronization Server

The Synchronization Server synchronizes changes in Contact details between the Audience Manager database on the Content Manager side and the Subscription database on the presentation side. You install the `profilesync` Web page in a .NET or Java Web application.

Presentation Server for Contact subscription

The Presentation Server is the Web application that accepts subscriptions and changes to subscriptions. You need to deploy Subscription Web pages (`subscribe`, `confirm`, `changeprofile` and `unsubscribe`) on a publicly accessible Web site.

Subscription database

The Subscription Management database (Oracle or Microsoft SQL Server) stores Contact information collected on the Presentation Server.

Outbound E-mail

Outbound E-mail consists of software components installed on the Content Manager that allow you to send Mailings, a Tracking page installed on a Presentation Server to track whether an e-mail recipient opened the e-mail or clicked any links (to monitor the effectiveness of an e-mail campaign), and a Tracking Synchronization page installed on a Synchronization Server to synchronize data between the two systems. You need to add code to your templates to track links.

On the Content Manager Server

You need to install the **Outbound E-mail Core** on every Content Manager Server that runs the Content Manager Explorer so that users can manage Mailings and Distribution Lists. The following Outbound E-mail services are installed as part of the core:

- **Queue Triggered Mailings service**—is responsible for handling triggered mailings
- **Collect Tracking Data service**—retrieves tracking information from the Tracking database on the presentation side
- **Bounce Processor service**—keeps track of bounced e-mails and updates the mailing statistics by periodically polling a preconfigured POP3 or IMAP mailbox account for bounced e-mails
- **Mailer service**—sends out the mailings to the SMTP Gateway server(s)

Synchronization Server

The Synchronization Server retrieves tracking information from the Tracking database on the presentation side. You install Tracking components in a .NET or Java Web application. `trackingsync` (JSP or ASPX) —synchronizes the Outbound E-mail database and the Tracking database. Only the Outbound E-mail Server requires access to this component, therefore for security reasons it is recommended to restrict access to this component

Presentation Server for Tracking

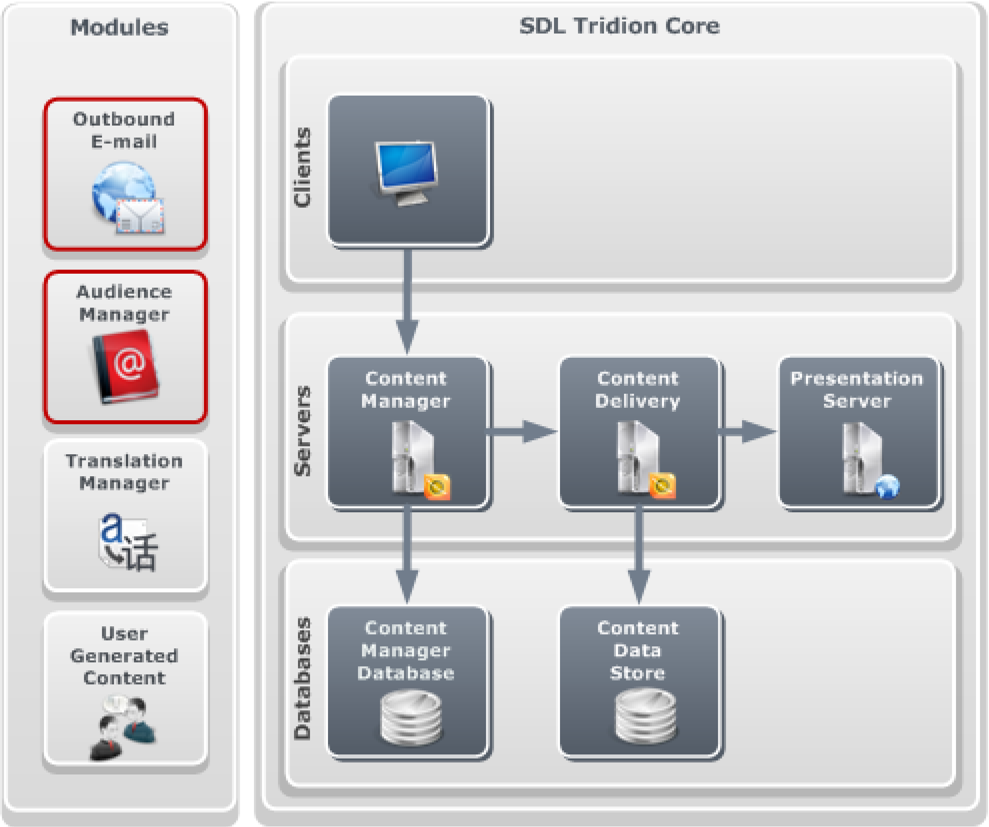
The Presentation Server is the Web application that tracks e-mail response. You need to deploy the `tracking` (JSP or ASPX) to register e-mail opening and link clicking. You need to install this component on a publicly accessible Web site.

Tracking database

The Tracking database (Oracle or Microsoft SQL Server) stores tracking information collected on the Presentation Server.

12.2 Audience Manager and Outbound E-mail prerequisites

This topic lists the required and supported software for part of the Audience Manager and Outbound E-mail systems.



Content Manager side prerequisites

Audience Manager and Outbound E-mail is fully integrated into the Content Manager and is installed as a part of the Content Manager installation, therefore Content Manager server prerequisites apply.

Database prerequisites on Content Manager side

Audience Manager and Outbound E-mail require a database to store Contacts. The supported versions of Oracle or Microsoft SQL Server are identical to those of the Content Manager database.

Additionally, the **Oracle Database Client** must be installed with the following options:

- Oracle Database Utilities (specifically, your database client requires SQL*Loader installed)
- Oracle Windows interfaces—Oracle Data Provider for .NET 2.0

Required third-party software

Outbound E-mail requires a number of third-party software components to work:

- SMTP server—for sending out e-mails
- POP3 or Internet Message Access Protocol (IMAP) server—for receiving bounced messages and the like for e-mails Outbound E-mail failed to deliver

Note: For information on how to install these third-party software items, refer to the documentation of these software packages.

Content Delivery side prerequisites

You install Contact Subscription (Audience Manager) and Tracking (Outbound E-mail) on the Presentation Server where your public-facing SDL Tridion-drive Web site is running, and you install Contact Synchronization and Tracking synchronization on your Content Delivery Server (or on a separate dedicated machine). These components require specific Content Delivery Server Roles installed and configured on these machines (described in the installation documentation).

Concerning the prerequisites for these machines, Content Delivery prerequisites apply.

Database prerequisites on Content Delivery side

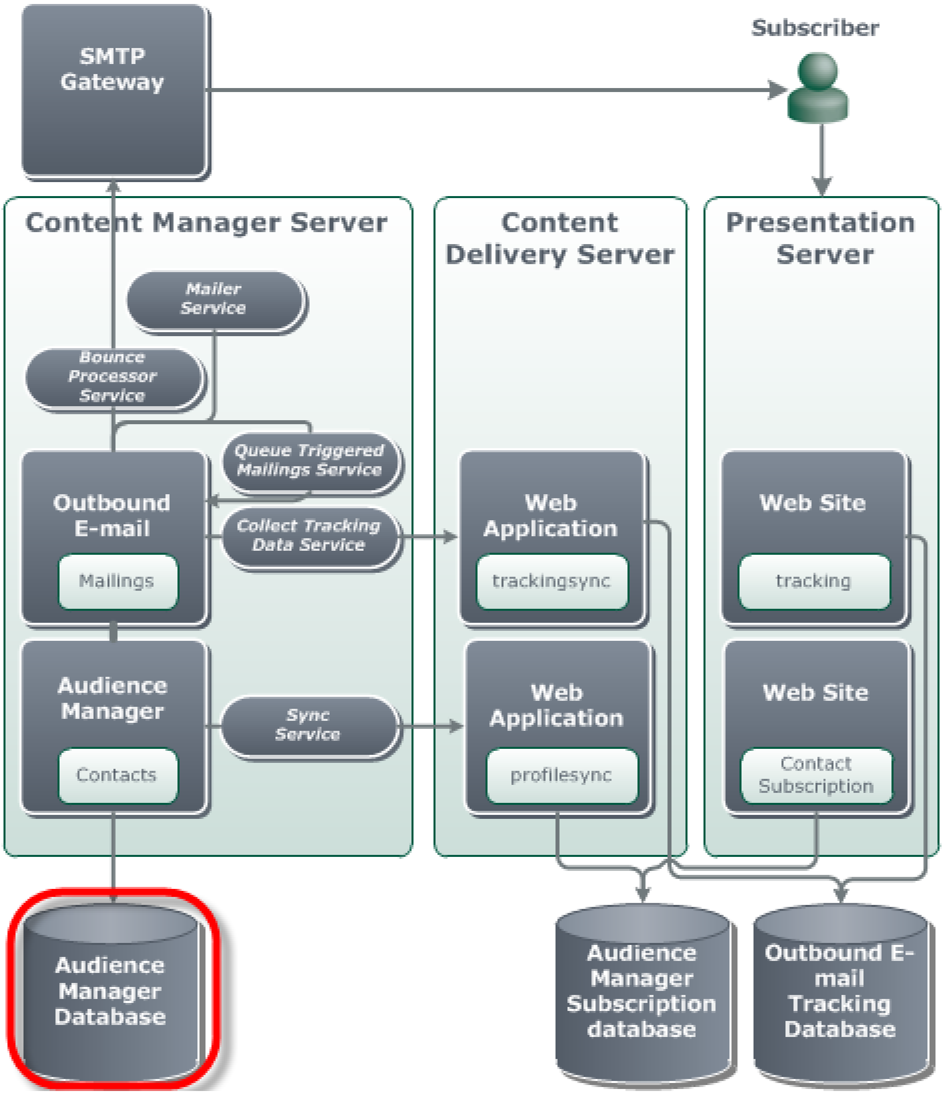
Audience Manager requires a database on the presentation side to store subscriptions and changes to subscriptions, and Outbound E-mail requires a database to store e-mail responses.

The **Tracking database** and **Subscription database** support the same Microsoft SQL Server and Oracle databases as the Content Data Store. **IBM DB2 database is not supported**.

The databases may run on any operating system that supports the specific DBMS you choose to use. Refer to your database vendor documentation to learn which operating systems are supported.

12.3 Creating the Audience Manager database

Audience Manager and Outbound E-mail needs a database. The Audience Manager database stores Contact information.



12.3.1 Creating the Audience Manager database

Use a PowerShell script to create a Microsoft SQL Server or Oracle database.

Requirements

The prerequisites for the Audience Manager Database are the same as for the Content Manager Database.

To install SDL Tridion databases on a Windows machine requires Windows PowerShell 3.0 and Microsoft .NET Framework 4.0 or higher.

Context

Note: It is recommended to install the Audience Manager database on a different machine than Audience Manager, except in development or demo environments.

Steps to execute

1. Open Windows PowerShell from the Windows Start Menu.
2. In PowerShell, depending on your database navigate to one of the following folders on the installation media:
   - `Database\MSSQL\` or
   - `Database\Oracle\`
3. Type the following command to install the Audience Manager database:

   ```
   & '.\Install Audience Manager database.ps1'
   ```

4. Follow the instructions in the PowerShell console to create the database.

Result

You have created the Audience Manager database.

Next steps

When you install Audience Manager and Outbound E-mail, you are prompted to provide configuration details for the Audience Manager database. For more information, see Running the installer (see page 169).

12.3.2 Configuring the Audience Manager database

The `<Database>` section of the configuration file `OutboundEmail.xml` allows you to specify database connection properties. Note that when you run the installer, you are prompted to provide database details to set these configurations automatically.

Steps to execute

1. Open `OutboundEmail.xml` located by default in the `config` subdirectory of your Content Manager root location in a plain-text editor or XML editor.
2. Locate the `<Database>` section, for example:

   ```
   <Database>
       <Type>Oracle</Type>
       <Server>orcl11g</Server>
       <Name>tridion_cm_email</Name>
       <User>tmsdbuser</User>
       <Password>tridion</Password>
       <Timeout>30</Timeout>
       <GetContactsTimeout>30</GetContactsTimeout>
       <LongQueryTimeout>120</LongQueryTimeout>
       <ImportTimeout>300</ImportTimeout>
       <QueueMailingTimeout>300</QueueMailingTimeout>
       <TransactionTimeout>1200</TransactionTimeout>
   ```

```
</Database>
```

3. Specify the following settings:

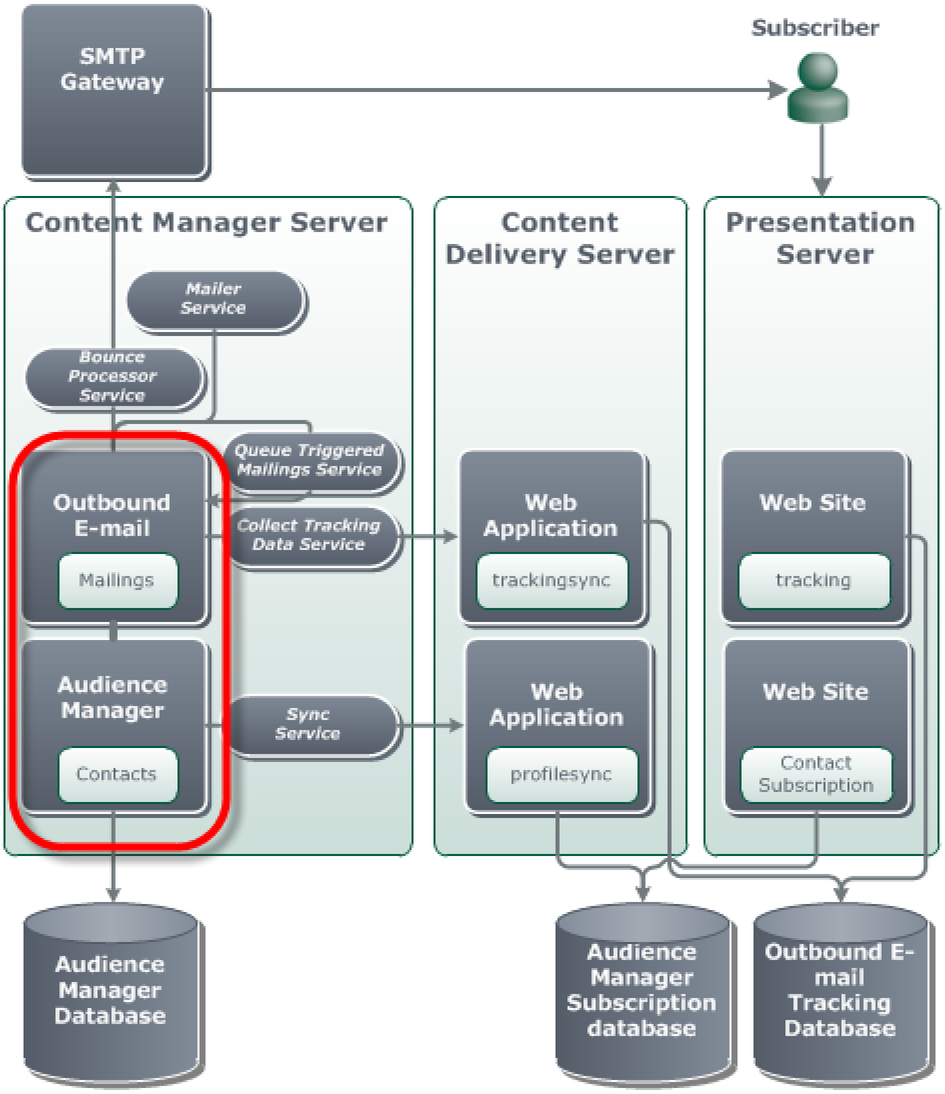
Element	Description
`Type`	The type of Outbound E-mail database you installed. Specify one of the following values: • `mssql` • `oracle`
`Server`	The server where the database is located. Specify `(local)` if the database is running on the same machine as the Outbound E-mail Web site. Note: You created this database during installation.
`Name`	The name of the database you created, by default `tridion_cm_email`.
`User`	The name of the database user you created, by default `TMSDBUser`. For more information, see Encrypting sensitive strings for Audience Manager (see page 171)
`Password`	The password of the database user you created.
`Timeout`	The default time, in seconds, a database connection is allowed to be open for general database requests and all other timeouts if not configured. If this time is exceeded, the connection is closed.
`GetContactsTimeout`	(Optional) The time, in seconds, a database connection is allowed to be open when getting lists of Contacts.
`LongQueryTimeout`	(Optional) The time, in seconds, a database connection is allowed to be open when performing miscellaneous heavy operations such as deleting an Address Book, copying Contacts to a Static Distribution List, refreshing statistics, and so on. If this time is exceeded, the connection is closed.
`ImportTimeout`	(Optional) The time, in seconds, a database connection is allowed to be open when importing Contacts or Keywords. If the time limit is exceeded and the connection is closed, import fails.

Element	Description
`QueueMailingTimeout`	(Optional) The time, in seconds, a database connection is allowed to be open when queuing a Mailing and getting the number count of Contacts to queue for a Mailing. If the time limit is exceeded and the connection is closed, the Mailing is not sent.
`TransactionTimeout`	The time, in seconds, a transaction in the database will timeout. If you experience transaction timeouts, increase the default of 1200 (20 minutes).

4. Save and close `OutboundEmail.xml`.

12.4 Installing Audience Manager and Outbound E-mail

This section describes how to install Audience Manager and Outbound E-mail on a machine on which Content Manager Server is installed.



12.4.1 Running the installer

You install Outbound E-mail and Audience Manager by running the Content Manager Server installer.

Requirements

Before running the installer you need to have created the Audience Manager database. The installer will prompt you for details about the database and database user credentials.

Context

By default, the Content Manager installer installs all software components. You will therefore only need to install Outbound E-mail and Audience Manager if you chose not to when you installed the Content Manager Server, or when you want to outscale components.

Steps to execute

1. Access the SDL Tridion installation media.

2. Navigate to the `Content Manager\` directory.

3. Double-click the executable `SDLTridion2013CM.exe`.

4. Follow the instructions on the screen:
   - You can select individual software components to install by choosing the **Advanced** option (by default, the installer installs all software components).
   - You can install software components you choose not to install by running the installer again and selecting the **Modify** option.

5. After the installer finishes, if the installer does not prompt you to restart your machine, perform a manual reboot.

Result

The installer has created a log file in the following location:

- the subdirectory `SDL\Tridion\Logs\` of the path indicated by the `%PROGRAMDATA%` environment variable (defaults to `c:\ProgramData\`).

The following software components are installed for **Audience Manager**

- GUI Extensions
- Synchronization Service

The following software components are installed for **Outbound E-mail**

- Modular Templating Support (default templates)
- Outbound E-mail Script Extension (VBScript templating)
- Mailer Service
- Queue Triggered Mailings Service
- Bounce Processor Service
- Collect Tracking Data Service

12.4.2 Configuring logging

The `<Log>` section of the configuration file `OutboundEmail.xml` allows you to specify the location on your file system where logging information is generated and the level of generated information.

Steps to execute

1. Open `OutboundEmail.xml` located by default in the `config` subdirectory of your Content Manager root location in a plain-text editor or XML editor.

2. Locate the `<Log>` section, for example:

```
<Log>
    <Folder>C:\Program Files
    (x86)\Tridion\log</Folder>
    <Level>1</Level>
```

```
</Log>
```

3. Specify the following settings:

Element	Description
`Folder`	The location on your file system where logging information is generated.
`Level`	The logging level: • `0`—none • `1`—errors (logs exceptions) • `2`—warnings • `3`—information (logs high level information on synchronization, import progress, bounce processing, and dynamically created queries) • `4`—debug (logs low level information on synchronization) • `5`—trace (traces the internal flow of the system)

Note: As debug and trace logging levels create very large log files, and possibly many synchronization debug files, you should only use these levels temporarily.

4. Save and close `OutboundEmail.xml`.

12.4.3 Encrypting sensitive strings for Audience Manager

In the `OutboundEmail.xml` configuration file, the `<BounceProcessing>` and `<Database>` sections contain a `<Password>` element.

Context

As it is not good practice to include sensitive data in configuration files in cleartext, it is recommended to encrypt sensitive data using the command line tool delivered with Outbound E-mail.

Steps to execute

1. Open a command prompt.

2. Access the SDL Tridion installation media and navigate to the `Content Delivery\roles\` folder.

3. Navigate to the folder `Outbound E-Mail\Encryption` and run the following command:

```
ProtectConfiguration.cmd
```

Result

The command line tool encrypts the passwords specified in the `OutboundEmail.xml` configuration file.

If encryption failed, the command prompt remains open and describes the nature of the errors. Fix the errors before proceeding.

To remove encryption from the passwords, navigate to the folder `Outbound E-Mail\Encryption` on the SDL Tridion installation media and run the following command:

```
UnprotectConfiguration.cmd
```

12.4.4 Configuring Audience Manager and Outbound E-mail licenses

Audience Manager and Outbound E-mail require licenses to work. The exact functionality available is determined by the contents of the specific license file. To obtain a license file, contact SDL Tridion Customer Support. Configure the location of the license file in the `<System>` section of the configuration file `OutboundEmail.xml`.

Steps to execute

1. Open `OutboundEmail.xml` located by default in the `config` subdirectory of your Content Manager root location in a plain-text editor or XML editor.

2. Locate the `<System>` section, for example:

```
<System>
  <LicenseFile />
  <CoreServiceUrl>http://localhost/webservices/CoreService2011.svc/wsHttp</
CoreServiceUrl>
  <MonitoringAgentPort>20131</MonitoringAgentPort>
</System>
```

3. Specify the following settings:

```
<LicenseFile>C:/Program Files (x86)/Tridion/config/license.xml</LicenseFile>
```

4. Save and close `OutboundEmail.xml`.

12.5 Installing SMTP and POP3/IMAP servers

Outbound E-mail requires a SMTP server and a POP3 or IMAP server.

Context

Installing your SMTP server

Outbound E-mail requires a SMTP server for sending out e-mails. For installation instructions, see the third-party documentation for the specific server vendor.

The **Mailer service** needs SMTP connection to the SMTP server (TCP 25).

You configure outgoing mail in the `OutboundEmail.xml` file. For more information, see Configuring outgoing e-mail (see page 174).

Install your POP3 or IMAP server

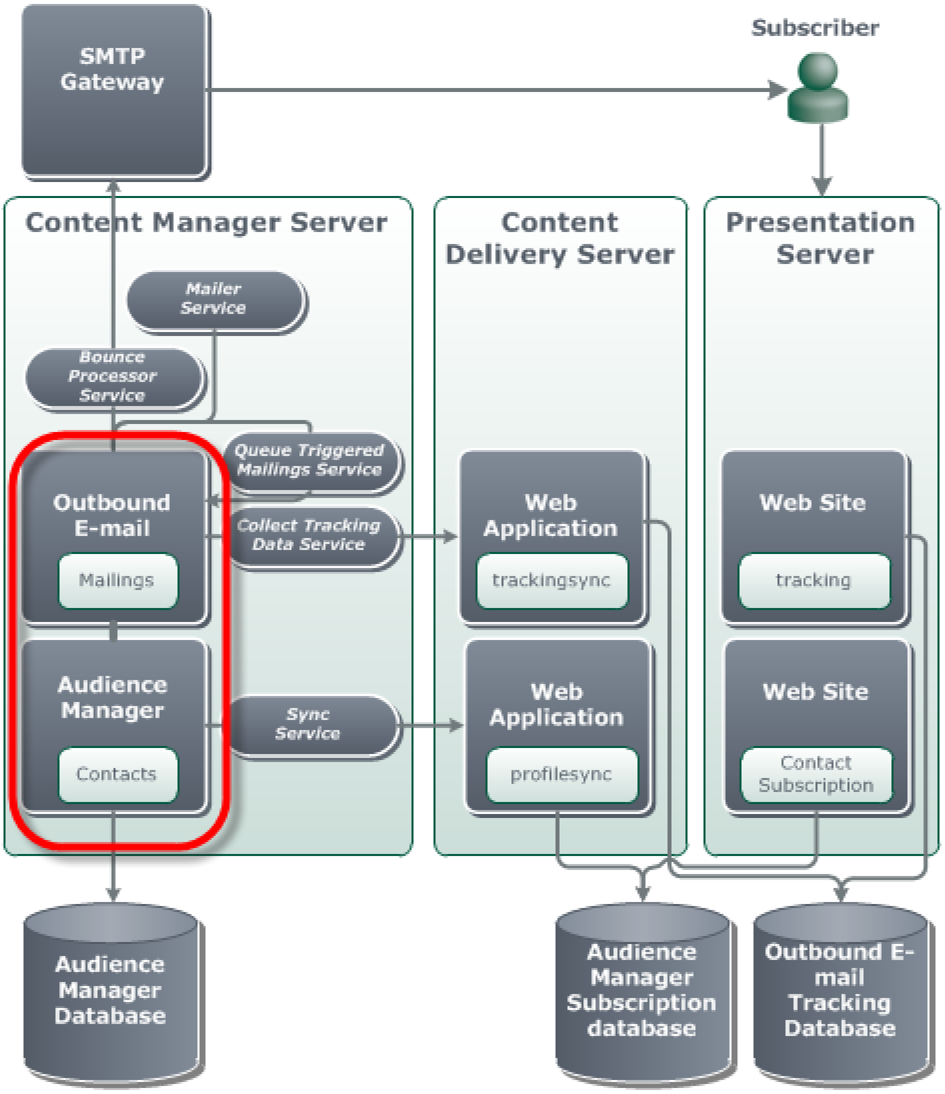
Outbound E-mail requires POP3 or Internet Message Access Protocol (IMAP) server for receiving bounced messages and other e-mails Outbound E-mail failed to deliver. For installation instructions, see the third-party documentation for the specific server vendor.

The **Bounce Processor service** needs POP3 connection to the POP3 server (TCP 110)

You configure bounce processing in the `OutboundEmail.xml`. For more information, see Configuring bounce processing (see page 177).

12.6 Configuring Outbound E-mail

This section describes how to configure Outbound E-mail.



12.6.1 Configuring outgoing e-mail

The <Mail> section of the configuration file OutboundEmail.xml allows you to handle outgoing e-mails.

Steps to execute

1. Open OutboundEmail.xml located by default in the config subdirectory of your Content Manager root location in a plain-text editor or XML editor.

2. Locate the <Mail> section, for example:

```
<Mail>
    <QueueFolder>C:\inetpub\mailroot\Pickup</QueueFolder>
    <MaxEmailsInQueueFolder alternativeFolder="">10000</MaxEmailsInQueueFolder>
    <BatchSize>1000</BatchSize>
    <PollingInterval>15</PollingInterval>
    <TriggeringInterval>15</TriggeringInterval>
    <TestEmailPriority>20</TestEmailPriority>
    <ConfirmationEmailPriority>30</ConfirmationEmailPriority>
    <MaxEmailsPerPeriod>0</MaxEmailsPerPeriod>
    <LimitingPeriod>30</LimitingPeriod>
    <ArchivePersonalizedEmails enforce="false">false</ArchivePersonalizedEmails>
    <RequireAllFieldsDefault>false</RequireAllFieldsDefault>
    <FixedEncoding></FixedEncoding>
    <FixedEmailAddress></FixedEmailAddress>
    <DefaultSenderAddress></DefaultSenderAddress>
</Mail>
```

3. You need to specify the following setting to get outgoing mail up and running:

   `<QueueFolder>`

   The folder where Outbound E-mail queues e-mails. You need to configure your SMTP Server to fetch e-mails from this folder.

4. Specify the following configuration settings that influence performance and handle the size of mailings:

   `<BatchSize>`

   `<BatchSize>` defines the number of e-mails the Mailer Service picks up each round trip. This setting influences the memory used by the Mailer Service; if the `<BatchSize>` is set too high in relation to the size of the e-mails, you may experience out-of-memory errors.

   `<PollingInterval>`

   The time in seconds the Mailer Service waits, after a polling action has completed, before polling the Mailing Queue for e-mails to send again.

   For example, if 20,000 e-mails are in the queue, the `BatchSize` is 1000 and `PollingInterval` is 30 seconds, the Mailer Service first performs 20 iterations of sending out e-mails and only then start polling every 30 seconds.

   `<MaxEmailsInQueueFolder>`

   To prevent sending more mails than your SMTP server can handle at any given time, you can set a maximum number of e-mails for the folder where Outbound E-mail queues e-mails. If you set a maximum, Outbound E-mail adds e-mails to the folder until the max is reached and will only add e-mails to the folder as the SMTP server starts emptying the folder.

   Set the `<MaxEmailsInQueueFolder alternativeFolder="">` attribute if, for example, your SMTP server moves e-mails to another work folder for processing (note that the default is defined by the `<QueueFolder>` element).

   Note: When you send a Mailing, you can choose to send the-mails in batches to control when you want Contacts to receive e-mails. For example, you can send Mailings at a specific time such as 02.00 AM and send them in batches of 10,000 e-mails per day. You need to set a batch size in relation to the number of e-mails in the Mailing; setting a batch size that is too small in relation to the overall number of e-mails degrades performance significantly. For example, if you send 1000 e-mails and you send them with a batch size of 10, you are actually queuing 100 batches each containing 10 e-mails and it is the queuing process rather than the sending which affects performance (queuing 10 e-mails or a 1000 e-mails takes roughly the same amount of time). Therefore, sending a Mailing in batches of 100 batches takes 100 times longer than sending 1 batch.

5. You can also specify the following settings:

   `<ArchivePersonalizedEmails>`

   The default for the **Archive personalized e-mail** check box when you create a new Mailing:

   - `true` selects the check box
   - `false` deselects the check box.

   Set the `enforce` attribute to `false` to allow users to select or deselect the check box, or `true` to prohibit changing the default.

   `<ConfirmationEmailPriority>`

   The priority of notification e-mails, such as confirming subscription (these should have the highest priority). The higher the number, the higher the priority.

   `<FixedEmailAddress>`

   For testing or demo purposes, you can send all outgoing e-mails to a single e-mail address instead of the Contact's e-mail address.

   `<DefaultSenderAddress>`

   The default value added to the **E-mail sender address** field when a user creates a new Mailing.

   `<FixedEncoding>`

   The encoding used in e-mails (optional). You can use `Unicode`, `iso-8859-*`, `windows-*`, `utf-7`, `utf-8`, `utf-16`, `utf-32`, `Shift_JIS`, `gb2312`, `ks_c_5601-1987`, `big5`, `iso-2022-*`, `euc-jp`, `euc-kr`, `x-mac-*`, `asmo-708`, `ibm*`, `dos-*`, `koi8-*`.

   For the complete list of supported charsets, refer to the *Chilkat C# Charset Class Reference* documentation.

   `<LimitingPeriod>`

   A period of time during which a Contact may receive no more than a set number of e-mails as specified by the `<MaxEmailsPerPeriod>` setting.

   `<MaxEmailsPerPeriod>`

   The maximum number of e-mails a Contact may receive within a given period of time as specified by the `<LimitingPeriod>`setting. 0 indicates unlimited.

   `<RequireAllFieldsDefault>`

   The default for the Send Mailing and Send all Mailings **Only send e-mail to Contacts with all required merge data** check box. `true` selects the check box by default, `false` deselects it.

   `<TriggeringInterval>`

   Fill in the number of seconds you want Outbound E-mail to wait before checking if new Contacts have joined a Distribution List that triggers a Mailing.

   `<TestEmailPriority>`

The priority of e-mails sent out for testing purposes (these should have a higher priority than regular Mailings). The higher the number, the higher the priority.

6. Save and close `OutboundEmail.xml`.

12.6.2 Configuring bounce processing

When e-mails do not reach their intended recipients the e-mail is bounced back to you (or a designated bounce e-mail address). Process bounced e-mails to prevent having your server blacklisted and not being able to send any more e-mails.

Requirements

The **E-mail error status** of a Contact indicates whether problems have been encountered sending e-mails to the Contact. The status is only updated when you have tracked links in your Mailing. You can track links by setting the **Track Links** parameter in the **Outbound E-mail Post-Processing** Template Building Block, or by using the `GetLinkTrackingURL()` method in your Dreamweaver Templates.

Context

The `<BounceProcessing>` section of the configuration file `OutboundEmail.xml` contains one or more `<Mailbox>` elements that allow you to specify multiple e-mail inboxes to retrieve bounce messages from (the Bounce Processor service loops through each of them). Each `<Mailbox>` element contains a `<Forwarding>` element in which you specify how incoming messages sent as a response to sent mailings, such as bounce messages and other responses, are handled.

Steps to execute

1. Open `OutboundEmail.xml` located by default in the `config` subdirectory of your Content Manager root location in a plain-text editor or XML editor.

2. Locate the `<BounceProcessing>` section.

3. If the section is encrypted:

   a. Open a command prompt.

   b. Navigate to the folder `Outbound E-Mail\Encryption` on the SDL Tridion installation media and run the following command `UnprotectConfiguration.cmd`:

      ```
      UnprotectConfiguration.cmd
      ```

   c. Reload `OutboundEmail.xml` to view the unencrypted `<BounceProcessing>` section:

```
<BounceProcessing>
    <PollingInterval>300</PollingInterval>
    <Mailbox name="MailBounce">
```

```
      <Protocol>POP3</Protocol>
      <Server>localhost</Server>
      <Port>110</Port>
      <UseSSL>false</UseSSL>
      <User>MailBounce</User>
      <Password>password</Password>
      <ImapFolder></ImapFolder>
      <Forwarding>
        <SenderName>Forwarder</SenderName>
        <SenderAddress>forwarder@localhost</SenderAddress>
        <BouncedEmails>
          <Subject></Subject>
          <EmailAddress>bounced@localhost</EmailAddress>
          <Message></Message>
        </BouncedEmails>
        <RepliesFromContacts>
          <Subject></Subject>
          <EmailAddress>replies@localhost</EmailAddress>
          <Message></Message>
        </RepliesFromContacts>
      </Forwarding>
  </BounceProcessing>
```

4. Define a `<PollingInterval>`:

Element	Description
`<PollingInterval>`	The time in seconds the Outbound E-mail Bounc waits, after a polling action has completed, befo available inboxes for bounced messages again. minutes).

5. Define a `<Mailbox>` element as follows:

   The `<BounceProcessing>` section can have one or more `<Mailbox>` elements defined. Each mailbox is identified by a unique `name` attribute.

Element	Description
`<Protocol>`	The protocol used to receive e-mails: `POP3` or `IM`
`<Server>`	The DNS name of the POP3 or IMAP mail server, `pop3.sdltridion.com`.
`<Port>`	The port of the POP3 or IMAP mail server: • **POP3**—by default POP3 uses port `110` or, to port `995` • **IMAP**—by default IMAP uses port `143` or, to port `993`
`<UseSSL>`	Set to `true` to instruct Chilkat to create an SSL c mail server, otherwise `false`. If you do not specify `<UseSSL>`, Chilkat will use i you set the `<Port>` value to the default SSL port for IMAP), Chilkat will connect over SSL.
`<User>`	The name of the mailbox account where bounce expected to be sent.
`<Password>`	The password of the `<User>` mailbox. For more i Encrypting sensitive strings for Audience Manag
`<Imap>`	For an IMAP Server, the name of the incoming m default `inbox`.

6. Define `<Forwarding>` for a `<Mailbox>` element as follows:

Important:All e-mails are deleted from your POP3 or IMAP inbox once they have been analyzed; if you do not set up forwarding, you will lose these e-mails permanently. This is a particular problem if you also do not set a **E-mail reply address** on your Mailing as you will then lose actual e-mails from Contacts.

Element	Description
`<SenderName>`	The sender name as displayed in the **From:** fiel
`<SenderAddress>`	The displayed e-mail address of the sender of th

In the `<BouncedEmails>` section, specify how to handle bounced e-mails:

Element in `<BouncedEmails>`	Description
`<EmailAddress>`	Bounced messages are sent to this e-mail addre
`<Subject>`	The subject of a bounced e-mail. Filling in a subj subject of the original, bounced e-mail message that subject, leave this field empty.
`<Message>`	The message body of a forwarded e-mail. The bo mail is appended under this message body.

In the `<RepliesFromContacts>` section, specify how to handle e-mails sent by a Contact:

Element in `<RepliesFromContacts>`	Description
`<EmailAddress>`	Forwarded messages that are not recognized as are presumed to be e-mails send by the contact mail address.
`<Subject>`	The subject of a forwarded e-mail. Fill in a subje the subject of the original e-mail message, other retain the original subject.
`<Message>`	The message body of a forwarded e-mail. The bo mail is appended under this message body.

7. Save and close `OutboundEmail.xml`.

12.6.3 Configuring import

The `<Import>` section of the configuration file `OutboundEmail.xml` allows you to define the location for uploading CSV import files. If you are uploading a lot of Contacts or large files, you also need to configure the upload limit of the Content Manager Explorer Web site.

Context

Importing through the API

When you import Contacts and Keywords through the API, the import mechanism uploads the CSV file to the `<folder>` specified in the `<Import>` section of the `OutboundEmail.xml` configuration file. The default value is set to the Tridion Upload directory (see below) as this ensures that the folder exists and that temporary unique file names are created for the uploaded files.

Importing through the Content Manager Explorer

When you import Contacts and Keywords through the Content Manager Explorer, Outbound E-mail uploads the CSV file to the Tridion Upload directory specified in the Content Manager Explorer configuration file. The file is located in your Tridion installation `web\WebUI\WebRoot\Configuration` directory. The default value for the Tridion Upload directory on Windows 2008 is:

```
<Tridion.UploadDirectory>C:\ProgramData\SDL\Upload\</Tridion.UploadDirectory>
```

Steps to execute

1. Open `OutboundEmail.xml` located by default in the `config` subdirectory of your Content Manager root location in a plain-text editor or XML editor.

2. Locate the `<Import>` section:

```
<Import>
    <Folder>C:\Program Files
    (x86)\Tridion\OutboundEmail\Bin\Import files</Folder>
</Import>
```

3. Specify the following setting:

Element	Description
`Folder`	The location on the Content Manager Server import files are uploaded to.

4. Save and close `OutboundEmail.xml`.

5. In IIS 7, by default the Content Manager Web site allows 30 MB of data to be uploaded. To change this setting, modify the `web.config` in the `web\WebUI\WebRoot` subdirectory of the Content Manager root location (defaults to
`C:\Program Files (x86)\Tridion\`)

6. In the `<system.webServer>` section, set the maximum allowed content length in bytes:

```
<system.webServer>
    <security>
      <requestFiltering>
        <requestLimits maxAllowedContentLength="1073741824" />
      </requestFiltering>
    </security>
  </system.webServer>
```

7. In the `<system.web>` section, set the maximum request length in bytes:

```
<httpRuntime maxRequestLength="1073741824"/>
```

Next steps

You can configure the `<ImportTimeout>` in the `<Database>` section of the `OutboundEmail.xml` file.

12.6.4 Creating a Publication Target

A Mailing requires a Publication Target that is used when rendering the Mailing. A Mailing is rendered when you test it; when you send the Mailing it uses the rendered content the last time the Mailing was tested.

Steps to execute

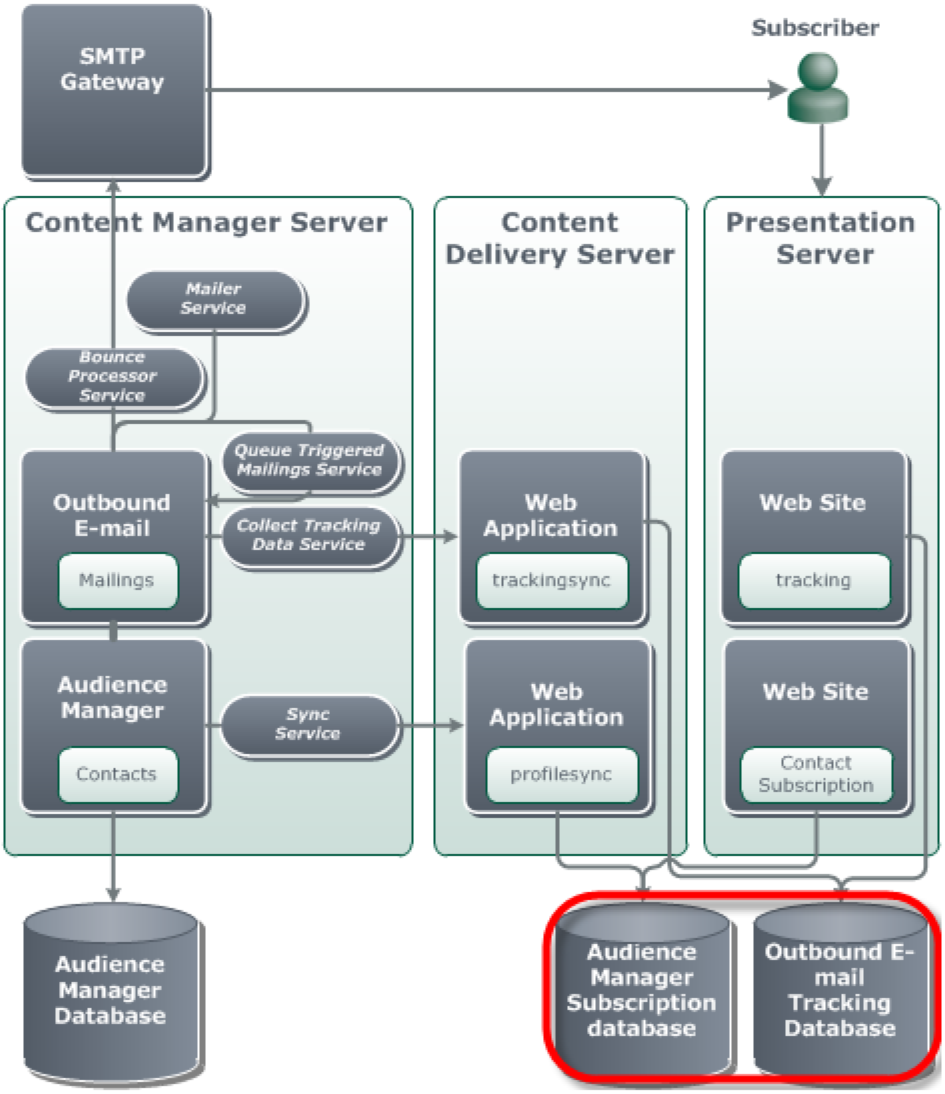
1. Open the Content Manager Explorer.
2. In the **System Administration** view, navigate to the **Publishing Management** node.
3. Select the **Publishing Management** node and choose **New Publication Target** from the context menu.
4. In the **New Publication Target** window, do the following:
   - Set **Name** to a name of your choosing.
   - Set **Description** to a description of your choosing.
   - Leave the **Target Language**, **Default Code Page**, **Minimal Approval Status** and **Priority** fields as they are.
   - Under **Destinations**, click **Add** and from the **Protocol** dropdown that appears on the right hand side, select **Local File System**. Set **Name** and **Location** to any value you like.
   - Click **Save and Close** to save and close the Publication Target.

Result

You have created a Publication Target which is used for rendering Mailings (Outbound E-mail users need to specify this value in a Mailing).

12.7 Creating Tracking and Subscription databases

Outbound E-mail requires you to create Tracking database to store tracking data, and Audience Manager requires you to create a Subscription database to store Contact subscription data.



12.7.1 Creating Tracking and Subscription databases using PowerShell scripts

Use a PowerShell script to create Microsoft SQL Server or Oracle databases for Tracking and Subscription.

Requirements

To install SDL Tridion databases on a Windows machine requires Windows PowerShell 3.0 and Microsoft .NET Framework 4.0 or higher.

Context

Note: You can install Subscription and Tracking on one or several presentation servers, and these can use the same or different Tracking and Subscription databases.

Steps to execute

1. Open Windows PowerShell from the Windows Start Menu.
2. In PowerShell, depending on your database navigate to one of the following folders on the installation media:
   - `Database\MSSQL\` or
   - `Database\Oracle\`
3. Type the following command to install a Tracking database:

   ```
   & '.\Install Outbound E-mail Tracking database.ps1'
   ```

4. Type the following command to install a Subscription database:

   ```
   & '.\Install Outbound E-mail Subscription Management database.ps1'
   ```

5. Follow the instructions in the PowerShell console to create the database. Repeat this procedure as needed.

Result

You have created Tracking and Subscription databases.

Next steps

When you create the Subscription and Tracking databases, you will need to retain the database details when you come to installing and configuring tracking and subscription on the presentation server.

12.7.2 Configuring the Tracking and Subscription databases

The `cd_storage_conf.xml` file is the Content Delivery Storage Layer configuration file. You need to configure your Tracking and Subscription databases in this file.

Steps to execute

1. Open the `cd_storage_conf.xml` configuration file.
2. In the `<Storages>` element, add the following `<StorageBindings>`:

   ```
   <StorageBindings>
      <Bundle src="AudienceManagerDAOBundle.xml"/>
   </StorageBindings>
   ```

   Note: The `AudienceManagerDAOBundle.xml` is embedded in the `oe_utils.jar` file.

3. In the `<ItemTypes>` element, add the following `<Item>`s:

   ```
   <ItemTypes defaultStorageId="defaultFile">
      <Item typeMapping="AudienceManagerProfile" storageId="profiledb"/>
      <Item typeMapping="AudienceManagerTracking" storageId="trackingdb"/>
   </ItemTypes>
   ```

Where `storageId` indicates the Subscription or Tracking database where data is stored.

4. In the `<Storages>` element, configure the **Outbound E-mail Tracking database**:

   - For Microsoft SQL database:

```
<Storage Id="trackingdbmssql" Class="com.tridion.storage.persistence.JPADAOFactory" Type="persistence"
 dialect="MSSQL">
   <Pool CheckoutTimeout="120" IdleTimeout="120" MonitorInterval="60" Size="5" Type="jdbc"/>
   <DataSource Class="com.microsoft.sqlserver.jdbc.SQLServerDataSource">
      <Property Name="serverName" Value="localhost"/>
      <Property Name="portNumber" Value="1433"/>
      <Property Name="databaseName" Value="tridion_tracking"/>
      <Property Name="user" Value="TMSPSUSER"/>
      <Property Name="password" Value="PASSWORD"/>
    </DataSource>
 </Storage>
```

   - For Oracle database:

```
<Storage Id="trackingdbOracle" Class="com.tridion.storage.persistence.JPADAOFactory" Type="persistence"
 dialect="ORACLESQL">
     <Pool CheckoutTimeout="120" IdleTimeout="120" MonitorInterval="60" Size="5" Type="jdbc" />
     <DataSource Class="oracle.jdbc.pool.OracleDataSource">
      <Property Name="driverType" Value="thin" />
      <Property Name="networkProtocol" Value="tcp" />
      <Property Name="serverName" Value="servername" />
      <Property Name="portNumber" Value="1521" />
      <Property Name="databaseName" Value="databasename" />
      <Property Name="user" Value="TMSPSUSER" />
      <Property Name="password" Value="PASSWORD" />
     </DataSource>
 </Storage>
```

5. In the `<Storages>` element, configure the **Audience Manager Subscription database**:

   - For Microsoft SQL database:

```
<Storage Id="profiledbmssql" Class="com.tridion.storage.persistence.JPADAOFactory" Type="persistence"
 dialect="MSSQL">
    <Pool CheckoutTimeout="120" IdleTimeout="120" MonitorInterval="60" Size="5" Type="jdbc"/>
    <DataSource Class="com.microsoft.sqlserver.jdbc.SQLServerDataSource">
        <Property Name="serverName" Value="localhost"/>
        <Property Name="portNumber" Value="1433"/>
        <Property Name="databaseName" Value="tridion_submgmt"/>
        <Property Name="user" Value="TMSSMUSER"/>
        <Property Name="password" Value="PASSWORD"/>
    </DataSource>
</Storage>
```

   - For Oracle database:

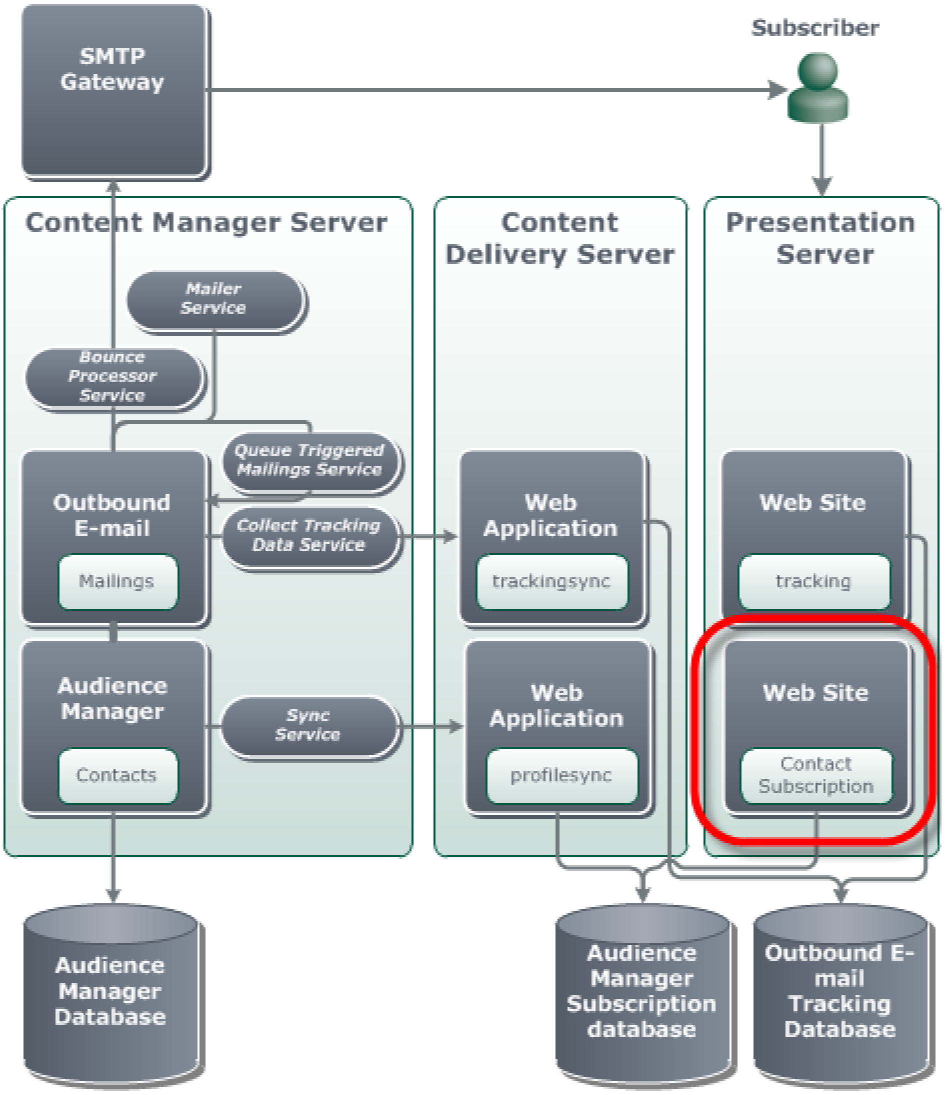
```
<Storage Id="profiledbOracle" Class="com.tridion.storage.persistence.JPADAOFactory" Type="persistence"
 dialect="ORACLESQL">
    <Pool CheckoutTimeout="120" IdleTimeout="120" MonitorInterval="60" Size="5" Type="jdbc" />
     <DataSource Class="oracle.jdbc.pool.OracleDataSource">
      <Property Name="driverType" Value="thin" />
      <Property Name="networkProtocol" Value="tcp" />
      <Property Name="serverName" Value="servername" />
      <Property Name="portNumber" Value="1521" />
      <Property Name="databaseName" Value="databasename" />
      <Property Name="user" Value="TMSSMUSER" />
      <Property Name="password" Value="PASSWORD" />
     </DataSource>
</Storage>
```

Next steps

For information on `cd_storage_conf.xml` and how to encrypt passwords, refer to Configuring content storage (see page 109) and Encrypting sensitive strings (see page 116).

12.8 Installing Contact Subscription on Presentation Server

The Presentation Server is the application that accepts subscriptions and changes to subscriptions. This section describes the installation and configuration of Audience Manager presentation side components used to implement Subscription functionality.



Note: If you want to install Presentation Side components on a 64-bit operating system, but you are using a 32-bit Java Virtual Machine, your Web site must be running in a 32-bit process (Application Pool).

12.8.1 Installing Contact subscription (ASP.NET)

You must install Contact subscription components on a publicly accessible Web site. This task describes how to install Contact subscription components on Internet Information Service (IIS).

Requirements

- You must have created a Subscription database.
- You must install Contact subscription on a publicly accessible Web site.

Context

Audience Manager provides sample Web pages that implement Subscription management.

Steps to execute

1. Open Internet Information Services (IIS) Manager.
2. Go to your existing public-facing Web site.
3. Install and configure the Content Delivery API Server Role:

   a.

   b. Installing the API Server Role as a .NET Web application (see page 85)

4. Access the SDL Tridion installation media and navigate to the `Content Delivery\roles\` folder.
5. Install DLLs for Contact Subscription: navigate to `Outbound E-Mail\Presentation System\NET\Samples\bin` and copy the DLLs to the `\bin` folder of your Web site.
6. Install ASPXs for Contact Subscription: navigate to `Outbound E-Mail\Presentation System\NET\Samples` and copy the ASPXs (and default `Web.config`, if necessary) to a directory in your Web site.

   Note: You need to specify these pages in the **Audience Management** tab of a **Publication**.

7. Install JARs: navigate to `Outbound E-Mail\Presentation System\NET\lib` and copy the JAR files to your Web site `\lib` directory.
8. Download the JDBC driver (JAR file) required for your specific database vendor and Java version, and copy it into the `lib` folder:

Database vendor	JDBC Driver	File to download
Microsoft SQL Server	Microsoft JDBC Driver 4.0 for SQL Server	`sqljdbc4.jar`

Database vendor	JDBC Driver	File to download
Oracle	Oracle JDBC driver 11.2.0.3	`ojdbc6.jar`

9. Configure the following files:

   a. `cd_storage_conf.xml`—see Configuring the Tracking and Subscription databases (see page 183)

   b. `cd_audience_manager.conf.xml`—see:

      - Configuring Outbound E-mail on Presentation Server (see page 207)
      - Configuring Audience Manager on Presentation Server (see page 192)

   c. `logback.xml` (configure logging).

10. Add the following settings to your `Web.config`:

```
<configuration>
 <appSettings>
  <add key="Subscribe.NotificationPageId" value="tcm:0-0-0"/>
  <add key="Confirm.NotificationPageId" value="tcm:0-0-0"/>
  <add key="ChangeProfile.NotificationPageId" value="tcm:0-0-0"/>
  <add key="Unsubscribe.NotificationPageId" value="tcm:0-0-0"/>
  <add key="KeywordNewsletter" value="tcm:6-518-1024"/>
  <add key="KeywordOffers" value="tcm:6-519-1024"/>
  <add key="ConfigurationFile" value="cd_audience_manager_conf.xml"/>
 </appSettings>
```

Result

You have installed the Subscription Web pages.

12.8.2 Installing Contact subscription (Java)

You must install Contact subscription components on a publicly accessible Web site. This task describes how to install Contact subscription components on a Java-based Web application Server.

Requirements

- You must have created a Subscription database.
- You must install Contact subscription on a publicly accessible Web site.

Context

Audience Manager provides sample Web pages that implement Subscription management.

Steps to execute

1. Go to your existing public-facing Web site.

2. Install and configure the Content Delivery API Server Role:

a.

b. Installing the API Server Role for a Java Web application (see page 104)

3. Access the SDL Tridion installation media and navigate to the `Content Delivery\roles\` folder.

4. Install JARs for Contact Subscription and Contact Synchronization: navigate to `Outbound E-Mail\Presentation System\Java \Subscription` and copy the JAR files to your Web site `\lib` directory.

5. Install JSPs for Contact Subscription: navigate to `Outbound E-Mail \Presentation System\Java\Subscription\Web` and copy the JSPs to a directory in your Web site.

Note: You need to specify these pages in the **Audience Management** tab of a **Publication**.

6. Download the JDBC driver (JAR file) required for your specific database vendor and Java version, and copy it into the `lib` folder:

Database vendor	JDBC Driver	File to download
Microsoft SQL Server	Microsoft JDBC Driver 4.0 for SQL Server	`sqljdbc4.jar`
Oracle	Oracle JDBC driver 11.2.0.3	`ojdbc6.jar`

7. Configure the following files:

a. `cd_storage_conf.xml`—see Configuring the Tracking and Subscription databases (see page 183)

b. `cd_audience_manager.conf.xml`—see:

- Configuring Outbound E-mail on Presentation Server (see page 207)
- Configuring Audience Manager on Presentation Server (see page 192)

c. `logback.xml` (configure logging).

Result

You have installed the Subscription Web pages.

12.8.3 Configuring Publications for Contact Subscription

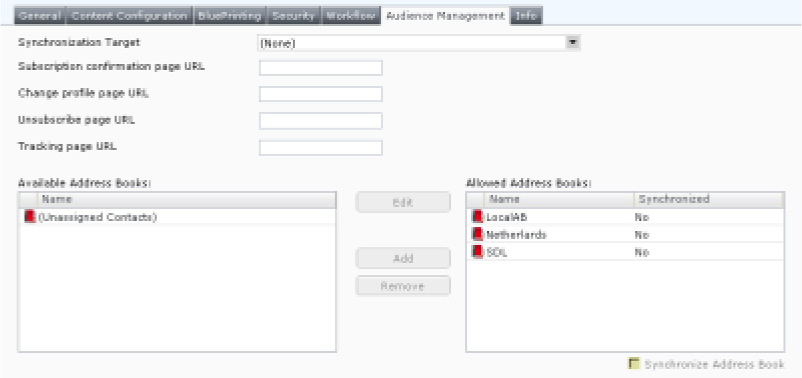
In a Publication **Audience Management** tab, you can specify the Synchronization Target used to synchronize data between the Content Manager and Presentation Server and the URLs of Contact Subscription pages and the Tracking page.

Requirements

To configure Publications for tracking and synchronization, you must have Tridion System Administrator rights or Publication Management rights.

Steps to execute

1. Open the Content Manager Explorer.
2. Select a Publication and choose **Properties** from the context menu.
3. Select the **Audience Management** tab.
4. Fill in the following fields:
   - **Subscription confirmation page URL**—the URL of the Web page in which a new subscriber confirms their subscription.
   - **Change profile page URL**—the URL of the Web page in which a subscriber can change their personal information.
   - **Unsubscribe page URL**—the URL of the Web page in which a subscriber can cancel their subscription.
   - **Tracking page URL**—the URL of the Web page that tracks mailing response, for example whether a email recipient opened a Mailing and what links were clicked. For more information, see Configuring Publications for Contact Subscription (see page 189).



Synchronization Target (None)

Subscription confirmation page URL

Change profile page URL

Unsubscribe page URL

Tracking page URL

Available Address Books:

Name
(Unassigned Contacts)

Allowed Address Books:

Name	Synchronized
LocalAB	No
Netherlands	No
SDL	No

□ Synchronize Address Book

5. Select an Address Book from the **Available Address Books** and click **Add** to add it to the **Allowed Address Books**. For each Address Book in the **Available Address Books** list you can choose to synchronize these Contacts or not by selecting the **Synchronize Address Book** check box.

Note: For performance reasons, you cannot synchronize Dynamic Address Books.

6. Click **Save and Close** to save and close the Publication.

Result

You have configured an E-mail Publication for tracking and synchronization.

12.8.4 Configuring confirmation e-mail defaults

In the `<PresentationSide>` section of the configuration file `OutboundEmail.xml`, configure `<ConfirmationEmailDefaults>` sections to specify the default e-mails sent to subscribers when they confirm, subscribe, change their profile, or unsubscribe from a Mailing. The defaults are used if no `<ConfirmationEmail>` elements are defined.

Steps to execute

1. Open `OutboundEmail.xml` located by default in the `config` subdirectory of your Content Manager root location in a plain-text editor or XML editor.

2. Locate the `<ConfirmationEmailDefaults>` section, for example:

```
<PresentationSide>
    ...
<ConfirmationEmailDefaults type="Subscribe">
      <Subject>Please confirm your subscription</Subject>
      <SenderName>Sender</SenderName>
      <SenderAddress>sender@localhost</SenderAddress>
      <ReplyAddress>sender@localhost</ReplyAddress>
      <RequireAllFields>false</RequireAllFields>
    </ConfirmationEmailDefaults>
    <ConfirmationEmailDefaults type="OptedIn">
      <Subject>Your subscription has been confirmed</Subject>
      <SenderName>Sender</SenderName>
      <SenderAddress>sender@localhost</SenderAddress>
      <ReplyAddress>sender@localhost</ReplyAddress>
      <RequireAllFields>false</RequireAllFields>
    </ConfirmationEmailDefaults>
    <ConfirmationEmailDefaults type="ChangeProfile">
      <Subject>Your profile has been updated</Subject>
      <SenderName>Sender</SenderName>
      <SenderAddress>sender@localhost</SenderAddress>
      <ReplyAddress>sender@localhost</ReplyAddress>
      <RequireAllFields>false</RequireAllFields>
    </ConfirmationEmailDefaults>
    <ConfirmationEmailDefaults type="Unsubscribe">
      <Subject>You have been unsubscribed</Subject>
      <SenderName>Sender</SenderName>
      <SenderAddress>sender@localhost</SenderAddress>
      <ReplyAddress>sender@localhost</ReplyAddress>
      <RequireAllFields>false</RequireAllFields>
    </ConfirmationEmailDefaults>
    ...
  </PresentationSide>
```

3. Specify a value for the `type` attribute for the `<ConfirmationEmailDefaults>` section from one of the following:

   - `Subscribe`
   - `OptedIn`
   - `ChangeProfile`
   - `Unsubscribe`

Specify the following settings:

Element	Description
`Subject`	The subject title of the e-mail message sent when the user has confirmed, subscribed, changed their profile, or unsubscribed from a Mailing.
`SenderName`	The name of the sender as you want it to appear in the confirmation e-mail message.
`SenderAddress`	The e-mail address of the sender as you want it to appear in the confirmation e-mail message.
`ReplyAddress`	The reply-to e-mail address as you want it to appear in the confirmation e-mail message.
`RequireAllFields`	Set to `true` to if you want the e-mail to be sent only if all Merge Fields can be resolved.

4. Save and close `OutboundEmail.xml`.

12.8.5 Configuring confirmation e-mails

In the `<PresentationSide>` section of the configuration file `OutboundEmail.xml`, add `<ConfirmationEmail>` elements to specify the e-mails sent to subscribers when they confirm, subscribe, change their profile, or unsubscribe from a Mailing.

Context

When a user confirms, subscribes, changes their profile, or unsubscribes via the Web site, the Web page they submit contains the Content Manager `TCM:URI` of the Page which is used in the e-mail to notify the user. Outbound E-mail checks the `<ConfirmationEmail>` sections and identifies the appropriate configuration using the `TCM:URI` specified in the page attribute. If the `TCM:URI` is not found, Outbound E-mail detects the type of the submitted Web page and uses the defaults in the `<ConfirmationEmailDefaults>` section.

Steps to execute

1. Open `OutboundEmail.xml` located by default in the `config` subdirectory of your Content Manager root location in a plain-text editor or XML editor.

2. Locate the `<ConfirmationEmails>` section, for example:

```
<PresentationSide>
  ...
  <PublicationTargetUri>tcm:0-4-65537</PublicationTargetUri>
  <ConfirmationEmails>
```

```
   <ConfirmationEmail page="tcm:11-227-64">
    <Subject>Please confirm your subscription</Subject>
    <SenderName>Sender</SenderName>
    <SenderAddress>sender@localhost</SenderAddress>
    <ReplyAddress>sender@localhost</ReplyAddress>
    <RequireAllFields>false</RequireAllFields>
   </ConfirmationEmail>
    </ConfirmationEmails>
  </PresentationSide>
```

3. In a `<ConfirmationEmail>` section, specify a page attribute whose value is the `TCM:URI` of the E-mail Page used in the e-mail response.

   Specify the following settings (if one or more elements are unspecified, the defaults are used in the `<ConfirmationEmailDefaults>` section):

Element	Description
`Subject`	The subject title of the e-mail message sent when the user has confirmed, subscribed, changed their profile, or unsubscribed from a Mailing.
`SenderName`	The name of the sender as you want it to appear in the confirmation e-mail message.
`SenderAddress`	The e-mail address of the sender as you want it to appear in the confirmation e-mail message.
`ReplyAddress`	The reply-to e-mail address as you want it to appear in the confirmation e-mail message.
`RequireAllFields`	Set to `true` to if you want the e-mail to be sent only if all Merge Fields can be resolved.

4. Specify the `<PublicationTargetUri>` setting:

```
<PublicationTargetUri>tcm:0-4-65537</PublicationTargetUri>
```

Element	Description
`PublicationTargetUri`	The Publication Target used to render confirmation e-mail messages.

5. Save and close `OutboundEmail.xml`.

12.8.6 Configuring Audience Manager on Presentation Server

The `cd_audience_manager.conf.xml` is the configuration file for Audience Manager and Outbound E-mail on the Presentation Server.

Steps to execute

1. On a .NET system, navigate to `Outbound E-Mail\Presentation System\NET\config` and copy the following file to your Tridion installation `\config` directory:
   - `cd_audience_manager.conf.xml`
2. On a Java system, navigate to `Outbound E-Mail\Presentation System\Java\config` and copy the following file to your Tridion installation `\classes` directory:
   - `cd_audience_manager.conf.xml`
3. Open `cd_audience_manager.conf.xml` in a text editor:
4. For Audience Manager, you need to configure your `<ExtendedDetails>`:

   `<EmailAddressFieldName>`

   The name of the extended details field that contains the e-mail address.

   `<ContactIdentification><Field>`

   By default, the extended detail fields `IDENTIFICATION_KEY` and `IDENTIFICATION_SOURCE` are used to uniquely identify Contacts.

   If you use different Contact extended detail fields for your unique index, configure `<Field>` elements accordingly.

   ```
   <Configuration>
     ...
     <ExtendedDetails>
        <EmailAddressFieldName>MAIL</EmailAddressFieldName>
           <ContactIdentification>
               <Field>IDENTIFICATION_KEY</Field>
               <Field>IDENTIFICATION_SOURCE</Field>
           </ContactIdentification>
     </ExtendedDetails>
   </Configuration>
   ```

   Note: If you are running multiple instances of Content Delivery on your ASP.NET Web site, you need to add the following code to your ASPX pages to use separate configuration files for each Web site (on a single line):

   ```
   Tridion.OutboundEmail.ContentDelivery.Utilities.
   Settings.ConfigFileName = "NameOfConfig.xml";
   ```

5. Save and close `cd_audience_manager.conf.xml`.

12.8.7 Configuring logging

The `logback.xml` file is the configuration file for the logback logging framework used by Content Delivery. This task describes how to update your existing `logback.xml` file.

Steps to execute

1. Open the file `logback.xml`, located in the `config` subdirectory of your Content Delivery root location, in a plain-text editor or XML editor.

2. In the `<!-- Appenders -->` section, add the following:

```
<appender name="rollingAudiencemanagerLog" class="ch.qos.logback.core.rolling.RollingFileAppender">
       <rollingPolicy class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">
           <fileNamePattern>${log.folder}/sync_audiencemanager.%d{yyyy-MM-dd}.log</fileNamePattern>
           <maxHistory>${log.history}</maxHistory>
       </rollingPolicy>
       <encoder>
           <pattern>${log.pattern}</pattern>
       </encoder>
       <prudent>true</prudent>
</appender>
```

3. In the `<!-- Loggers -->` section, add the following:

```
<logger name="com.tridion.marketingsolution">
   <appender-ref ref="rollingAudiencemanagerLog"/>
</logger>
```

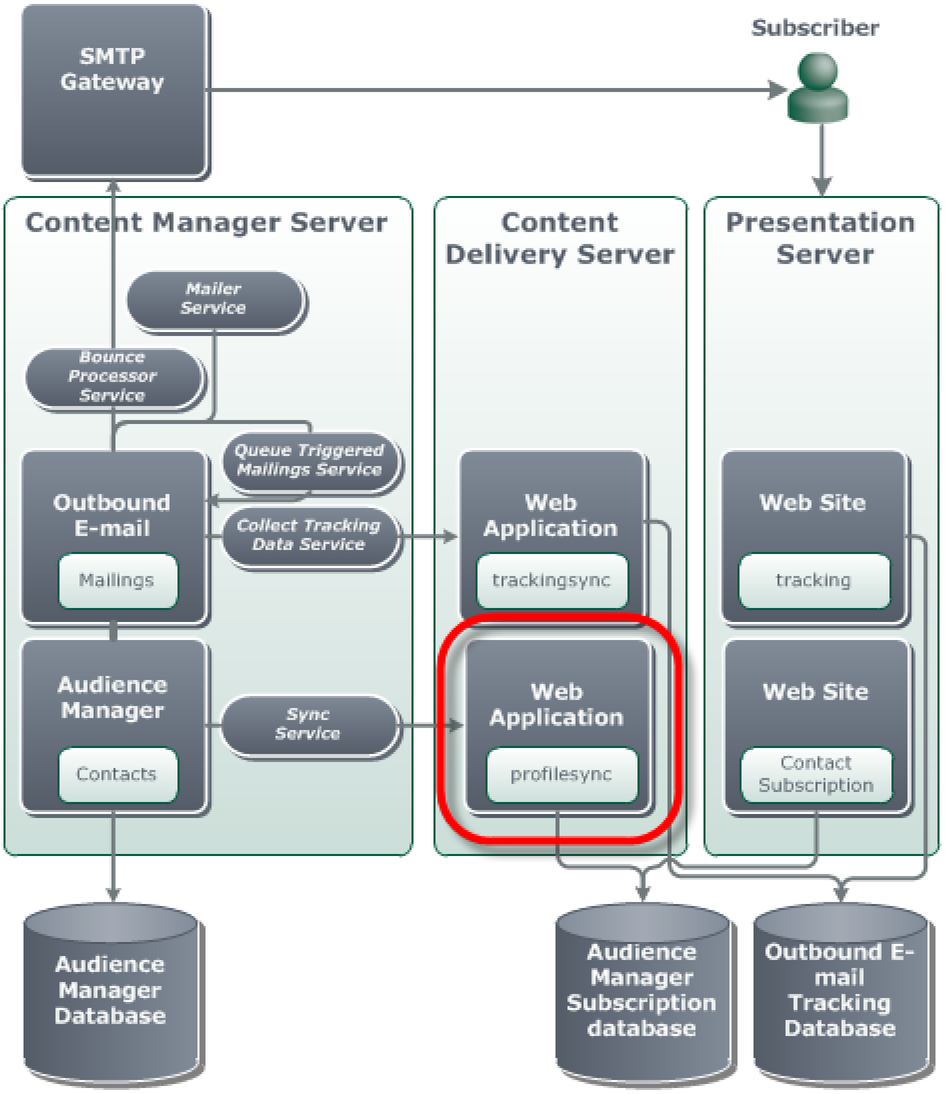
4. Save and close `logback.xml`.

Next steps

For information on changing the logging level, the location to log to, the number of log files and the name of log files, see Configuring logging on the Content Delivery server (see page 107).

12.9 Setting up a Synchronization Server for Contacts

The Synchronization Server synchronizes Contact data between the Audience Manager database on the Content Manager side and the Subscription database on the presentation side. You install Synchronization components in a .NET or Java Web application.



12.9.1 Installing Contact synchronization (ASP.NET)

The Synchronization Server synchronizes Contact and Segment data between the Content Manager Server and Presentation Server. This task describes how to set up a Synchronization Server for Contact synchronization on Internet Information Service (IIS).

Requirements

It is recommended to install Contact synchronization in a separate Web application that is not publicly available. Refer to Content Delivery prerequisites (see page 67) to learn which Web and Application Servers are supported.

Before installing Contact synchronization:

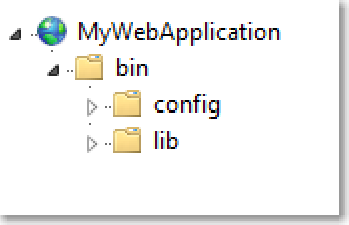
- Install the Content Delivery API Server Role in your Web application (for database access only—do not configure the Ambient Data Framework).
- Create a Subscription database.

Context

Audience Manager provides Web pages for synchronizing Contact and Segment data between the Content Manager Server and Presentation Server.

Steps to execute

1. Open Internet Information Services (IIS) Manager.
2. Create a New Web site:
   a. Select the Web sites and node and choose **New** > **Web site** from the context menu.
   b. Follow the steps in the Web Site Creation Wizard.
3. Create the following Web application directory structure:
   a. In the root folder, create a `bin\` subfolder.
   b. In the `bin\` folder, create subfolders `config\` and `lib\`.



4. In the `bin\` folder, create a subfolder called for example `syncComponent\`.
5. Install DLLs for Contact Synchronization, Tracking and Tracking retrieval: navigate to `Outbound E-Mail\Presentation System\NET\Synchronization\bin` and copy the DLLs to the `\bin` folder of your Web site.
6. Install ASPXs for Contact Synchronization: navigate to `Outbound E-Mail\Presentation System\NET\Synchronization\` and copy `Profilesync.aspx` and `Profilesync.aspx.cs` (and default `Web.config`, if necessary) to a directory in your Web site, for example `syncComponent\`.

   Note: You need to specify `profilesync.aspx` in your Synchronization Target.

7. Install JARs: navigate to `Outbound E-Mail\Presentation System\NET\lib` and copy the JAR files to your Web site `\lib` directory.
8. Download the JDBC driver (JAR file) required for your specific database vendor and Java version, and copy it into the `lib` folder:

Database vendor	JDBC Driver	File to download
Microsoft SQL Server	Microsoft JDBC Driver 4.0 for SQL Server	`sqljdbc4.jar`
Oracle	Oracle JDBC driver 11.2.0.3	`ojdbc6.jar`

9. Configure the following files:

   a. `cd_storage_conf.xml`—see Configuring the Tracking and Subscription databases (see page 183)

   b. `cd_audience_manager.conf.xml`—see:

      - Configuring Outbound E-mail on Presentation Server (see page 207)
      - Configuring Audience Manager on Presentation Server (see page 192)

   c. `logback.xml` (configure logging).

10. Test the installation by opening a browser and running the `profilesync` page.

11. Configure security so that only the Content Manager Server can access this Web application, for example as follows:

    - Create a specific user with limited access, and configure the Web application so that it only allows this user access.
    - Use HTTPS to allow only secure access.
    - Implement IP-based access control.

Result

You have installed the Synchronization Web pages.

12.9.2 Installing Contact synchronization (Java)

The Synchronization Server synchronizes Contact and Segment data between the Content Manager Server and Presentation Server. This task describes how to set up a Synchronization Server for Contact synchronization on a Java-based Web Application Server.

Requirements

- It is recommended to install Contact synchronization in a separate Web application that is not publicly available. Refer to Content Delivery prerequisites (see page 67) to learn which Web and Application Servers are supported.

Before installing Contact synchronization:

- Install the Content Delivery API Server Role in your Web application (for database access only—do not configure the Ambient Data Framework).
- Create a Subscription database.

Context

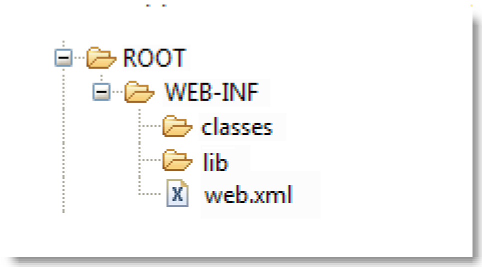
Audience Manager provides Web pages for synchronizing Contact and Segment data between the Content Manager Server and Presentation Server.

Steps to execute

1. Create a new Web application in your Web Application Server.
2. Create the following Web application directory structure:

   a. In the root folder, create a `WEB-INF/` directory.

   b. In the `WEB-INF/` directory, create subdirectories `classes/` and `lib/`.



```
ROOT
  WEB-INF
    classes
    lib
    web.xml
```

3. In the `WEB-INF/` directory, create a subdirectory called for example `syncComponent/`.
4. Install the Content Delivery API Server Role:

   a.

5. Access the SDL Tridion installation media and navigate to the `Content Delivery\roles\` folder.
6. Install JARs for Contact Subscription and Contact Synchronization: navigate to `Outbound E-Mail\Presentation System\Java\Subscription` and copy the JAR files to your Web site `\lib` directory.
7. Install JSP for Contact Synchronization: navigate to `Outbound E-Mail\Presentation System\Java\Subscription\Web` and copy `profilesync.jsp` to a directory in your Web site, for example `syncComponent\`.

   Note: You need to specify `profilesync.jsp` in your Synchronization Target.

8. Download the JDBC driver (JAR file) required for your specific database vendor and Java version, and copy it into the `lib` folder:

Database vendor	JDBC Driver	File to download
Microsoft SQL Server	Microsoft JDBC Driver 4.0 for SQL Server	`sqljdbc4.jar`
Oracle	Oracle JDBC driver 11.2.0.3	`ojdbc6.jar`

9. Configure the following files:

   a. `cd_storage_conf.xml`—see Configuring the Tracking and Subscription databases (see page 183)

   b. `cd_audience_manager.conf.xml`—see:

      - Configuring Outbound E-mail on Presentation Server (see page 207)
      - Configuring Audience Manager on Presentation Server (see page 192)

   c. `logback.xml` (configure logging).

10. Test the installation by opening a browser and running the `profilesync` page.

11. Configure security so that only the Content Manager Server can access this Web application, for example as follows:

    - Create a specific user with limited access, and configure the Web application so that it only allows this user access.
    - Use HTTPS to allow only secure access.
    - Implement IP-based access control.

Result

You have installed Synchronization Web pages.

12.9.3 Configuring a Synchronization Target

A Synchronization Target is used to synchronize data between a Content Manager Server and Presentation Server. For Audience Manager, a Synchronization Target is used to synchronize Contacts and Segments. You must specify the Synchronization Target used to keep Contacts and Mailings in that Publication in sync in a Publication **Audience Management** tab.

Requirements

To create a Synchronization Target you must have Tridion System Administration privileges.

Context

A Synchronization Target specifies a URL to a Synchronization Server where the `profilesync` Web page is installed. In a Publication, you need to specify the Synchronization Target that is used to keep Contacts and Segments in that Publication in sync.

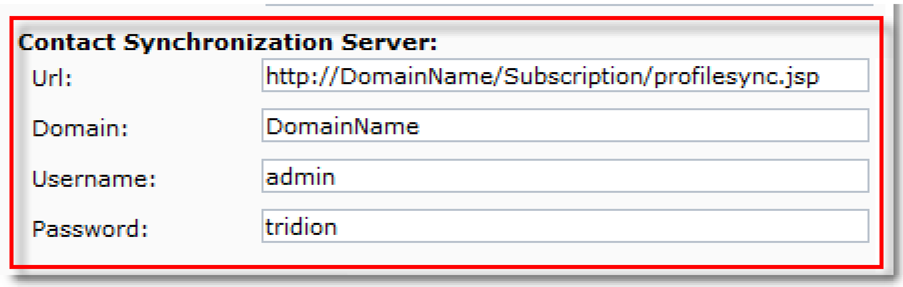
Contact Synchronization is performed by the **Synchronization service**. The service loops over the active Synchronization Targets that have a URL for **Contact Synchronization Server** defined and, for each server, synchronizes the Segments and the new/changed/deleted Contacts

You can implement Contact Subscription and Tracking on multiple Web sites. To avoid synchronization errors, you must only create a single Synchronization Target for each Tracking database and Subscription database used on the Presentation Server. When Web sites share databases, you assign the same Synchronization Target to the concerned Publications.

Note: It is recommended not to allow Contacts to subscribe or update subscription details on more than one Web site to ensure consistent synchronization.

Steps to execute

1. Open the Content Manager Explorer.
2. In the **System Administration** view, navigate to the **Synchronization Targets** node.
3. Select **Synchronization Targets** in the **Shortcuts** area and do one of the following:
   - Right-click **Synchronization Targets** and select **New** > **Synchronization Target** .
   - Access the **Administration** tab in the Ribbon and select **New Synchronization Target**.
4. In the **General** tab:
   - **Name**—enter a logical name
   - **Description**—enter a description
   - **Active**—deselect to deactivate the Synchronization Target, for example if a server is offline
5. In the **Contact Synchronization Server** section, specify the details for Audience Manager:
   - **Url**—the fully qualified URL of the Web page that synchronizes Contacts and Segments between Outbound E-mail and the Presentation Server (`profilesync.jsp` or `profilesync.aspx`).
   - **Domain**—the user domain name.
   - **Username**—the name of a user that can connect to the `Url`, for example `administrator`.
   - **Password**—the password used by the `Username`.



Contact Synchronization Server:

Url:	http://DomainName/Subscription/profilesync.jsp
Domain:	DomainName
Username:	admin
Password:	tridion

6. Click **Save and Close**.

7. Configure a Publication to use the Synchronization Target:

   a. Select a Publication and choose **Properties** from the context menu.

   b. Select the **Audience Management** tab.

   c. In the **Synchronization Target** field, select the Synchronization Target.

   d. Click **Save and Close** to save and close the Publication.

Result

You have created and configured a Synchronization Target.

12.9.4 Configuring synchronization

Configure synchronization settings in the `<PresentationSide>` section of the `OutboundEmail.xml` configuration file.

Steps to execute

1. Open `OutboundEmail.xml` located by default in the `config` subdirectory of your Content Manager root location in a plain-text editor or XML editor.

2. Locate the `<PresentationSide>` section:

```
<PresentationSide>
  <SynchronizationTimeout>100</SynchronizationTimeout>
  <TrackingCollectionInterval>60</TrackingCollectionInterval>
  <SynchronizationInterval>10</SynchronizationInterval>
  <SynchronizationBatchSize>100</SynchronizationBatchSize>
...
```

3. Specify the following synchronization settings:

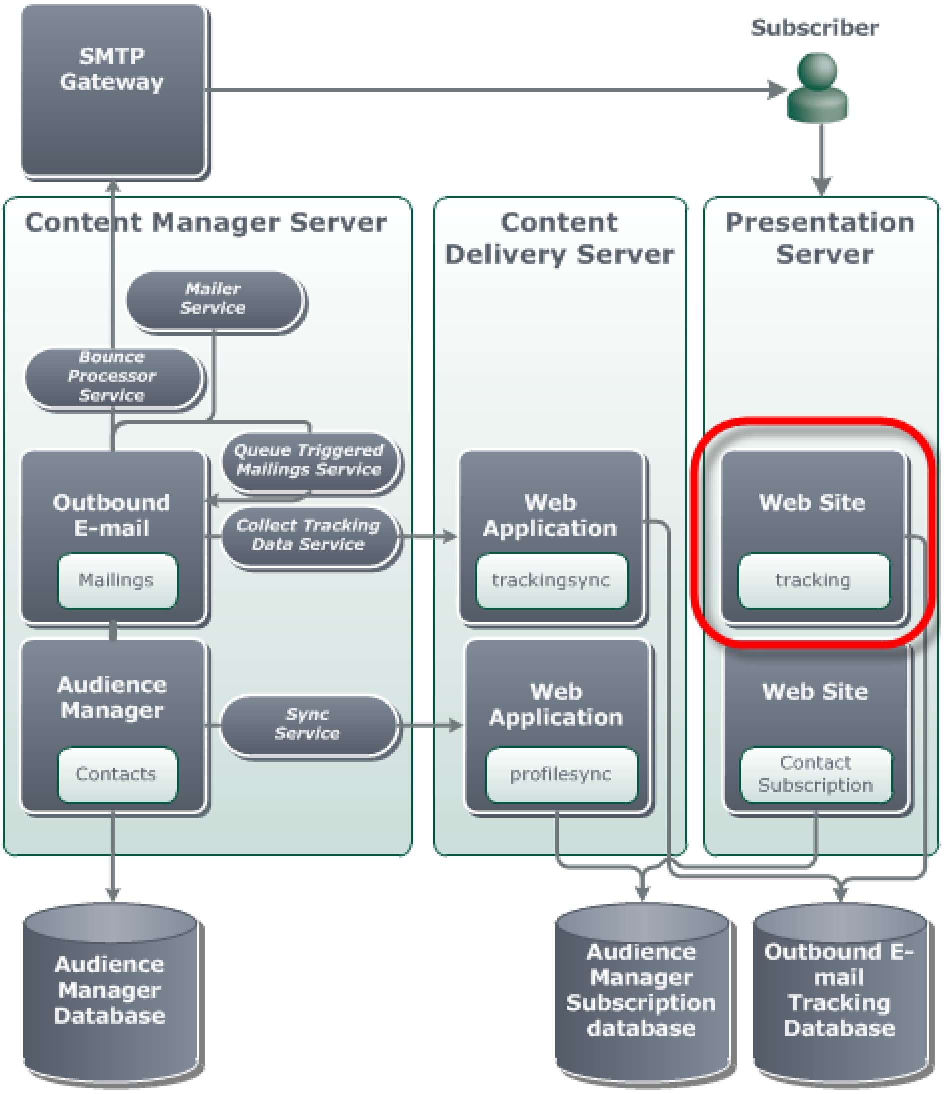
Element	Description
`<SynchronizationTimeout>`	(Optional) The number of seconds the Synchronization service has to perform an iteration before the service is timed out.

Element	Description
`<SynchronizationInterval>`	The number of seconds the Synchronization service waits before checking to see if there is new Contact data requiring synchronization.
`<SynchronizationBatchSize>`	The batch size for Contact Synchronization, that is the number of Contacts Outbound E-mail synchronizes at a time until there are no batches left to process.

4. Save and close `OutboundEmail.xml`.

12.10 Installing Tracking on Presentation Server

The Presentation Server is the Web application that tracks e-mail response. This section describes the installation and configuration of components used to implement Tracking.



Note: If you want to install Outbound E-mail Presentation Side components on a 64-bit operating system, but you are using a 32-bit Java Virtual Machine, your Web site must be running in a 32-bit process (Application Pool).

12.10.1 Installing tracking (ASP.NET)

You must install Tracking components on a publicly accessible Web site. This task describes how to install Tracking components on Internet Information Service (IIS).

Requirements

- You must have created a Tracking database.
- You must install Tracking on a publicly accessible Web site.

Context

Outbound E-mail provides a tracking Web page that you can use to track mailing response.

Steps to execute

1. Open Internet Information Services (IIS) Manager.
2. Go to your existing public-facing Web site.
3. Install and configure the Content Delivery API Server Role:
    a.
    b. Installing the API Server Role as a .NET Web application (see page 85)
4. Access the SDL Tridion installation media and navigate to the `Content Delivery\roles\` folder.
5. Install DLLs for Contact Synchronization, Tracking and Tracking retrieval: navigate to `Outbound E-Mail\Presentation System\NET\Synchronization\bin` and copy the DLLs to the `\bin` folder of your Web site.
6. Install ASPXs and default images for Tracking: navigate to `Outbound E-Mail\Presentation System\NET\Synchronization\` and copy `Tracking.aspx` and `Tracking.aspx.cs` (and default `Web.config`, if necessary) to the `\config` folder of your Web site.

    Note: You need to specify this page in the **Audience Management** tab of a **Publication**.

7. Install JARs for Tracking and Tracking retrieval: navigate to the `Outbound E-Mail\Presentation System\Java\Tracking` directory and copy the JAR files to your Web site `\lib` directory.
8. Download the JDBC driver (JAR file) required for your specific database vendor and Java version, and copy it into the `lib` folder:

Database vendor	JDBC Driver	File to download
Microsoft SQL Server	Microsoft JDBC Driver 4.0 for SQL Server	`sqljdbc4.jar`
Oracle	Oracle JDBC driver 11.2.0.3	`ojdbc6.jar`

9. Configure the following files:
    a. `cd_storage_conf.xml`—see Configuring the Tracking and Subscription databases (see page 183)
    b. `cd_audience_manager.conf.xml`—see:

- Configuring Outbound E-mail on Presentation Server (see page 207)
- Configuring Audience Manager on Presentation Server (see page 192)

c. `logback.xml` (configure logging).

Result

You have installed the Tracking Web page.

12.10.2 Installing tracking (Java)

You must install Tracking components on a publicly accessible Web site. This task describes how to install Tracking components on a Java-based Web Application Server.

Requirements

- You must have created a Tracking database.
- You must install Tracking on a publicly accessible Web site.

Context

Outbound E-mail provides a tracking Web page that you can use to track mailing response.

Steps to execute

1. Go to your existing public-facing Web site.
2. Install and configure the Content Delivery API Server Role:

   a.

   b. Installing the API Server Role for a Java Web application (see page 104)
3. Access the SDL Tridion installation media and navigate to the `Content Delivery\roles\` folder.
4. Install JARs for Tracking and Tracking retrieval: navigate to the `Outbound E-Mail\Presentation System\Java\Tracking` directory and copy the JAR files to your Web site `\lib` directory.
5. Install JSP and default images for Tracking: navigate to `Outbound E-Mail\Presentation System\Java\Tracking\Web` and copy all files to a directory in your Web site.

   Note: You need to specify this page in the **Audience Management** tab of a **Publication**.
6. Download the JDBC driver (JAR file) required for your specific database vendor and Java version, and copy it into the `lib` folder:

Database vendor	JDBC Driver	File to download
Microsoft SQL Server	Microsoft JDBC Driver 4.0 for SQL Server	`sqljdbc4.jar`
Oracle	Oracle JDBC driver 11.2.0.3	`ojdbc6.jar`

7. Configure the following files:

   a. `cd_storage_conf.xml`—see Configuring the Tracking and Subscription databases (see page 183)

   b. `cd_audience_manager.conf.xml`—see:

      - Configuring Outbound E-mail on Presentation Server (see page 207)
      - Configuring Audience Manager on Presentation Server (see page 192)

   c. `logback.xml` (configure logging).

Result

You have installed the Tracking Web page.

12.10.3 Configuring Publications for tracking

In a Publication **Audience Management** tab, you can specify the Synchronization Target used to synchronize data between the Content Manager and Presentation Server and the URLs of Contact Subscription pages and the Tracking page.

Requirements

To configure Publications for tracking and synchronization, you must have Tridion System Administrator rights or Publication Management rights.

Steps to execute

1. Open the Content Manager Explorer.

2. Select a Publication and choose **Properties** from the context menu.

3. Select the **Audience Management** tab.

4. Fill in the following field:

   - **Tracking page URL**—the URL of the Web page that tracks mailing response, for example whether a email recipient opened a Mailing and what links were clicked. You install the `tracking.jsp` or `Tracking.aspx` Web page on a publicly accessible Web site.

Note: For information about configuring other fields the **Audience Management** tab, see Configuring Publications for Contact Subscription (see page 189).

5. Click **Save and Close** to save and close the Publication.

Result

You have configured a Publication for tracking.

Next steps

For more information on subscription management and tracking and assigning Address Books to Publications, see the Implementing Audience Manager and Outbound E-mail section of the implementor's documentation portal.

12.10.4 Configuring Outbound E-mail on Presentation Server

The `cd_audience_manager.conf.xml` is the configuration file for Audience Manager and Outbound E-mail on the Presentation Server.

Steps to execute

1. On a .NET system, navigate to `Outbound E-Mail\Presentation System \NET\config` and copy the following file to your Tridion installation `\config` directory:
   - `cd_audience_manager.conf.xml`
2. On a Java system, navigate to `Outbound E-Mail\Presentation System \Java\config` and copy the following file to your Tridion installation `\classes` directory:
   - `cd_audience_manager.conf.xml`
3. Open `cd_audience_manager.conf.xml` in a text editor:
4. For Outbound E-mail you can configure various settings:

   `<OpenImage>`

   The URL of the image to display in an e-mail (to track when it is opened).

   `<ErrorResolvingLink>`

   The URL to redirect to when a Component Link cannot be resolved.

   `<ErrorResolvingBinary>`

   The URL to redirect to when a MultiMedia Component Link cannot be resolved.

   `<GeneralError>`

   The URL to redirect to in case of a general error.

`<AppendTrackingInfo>`

Whether or not the TrackingInfo parameter should be added to the URL when redirecting.

```
<Configuration>
  <Email>
     <OpenImage>MailOpened.gif</OpenImage>
  </Email>
  <Redirect>
     <ErrorResolvingLink>Resolve.html</ErrorResolvingLink>
     <ErrorResolvingBinary>Empty.gif</ErrorResolvingBinary>
     <GeneralError>Error.html</GeneralError>
     <AppendTrackingInfo>false</AppendTrackingInfo>
  </Redirect>
  ...
</Configuration>
```

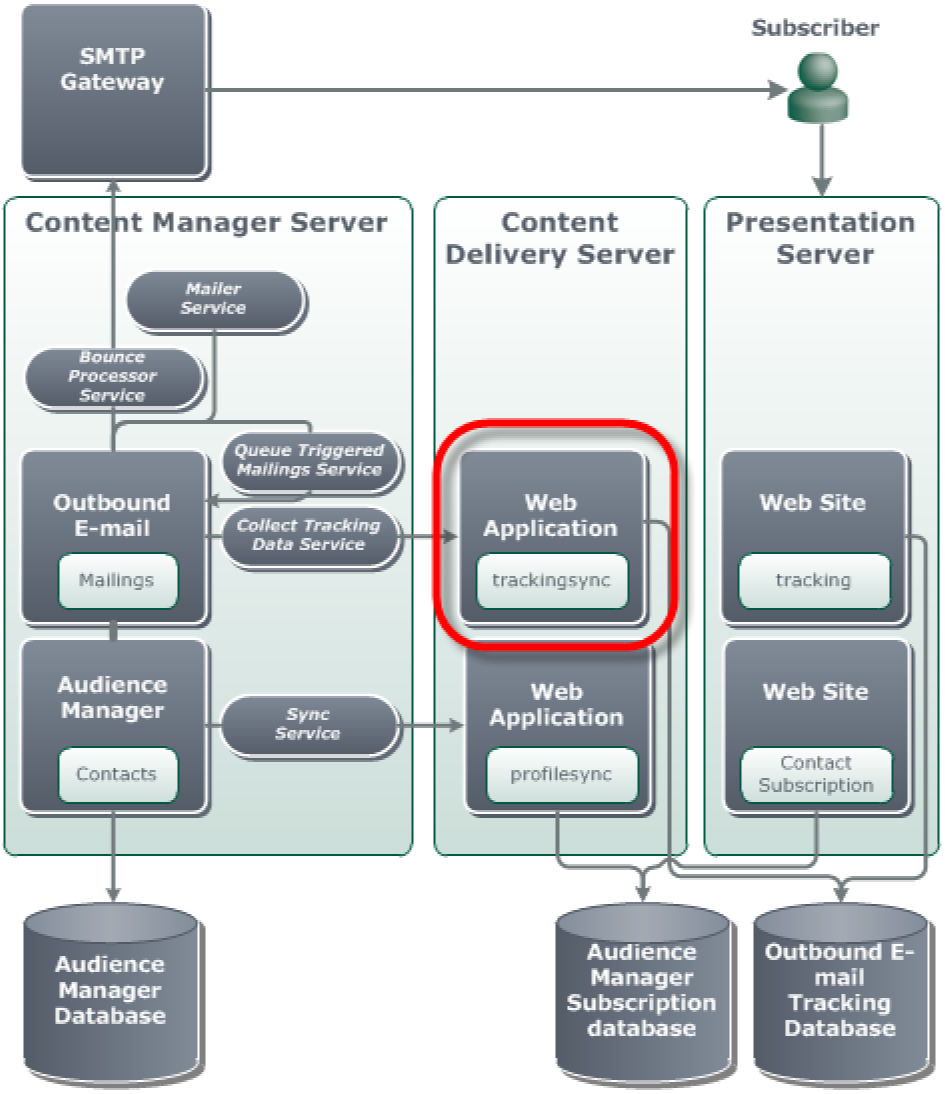
Note: If you are running multiple instances of Content Delivery on your ASP.NET Web site, you need to add the following code to your ASPX pages to use separate configuration files for each Web site (on a single line):

```
Tridion.OutboundEmail.ContentDelivery.Utilities.
Settings.ConfigFileName = "NameOfConfig.xml";
```

5. Save and close `cd_audience_manager.conf.xml`.

12.11 Setting up a Synchronization Server for tracking retrieval

The Synchronization Server retrieves tracking information from the Tracking database on the presentation side. You install Tracking components in a .NET or Java Web application.



12.11.1 Installing tracking retrieval (ASP.NET)

The Synchronization Server retrieves tracking information from the Tracking database on the presentation side. This task describes how to set up a Synchronization Server for tracking retrieval on Internet Information Service (IIS).

Requirements

- It is recommended to install Contact synchronization in a separate Web application that is not publicly available. Refer to Content Delivery prerequisites (see page 67) to learn which Web and Application Servers are supported.

Before installing tracking retrieval:

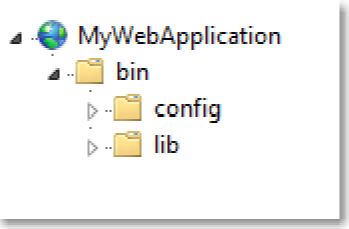
- Install the Content Delivery API Server Role in your Web application (for database access only—do not configure the Ambient Data Framework).
- Create a Tracking database.

Context

Outbound E-mail provides a tracking synchronization Web page to retrieve tracking information from the Tracking database on the presentation side.

Steps to execute

1. Open Internet Information Services (IIS) Manager.
2. Create a New Web site:
   a. Select the Web sites and node and choose **New** > **Web site** from the context menu.
   b. Follow the steps in the Web Site Creation Wizard.
3. Create the following Web application directory structure:
   a. In the root folder, create a `bin\` subfolder.
   b. In the `bin\` folder, create subfolders `config\` and `lib\`.



4. In the `bin\` folder, create a subfolder called for example `syncComponent\`.
5. Install the Content Delivery API Server Role:
   a.
6. Access the SDL Tridion installation media and navigate to the `Content Delivery\roles\` folder.
7. Install DLLs for Contact Synchronization, Tracking and Tracking retrieval: navigate to `Outbound E-Mail\Presentation System\NET\Synchronization\bin` and copy the DLLs to the `\bin` folder of your Web site.
8. Install ASPXs for Tracking retrieval: navigate to `Outbound E-Mail\Presentation System\NET\Synchronization\` and copy `Trackingsync.aspx` and `Trackingsync.aspx.cs` (and default `Web.config`, if necessary) to a directory in your Web site, for example `syncComponent\`.

Note: You need to specify Trackingsync.aspx in your Synchronization Target.

9. Install JARs for Tracking and Tracking retrieval: navigate to the Outbound E-Mail\Presentation System\Java\Tracking directory and copy the JAR files to your Web site \lib directory.

10. Download the JDBC driver (JAR file) required for your specific database vendor and Java version, and copy it into the lib folder:

Database vendor	JDBC Driver	File to download
Microsoft SQL Server	Microsoft JDBC Driver 4.0 for SQL Server	sqljdbc4.jar
Oracle	Oracle JDBC driver 11.2.0.3	ojdbc6.jar

11. Configure the following files:

    a. cd_storage_conf.xml—see Configuring the Tracking and Subscription databases (see page 183)

    b. cd_audience_manager.conf.xml—see:

        - Configuring Outbound E-mail on Presentation Server (see page 207)
        - Configuring Audience Manager on Presentation Server (see page 192)

    c. logback.xml (configure logging).

12. Test the installation by opening a browser and running the trackingsync page.

13. Configure security so that only the Content Manager Server can access this Web application, for example as follows:

    - Create a specific user with limited access, and configure the Web application so that it only allows this user access.
    - Use HTTPS to allow only secure access.
    - Implement IP-based access control.

Result

You have installed the Tracking Retrieval Web page.

12.11.2 Installing tracking retrieval (Java)

The Synchronization Server retrieves tracking information from the Tracking database on the presentation side. This task describes how to set up a Synchronization Server on a Java-based Web Application Server.

Requirements

- It is recommended to install Contact synchronization in a separate Web application that is not publicly available. Refer to Content Delivery prerequisites (see page 67) to learn which Web and Application Servers are supported.

Before installing tracking retrieval:

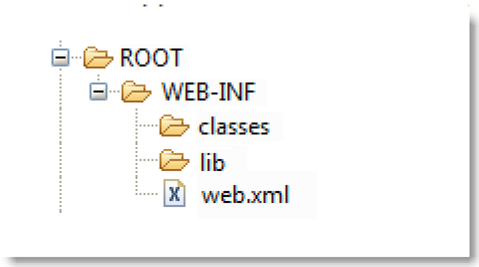
- Install the Content Delivery API Server Role in your Web application (for database access only—do not configure the Ambient Data Framework).
- Create a Tracking database.

Context

Outbound E-mail provides a tracking synchronization Web page to retrieve tracking information from the Tracking database on the presentation side.

Steps to execute

1. Create a new Web application in your Web Application Server.
2. Create the following Web application directory structure:
   a. In the root folder, create a `WEB-INF/` directory.
   b. In the `WEB-INF/` directory, create subdirectories `classes/` and `lib/`.



3. In the `WEB-INF/` directory, create a subdirectory called for example `syncComponent/`.
4. Install the Content Delivery API Server Role:
   a.
5. Access the SDL Tridion installation media and navigate to the `Content Delivery\roles\` folder.
6. Install JARs for Tracking and Tracking retrieval: navigate to the `Outbound E-Mail\Presentation System\Java\Tracking` directory and copy the JAR files to your Web site `\lib` directory.
7. Install JSP for Tracking retrieval: navigate to `Outbound E-Mail\Presentation System\Java\Tracking\Web` and copy `trackingsync.jsp` to a directory in your Web site, for example `syncComponent\`.

Note: You need to specify `trackingsync.jsp` in your Synchronization Target.

8. Download the JDBC driver (JAR file) required for your specific database vendor and Java version, and copy it into the `lib` folder:

Database vendor	JDBC Driver	File to download
Microsoft SQL Server	Microsoft JDBC Driver 4.0 for SQL Server	`sqljdbc4.jar`
Oracle	Oracle JDBC driver 11.2.0.3	`ojdbc6.jar`

9. Configure the following files:

   a. `cd_storage_conf.xml`—see Configuring the Tracking and Subscription databases (see page 183)

   b. `cd_audience_manager.conf.xml`—see:
      - Configuring Outbound E-mail on Presentation Server (see page 207)
      - Configuring Audience Manager on Presentation Server (see page 192)

   c. `logback.xml` (configure logging).

10. Test the installation by opening a browser and running the `trackingsync` page.

11. Configure security so that only the Content Manager Server can access this Web application, for example as follows:
    - Create a specific user with limited access, and configure the Web application so that it only allows this user access.
    - Use HTTPS to allow only secure access.
    - Implement IP-based access control.

Result

You have installed the Tracking Retrieval Web page.

12.11.3 Configuring a Synchronization Target

A Synchronization Target synchronizes data between Content Manager and a Presentation Server. In Outbound E-mail, it synchronizes e-mail tracking data to gather statistics about the status of Mailings. Specify the Synchronization Target used to keep Contacts and Mailings in that Publication in sync in the **Audience Management** tab.

Requirements

To create a Synchronization Target you must have Tridion System Administration privileges.

Context

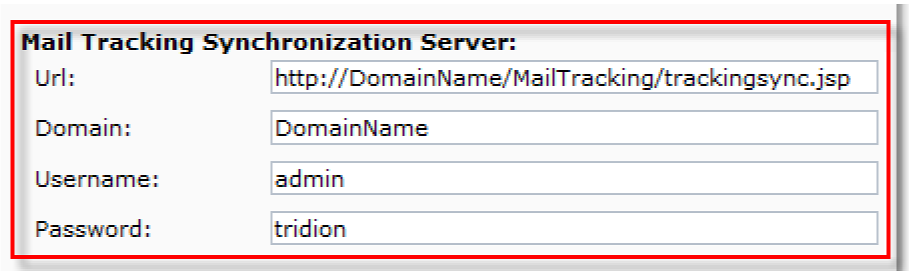
A Synchronization Target specifies a URL to a Synchronization Server where the `trackingsync` Web page is installed. In a Publication, you need to specify the Synchronization Target that is used to keep Mailings in that Publication in sync.

Synchronization is performed by the **Collect Tracking Data** service. The **Collect tracking data service** loops over the active Synchronization Targets that have a URL for **Tracking Synchronization Server** defined and retrieves all tracking data from the Synchronization Target, including data for Contacts not synchronized to the server.

You can implement Contact Subscription and Tracking on multiple Web sites. To avoid synchronization errors, you must only create a single Synchronization Target for each Tracking database and Subscription database used on the Presentation Server. When Web sites share databases, you assign the same Synchronization Target to the concerned Publications.

Steps to execute

1. Open the Content Manager Explorer.
2. In the **System Administration** view, navigate to the **Synchronization Targets** node.
3. Select **Synchronization Targets** in the **Shortcuts** area and do one of the following:
    - Right-click **Synchronization Targets** and select **New** > **Synchronization Target** .
    - Access the **Administration** tab in the Ribbon and select **New Synchronization Target**.
4. In the **General** tab:
    - **Name**—enter a logical name
    - **Description**—enter a description
    - **Active**—deselect to deactivate the Synchronization Target, for example if a server is offline
5. In the **Mail Tracking Synchronization Server** section, enter details for Outbound E-mail:
    - **Url**—the fully qualified URL of the `trackingsync` Web page that performs the tracking of e-mail response.
    - **Domain**—the user domain name.
    - **Username**—the name of a user that can connect to the `Url`, for example `administrator`.
    - **Password**—the password used by the `Username`.



Mail Tracking Synchronization Server:

Url:	http://DomainName/MailTracking/trackingsync.jsp
Domain:	DomainName
Username:	admin
Password:	tridion

6. Click **Save and Close**.

7. Configure a Publication to use the Synchronization Target:

    a. Select a Publication and choose **Properties** from the context menu.

    b. Select the **Audience Management** tab.

    c. In the **Synchronization Target** field, select the Synchronization Target.

    d. Click **Save and Close** to save and close the Publication.

Result

You have created and configured a Synchronization Target.

12.11.4 Configuring tracking

Configure tracking settings in the `<PresentationSide>` section of the `OutboundEmail.xml` configuration file.

Steps to execute

1. Open `OutboundEmail.xml` located by default in the `config` subdirectory of your Content Manager root location in a plain-text editor or XML editor.

2. Locate the `<PresentationSide>` section:

```
<PresentationSide>
  <SynchronizationTimeout>100</SynchronizationTimeout>
  <TrackingCollectionInterval>60</TrackingCollectionInterval>
  <SynchronizationInterval>10</SynchronizationInterval>
  <SynchronizationBatchSize>100</SynchronizationBatchSize>
...
```

3. Specify the following tracking settings:

Element	Description
`<SynchronizationTimeout>`	(Optional) The number of seconds the Synchronization service has to perform an iteration before the service is timed out.

Element	Description
`<TrackingCollectionInterval>`	The number of seconds the Collect Tracking Data service waits before checking to see if there is new tracking data on the presentation side.

4. Save and close `OutboundEmail.xml`.

12.12 Outscaling the Mailer service

You can install the Mailer service on a separate machine to improve performance.

Requirements

You have installed and configured Content Manager Server, Audience Manager and Outbound E-mail on a machine.

Outbound E-mail requires a SMTP server to process outgoing e-mails.

Context

When you install the Mailer service on a separate machine, you need to configure:

- the user running the Mailer service (in the service **Properties**)
- database usage in `OutboundEmail.xml` (use the same database so that each system on which the Mailer service is installed reads from a shared database queue)
- communication with the Content Manager Server via the Core Service in `OutboundEmail.xml`

Note: It is recommended to use the default `NetTcpBinding` or `WSHttpBinding` to connect to the Core Service. If you want to develop your own `NetTcpBinding` or `WSHttpBinding` to override the default ones, create a configuration file for the service, called `OEMailer.exe.config`, and save in your `\Tridion\bin` directory (the same location as `OEMailer.exe`). For more information, see the Implementing Content Manager section of the implementer's documentation portal.

The following procedure describes how to install and configure the Mailer service on a separate machine:

Steps to execute

1. Access the SDL Tridion installation media.

2. Navigate to the `Content Manager\` directory.

3. Double-click the executable `SDLTridion2013CM.exe`.

4. In the features to install screen, switch to the **Advanced** view and select the following options: **Audience Manager** > **Outbound E-mail** > **Mailer Service** .
5. Click **Next** and follow the instructions to complete the installation.

   Note: The installer also installs the Content Manager Server because the Mailer Service requires some Content Manager Server components to run. However, you do not need to set up a running Content Manager Server on the machine on which you added this Mailer Service.

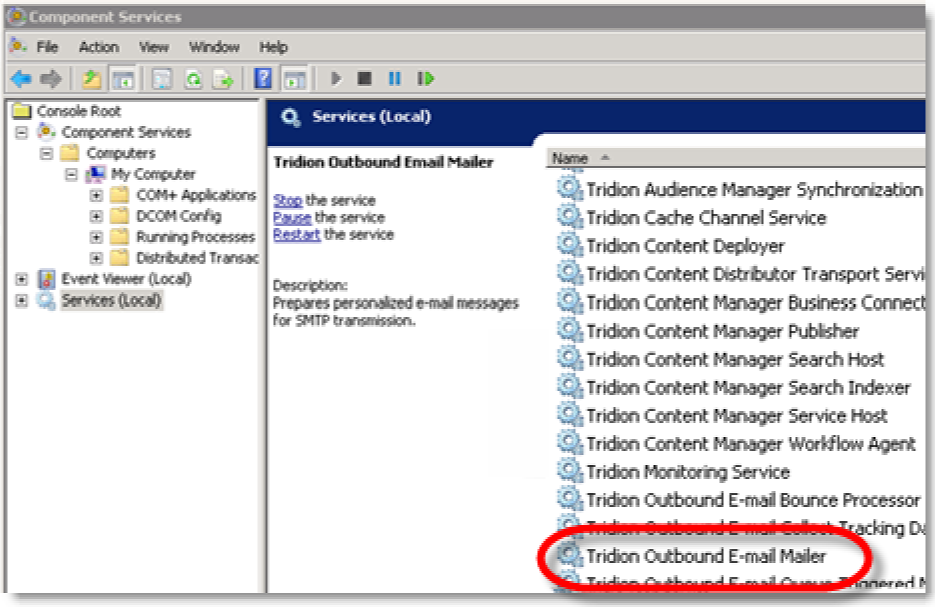
6. Configure the user that is used to run the Mailer service:

   Note: The user that runs the Mailer service can be any user but you must also add this user to the list of Content Manager users. You can add Content Manager users from your existing Windows Domain (Active Directory) and from LDAP directories.

   a. Open Component Services.

   b. Stop the **Tridion Outbound Email Mailer** service.

   c. Select **Tridion Outbound Email Mailer** and choose **Properties** in the context menu:



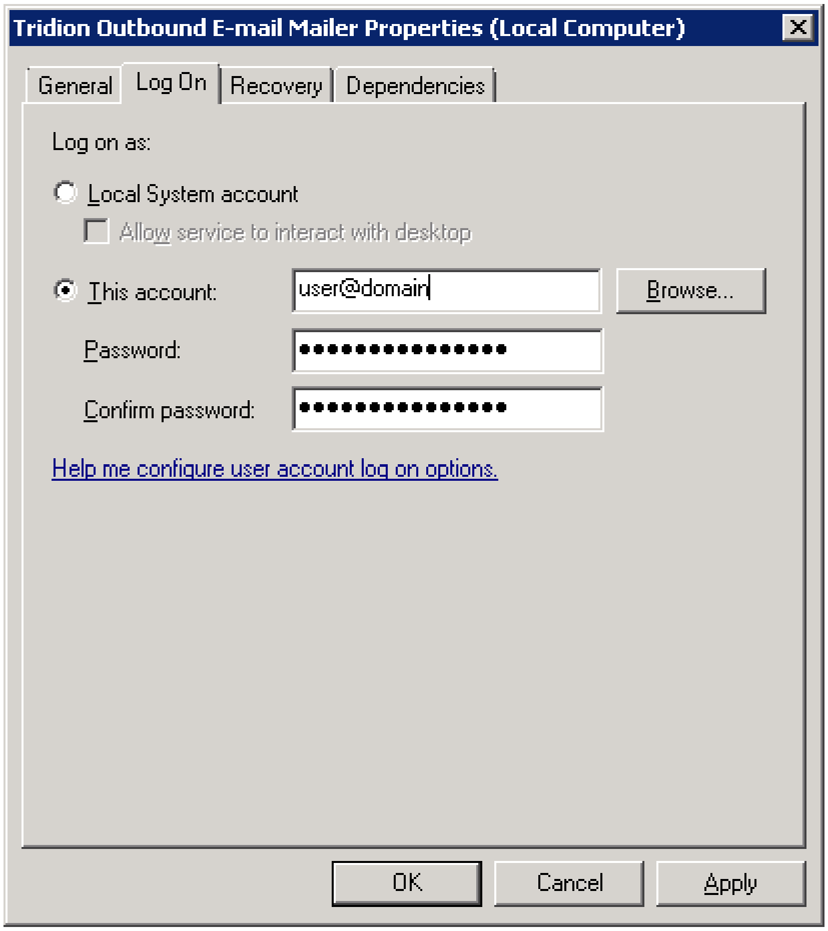
Tridion Outbound Email Mailer

Stop the service
Pause the service
Restart the service

Description:
Prepares personalized e-mail messages for SMTP transmission.

Tridion Audience Manager Synchronization
Tridion Cache Channel Service
Tridion Content Deployer
Tridion Content Distributor Transport Servi
Tridion Content Manager Business Connect
Tridion Content Manager Publisher
Tridion Content Manager Search Host
Tridion Content Manager Search Indexer
Tridion Content Manager Service Host
Tridion Content Manager Workflow Agent
Tridion Monitoring Service
Tridion Outbound E-mail Bounce Processor
Tridion Outbound E-mail Collect Tracking D
Tridion Outbound E-mail Mailer
Tridion Outbound E-mail Queue Triggered M

   d. In the **Log On** menu, select **This account** and click **Browse**:


Tridion Outbound E-mail Mailer Properties (Local Computer)
General
Log On
Recovery
Dependencies
Browse...
Help me configure user account log on options.
OK
Cancel
Apply

Log on as:

○ Local System account

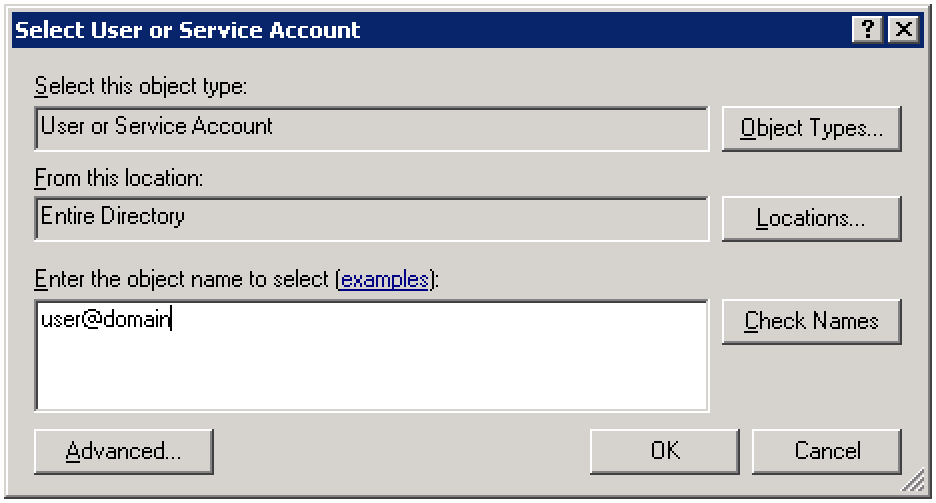
□ Allow service to interact with desktop

● This account: user@domain

Password: ●●●●●●●●●●●●●●●

Confirm password: ●●●●●●●●●●●●●●●

e. Select a user and click **OK** (twice):



Select this object type: User or Service Account

From this location: Entire Directory

Enter the object name to select (examples): user@domain

f. Restart the **Tridion Outbound Email Mailer** service.

7. Configure the `OutboundEmail.xml` configuration file:

   a. Open `OutboundEmail.xml`, located by default in the `config\` subdirectory of the Content Manager root location (defaults to

`C:\Program Files (x86)\Tridion\`in a plain-text editor or XML editor.

b. Locate the `<System>` section. If you want to connect to the Core Service using the default `NetTcpBinding` provided, replace `localhost` with the server name of the Content Manager Server:

```
<System>
    <LicenseFile></LicenseFile>
    <CoreServiceUrl>net.tcp://localhost:2660/CoreService/netTcp</CoreServiceUrl>
    <MonitoringAgentPort>20131</MonitoringAgentPort>
</System>
```

Note: If connecting using `NetTcpBinding`, the user running the Core Service user must be an Impersonation User.

To connect to the Core Service using `wsHttpBinding`, replace `<CoreServiceUrl>` with, for example, the following and replace `localhost` with the server name of the Content Manager Server:

```
<CoreServiceUrl>http://localhost/webservices/CoreService2011.svc/wsHttp</
CoreServiceUrl>
```

If you are using the SDL Tridion 2011-compatible endpoints, replace `CoreService2011.svc` with `CoreService.svc`.

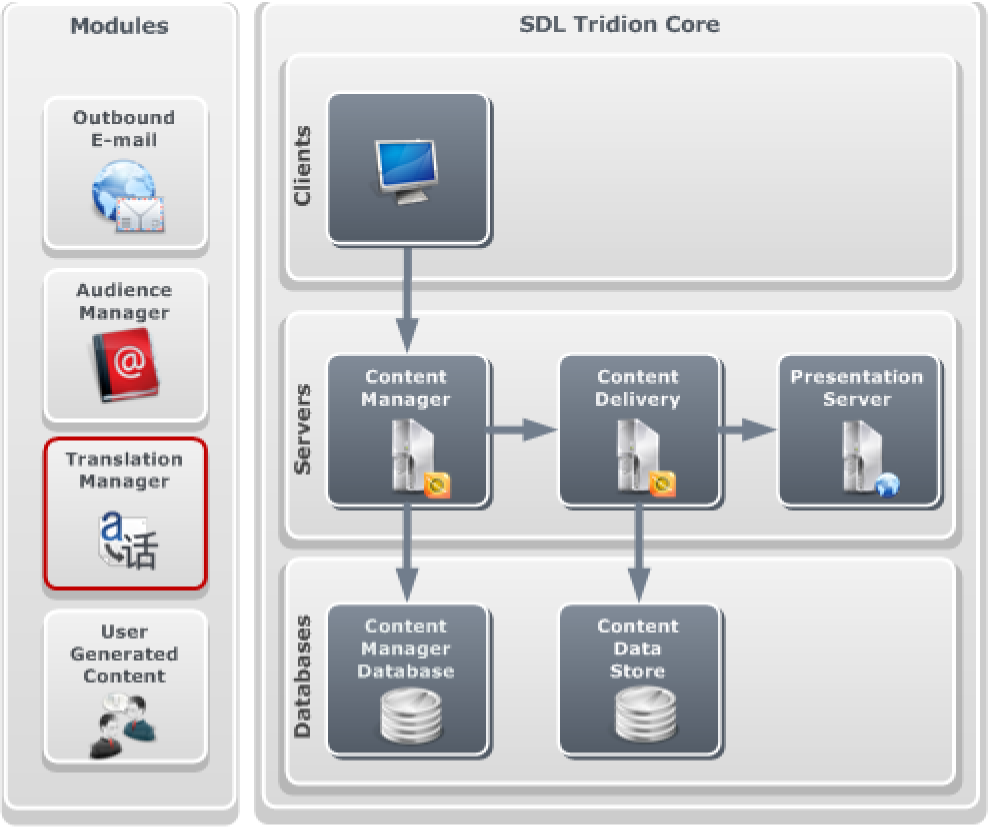
c. Locate and configure the `<Database>` section—for more information, see Configuring the Audience Manager database (see page 166)

d. Locate and configure the `<Mail>` section—for more information, see Configuring outgoing e-mail (see page 174)

e. Save and close `OutboundEmail.xml`.

Chapter 13 Installing Translation Manager

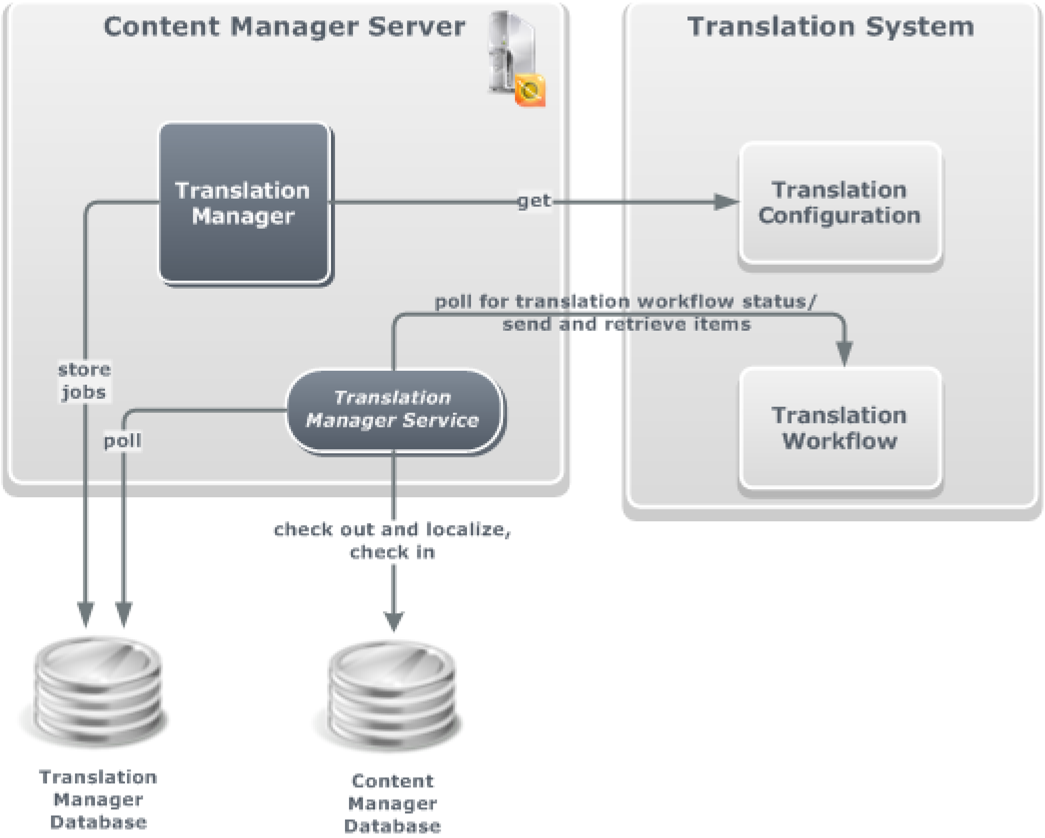
Translation Manager allows you to send content stored in the Content Manager of SDL Tridion for translation to a translation management system. This section describes how to install and get Translation Manager up and running.



13.1 Translation Manager system diagram

Translation Manager integrates Content Manager of SDL Tridion with a translation management system (the SDL **T**ranslation **M**anagement **S**ystem (SDL TMS) or SDL WorldServer).

The following diagram shows the Translation Manager software components and how they integrate the Content Manager of SDL Tridion with the translation management system.


Content Manager Server
Translation System
Translation Manager
get
Translation Configuration
poll for translation workflow status/ send and retrieve items
store jobs
Translation Manager Service
poll
Translation Workflow
check out and localize, check in
Translation Manager Database
Content Manager Database

Translation Manager consists of the following software components that integrate the Content Manager of SDL Tridion with a translation management system.

Content Manager Explorer Translation Extension

Enables users to initiate and track translations from the Content Manager Explorer, the Web-based Graphical User Interface to the Content Manager.

Translation Manager core

Contains the communication logic between translation management system and the Content Manager. The Translation Manager accesses the translation management system to retrieve configuration information used to configure Publications and Organizational Items for translation, stores Translation Jobs in a database, and receives status updates on translations.

Translation Manager Service

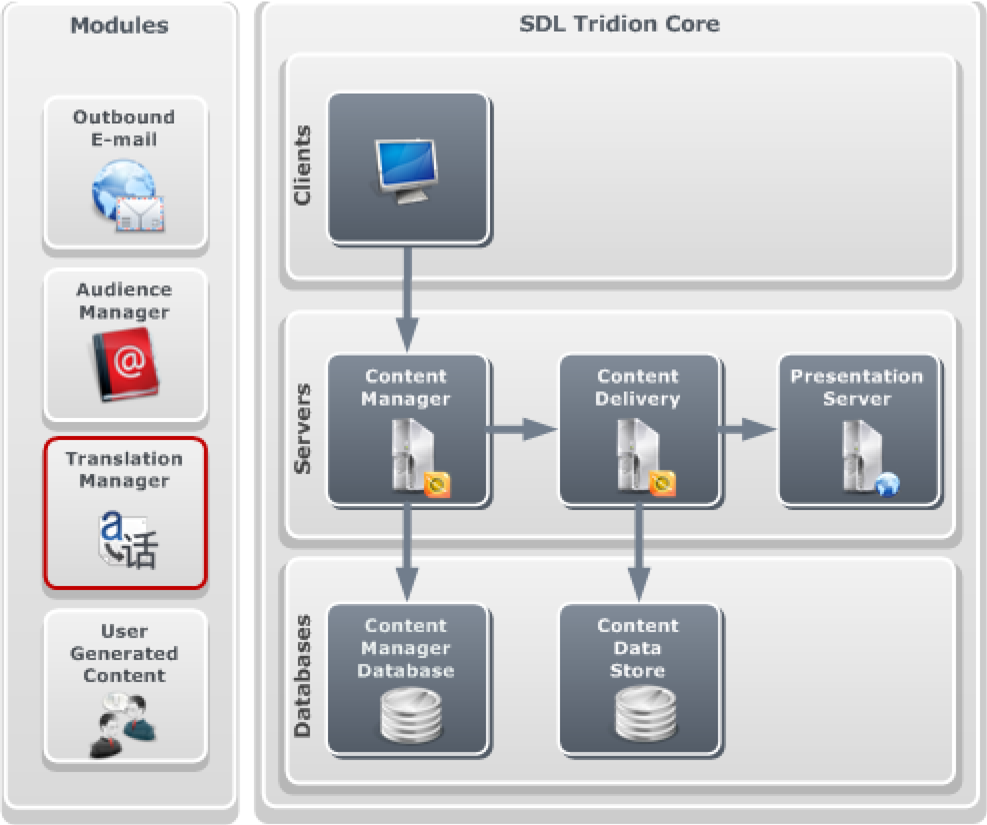
Synchronizes the translation management and Content Manager systems. The Translation Manager Service reads Translation Jobs from the Translation Manager database and performs the actions necessary to handle the jobs such as localizing and checking items out, creating jobs for transferral to translation management system, copying over and checking items in when translations have been completed, and updating the status of jobs.

Translation Manager Database

Stores Translation Jobs created in the Content Manager.

13.2 Translation Manager prerequisites

This topic lists the required and supported software for Translation Manager.



Content Manager side prerequisites

Translation Manager is fully integrated into the Content Manager and is installed as a part of the Content Manager installation, therefore Content Manager server prerequisites apply.

Database prerequisites on Content Manager side

The Translation Manager requires an Oracle or Microsoft SQL Server database server for storing Translation Jobs created in the Content Manager.

The supported versions of Oracle or Microsoft SQL Server are identical to those of the Content Manager database. For **Oracle**, you require in addition the Oracle XML Developer Kit installed (in a default installation, this is installed automatically). Refer to Content Manager database prerequisites (see page 14) for details.

Translation Management system prerequisites

Translation Manager requires connection to a translation system. Translation Manager supports SDL Translation Management System (SDL TMS) and SDL WorldServer:
SDL WorldServer

Translation Manager supports SDL WorldServer 10.1.

SDL TMS

Translation Manager supports the following versions of SDL TMS:

- SDL TMS 2011 SP3
- SDL TMS 2011 SP2 (deprecated)
- SDL TMS 2007 SP4[1] (latest patch applied)

Note: [1] You can use SDL TMS version 2007 SP4 on all Translation Manager versions provided you set `compression="false"`. You configure compression in the `<TmsServer>` element in the `TranslationManager.xml` configuration file, for example:

```
<TmsServer port="80" compression="false" ssl="false">http://tridion.sdlproducts.com</TmsServer>
```

13.3 Creating the Translation Manager Database

Translation Manager needs a database. The Translation Manager database stores Translation Jobs. You use a PowerShell script to create a Microsoft SQL Server or Oracle database for Online Marketing Explorer.

13.3.1 Creating the Translation Manager database

Use a PowerShell script to create a Microsoft SQL Server or Oracle database for Translation Manager.

Requirements

The prerequisites for the Translation Manager Database are the same as the Content Manager Database.

To install SDL Tridion databases on a Windows machine requires Windows PowerShell 3.0 and Microsoft .NET Framework 4.0 or higher.

Steps to execute

1. Open Windows PowerShell from the Windows Start Menu.
2. In PowerShell, depending on your database navigate to one of the following folders on the installation media:
   - `Database\MSSQL\` or
   - `Database\Oracle\`
3. Type the following command to install the Translation Manager database:

   ```
   & '.\Install Translation Manager database.ps1'
   ```

4. Follow the instructions in the PowerShell console to create the database.

Result

You have created the Translation Manager database.

Next steps

When you install Translation Manager, you are prompted to provide configuration details for the Audience Manager database. For more information, see Running the installer (see page 224).

13.4 Setting up a Translation System

When you run the installer you are prompted to provide details of your translation system. For information about *Setting up SDL Translation Management System* and *Setting up SDL WorldServer*, see the implementor's documentation portal.

13.5 Installing Translation Manager

This section describes how to install and configure Translation Manager on your Content Manager Server using the installer. The installer installs the **Content Manager Explorer Translation Extension**, the **Translation Manager** core component and the **Translation Manager Service**.

13.5.1 Running the installer

You install Translation Manager by running the Content Manager Server installer

Requirements

Before running the installer you need to have created the Translation Manager database. The installer will prompt you for details about the database and database user credentials.

Context

By default, the Content Manager installer installs all software components. You will therefore only need to install Translation Manager if you chose not to when you installed the Content Manager Server, or when you want to outscale components.

Steps to execute

1. Access the SDL Tridion installation media.
2. Navigate to the `Content Manager\` directory.
3. Double-click the executable `SDLTridion2013CM.exe`.

4. Follow the instructions on the screen:
   - You can select individual software components to install by choosing the **Advanced** option (by default, the installer installs all software components).
   - You can install software components you choose not to install by running the installer again and selecting the **Modify** option.

5. After the installer finishes, if the installer does not prompt you to restart your machine, perform a manual reboot.

Result

The installer has created a log file in the following location:

- the subdirectory `SDL\Tridion\Logs\` of the path indicated by the `%PROGRAMDATA%` environment variable (defaults to `c:\ProgramData\`).

The following software components are installed for Translation Manager.

- Content Manager Explorer Translation Extension
- Translation Manager core
- Translation Manager Service

Next steps

The settings you specified are stored in the Translation Manager configuration file `TranslationManager.xml` located by default in your Tridion installation `\config` directory.

13.5.2 Configuring logging

The `<LogFolder>` element in the `TranslationManager.xml` file specifies the location on your file system where logging information is generated.

Steps to execute

1. Open the `TranslationManager.xml` file in a text editor.

   By default, the file is located in your `TRIDION_HOME\config` directory.

2. In the `<LogFolder>` section, enter the location on your file system where logging information is generated. For example:

```
<LogFolder verbose="true">C:\Program Files
    (x86)\Tridion\log</LogFolder>
```

Attributes

The `<LogFolder>` element has the following attributes:

Attribute	Description
`verbose`	Boolean for generating extensive log files for troubleshooting. The default is `false`.

Note: The default logging folder is the same as your Tridion logging folder.

3. Save and close the file.

13.5.3 Encrypting sensitive strings for Translation Manager

The `TranslationManager.xml` is the Translation Manager configuration file. You can encrypt passwords in the `TranslationManager.xml` configuration file using the encryption tool delivered with Translation Manager.

Context

As it is not good practice to include sensitive data in configuration files in cleartext, it is recommended to encrypt sensitive data using the command line tool.

Steps to execute

1. Navigate to your Tridion installation `/Translation Manager` folder.
2. Run the executable with the following command to **encrypt** passwords:

   ```
   EncryptTranslationManagerConfiguration.exe /e
   ```

3. Run the executable with the following command to **decrypt** passwords:

   ```
   EncryptTranslationManagerConfiguration.exe /d
   ```

Result

The tool encrypts the `<Database>` password and, if specified, the `<SmtpServer>` password in the `TranslationManager.xml` configuration file. If encryption failed, the tool remains open and describes the nature of the errors. Fix the errors before proceeding.

13.5.4 Connecting to the translation management system using HTTPS

To connect to the translation management system using HTTPS (SSL), configure the `<TmsServer>` element in the `TranslationManager.xml` file and install a security certificate on each machine where the Translation Manager Service is installed.

Context

If you have local policies in place for obtaining security certificates, use these. Otherwise, you can obtain a certificate via the certificate authorities which come pre-installed in Web browsers.

Note: The log file reports the following error if you do not have a certificate installed: `Using https causes Could not establish trust relationship for the SSL/TLS secure channel.`

Steps to execute

1. Open the `TranslationManager.xml` file by default located in your Tridion installation `\config` directory.

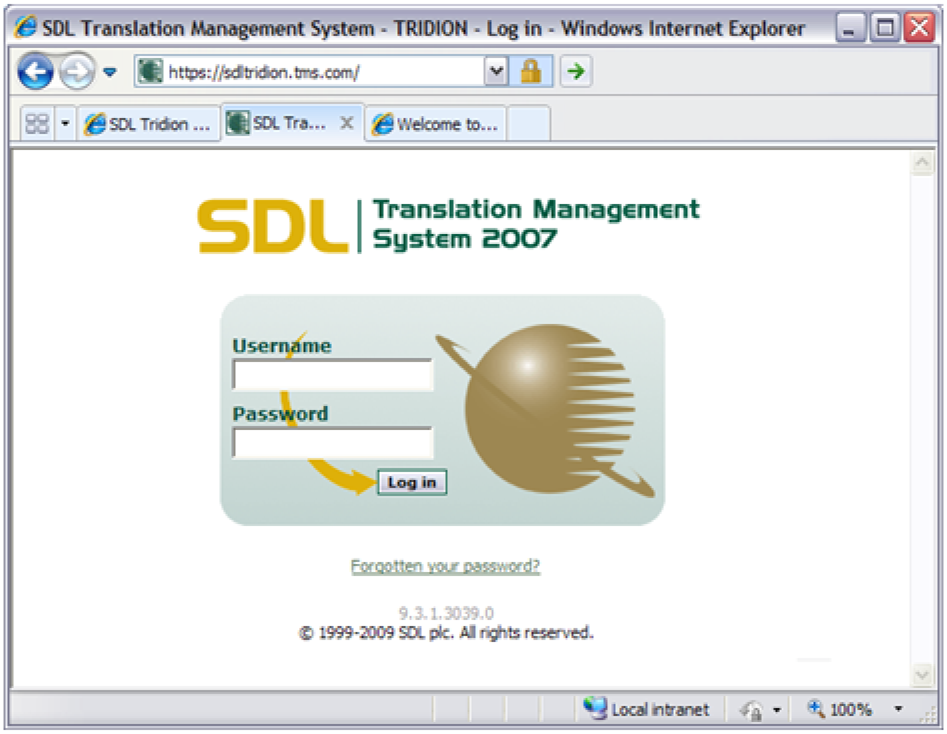
2. In the `<TmsServer>` element, specify connection details to the translation management system you are using, for example:

   ```
   <TmsServer port="443" compression="true" ssl="true">https://
   sdltridion.tms.com</TmsServer>
   ```

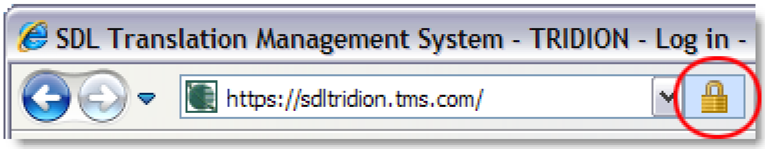
   - Specify the correct `port`
   - Set `ssl` to `true`
   - Specify the `https://` address of the translation management system server you are using

3. Save and close `TranslationManager.xml` and restart the Translation Manager Service.

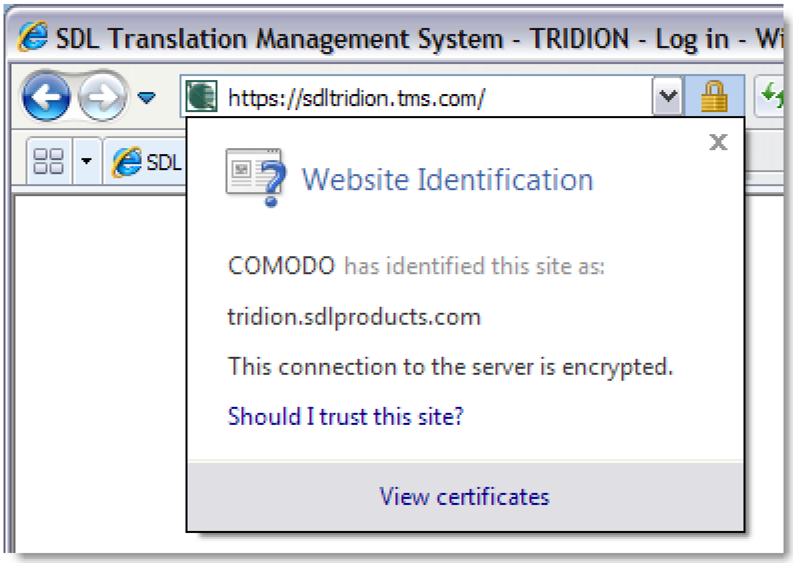
4. On each machine where you have installed Translation Manager Service, get a certificate by opening a Web browser—for example Internet Explorer—and enter the address (URL) of the translation management system server:


SDL Translation Management System - TRIDION - Log in - Windows Internet Explorer
https://sdltridion.tms.com/
SDL Tridion ...
SDL Tra...
Welcome to...
SDL
Translation Management System 2007
Username
Password
Log in
Forgotten your password?
9.3.1.3039.0
© 1999-2009 SDL plc. All rights reserved.
Local intranet
100%

5. Click the **Security Report** icon:


SDL Translation Management System - TRIDION - Log in -
https://sdltridion.tms.com/

6. Click **View certificates**:


SDL Translation Management System - TRIDION - Log in - Wi
https://sdltridion.tms.com/
SDL

Website Identification

COMODO has identified this site as:

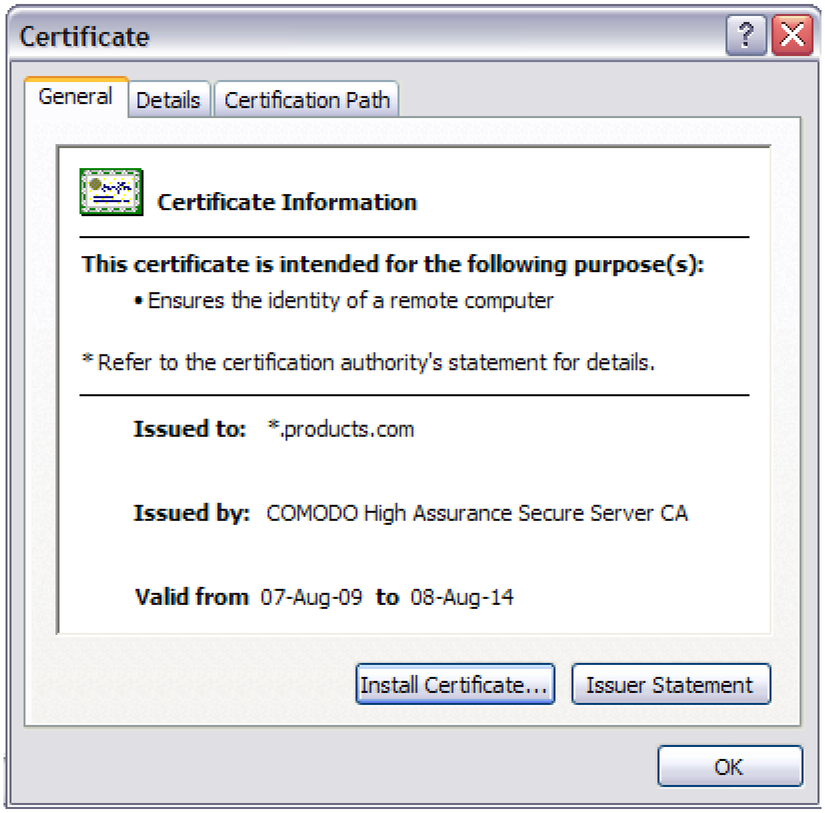
tridion.sdlproducts.com

This connection to the server is encrypted.

Should I trust this site?

View certificates

7. Click **Install Certificate**:



Certificate Information

This certificate is intended for the following purpose(s):

- Ensures the identity of a remote computer

* Refer to the certification authority's statement for details.

Issued to: *.products.com

Issued by: COMODO High Assurance Secure Server CA

Valid from 07-Aug-09 **to** 08-Aug-14

8. In the **Certificate Import Wizard**, follow the instructions to install a certificate.

13.5.5 Uninstalling Translation Manager

Uninstall Translation Manager by running the installer in **Modify** mode.

Steps to execute

1. Access the SDL Tridion installation media.
2. Navigate to the `Content Manager\` directory.
3. Double-click the executable `SDLTridion2013CM.exe`.
4. In the **Application Maintenance** screen, select **Modify**.
5. Deselect the features you want to uninstall and click **Next**.

   Uninstall Translation Manager using the installer provided.

   (You can also select features you want to install, in which case you may be prompted to provide additional information.)
6. Click **Uninstall** to confirm that you want to uninstall features.
7. Click **Finish** to exit the installer.

Result

The Translation Manager and its components are uninstalled.

13.6 Outscaling Translation Manager Service

Translation Manager supports the outscaling of the Translation Manager Service. Outscaling the Translation Manager Service ensures that when a Translation Manager Service fails while sending Translation Jobs or retrieving translated Content manager items, another Translation Manager Service can pick it up and continue processing the translation.

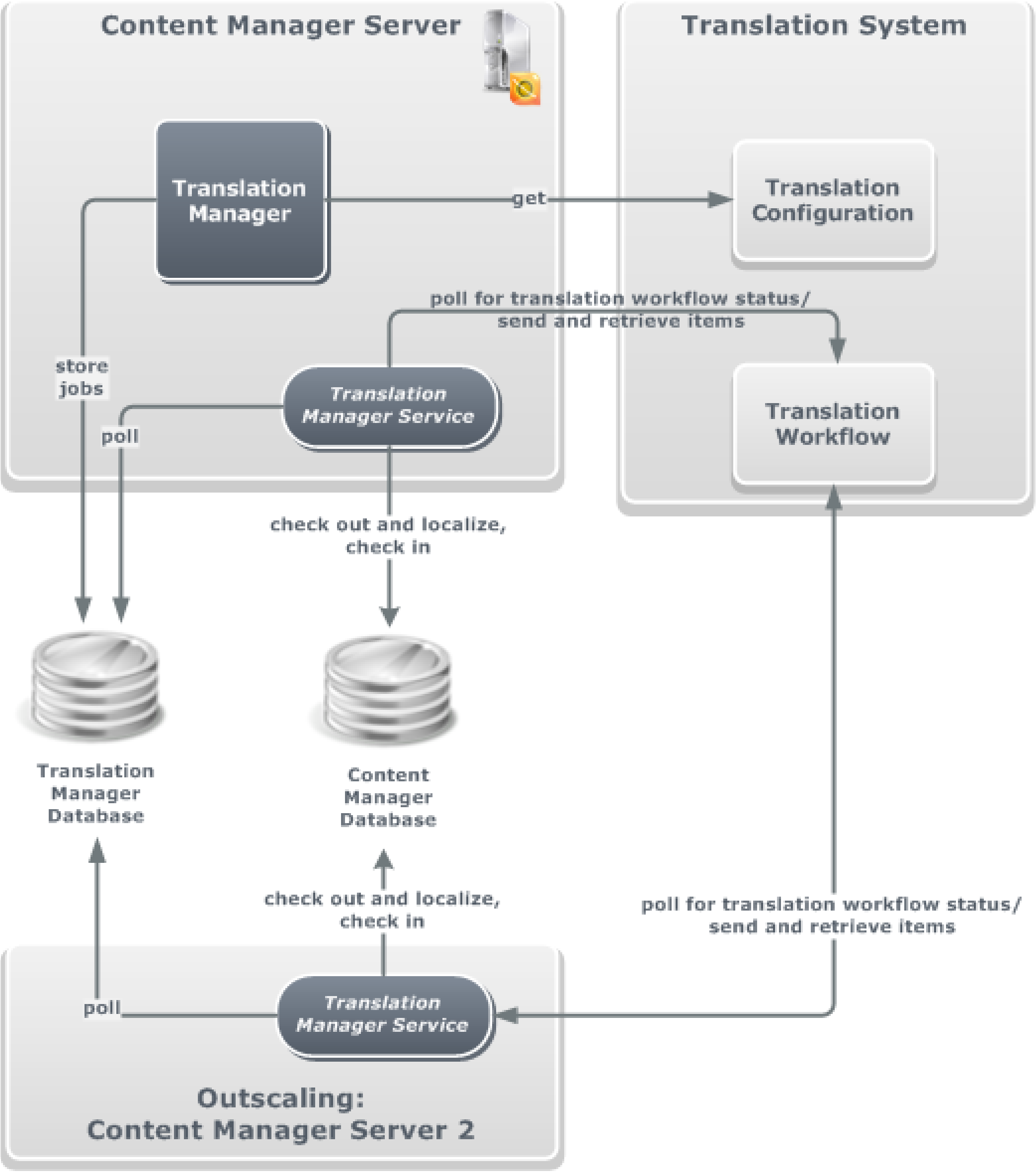
The Translation Manager Service synchronizes the translation management and Content Manager systems. It reads Translation Jobs from the Translation Manager database and performs the actions necessary to handle the jobs such as localizing and checking items out, creating jobs for transferral to the translation management system, copying over and checking items in when translations have been completed, and updating the status of jobs.

13.6.1 Translation Manager outscaling architecture

You can outscale the Translation Manager Service by installing it on a separate machine and configuring it to use the same Content Manager Database and Translation Manager Database.

The following describes the basic outscale architecture of Translation Manager:

- **Server 1**:
  - Perform a full installation of Content Manager Server
  - Perform a full installation of Translation Manager
- **Server 2**:
  - Install Content Manager Server core
  - Install Translation Manager Service
- **Server 3**: to outscale further, install Content Manager Server core and Translation Manager Service.


Content Manager Server
Translation System
Translation Manager
get
Translation Configuration
poll for translation workflow status/ send and retrieve items
store jobs
Translation Manager Service
poll
Translation Workflow
check out and localize, check in
Translation Manager Database
Content Manager Database
check out and localize, check in
poll for translation workflow status/ send and retrieve items
poll
Translation Manager Service
Outscaling: Content Manager Server 2

When you install multiple Translation Manager Services on multiple systems, the point of contact is the Translation Manager Database. The following describes how Translation Jobs are handled when you have multiple Translation Manager Services running:

- Each Translation Manager Service polls the Translation Manager Database at configured intervals for Translation Jobs awaiting transferral to the translation management system, or translated Content Manager items marked as awaiting retrieval from the translation management system. The `<TcmPollingInterval>` configuration element specifies the polling interval.
- When the status of a Translation Job is set to **Ready for Translation**, the first Translation Manager Services that subsequently polls the Translation Manager Database picks it up, processes it, then transfers it to the translation management system.

- Each Translation Manager Service polls translation management system at configured intervals for items awaiting retrieval from translation management system. The `<TmsPollingInterval>` configuration element specifies the polling interval.
- When a Translation Manager Service sees that items have been translated and are ready to be returned, it marks them as ready for retrieval in the Translation Manager Database.
- When an item has been marked for retrieval in the Translation Manager Database, any Translation Manager Service can retrieve that item.
- When sending or retrieving items to the translation management system, a timestamp is applied to an item to indicate when it was picked up by a service so that if a Translation Manager Service fails while sending or retrieving items to translation management system, after 2 hours has elapsed another Translation Manager Service can pick it up. If a Translation Manager Service is restarted before the 2 hour period has elapsed, it will continue processing the sending of Translation Jobs or retrieving of Content Manager items it was busy with when it failed.

13.6.2 Installing the Translation Manager Service on multiple machines

Outscale the Translation Manager Service by installing it on a separate machine and configuring it to use the same Content Manager Database and Translation Manager Database.

Requirements

Install the Translation Manager Service and Content Manager core on a supported Microsoft Windows operating systems. For more information, see Content Manager server prerequisites (see page 25).

Steps to execute

1. Create the **Content Manager Database** using the appropriate PowerShell script.

2. **Create Translation Manager Database**—run database scripts provided to create the Microsoft SQL Server or Oracle database where the Translation Manager stores Translation Jobs.

3. On machine 1, perform a full installation of the following:

    - **Content Manager Server**
    - **Translation Manager**

4. On machine 2, install the following components (these are the minimum components required to run the Translation Manager Service):

    - **Content Manager core**
    - **Translation Manager Service**—run the installer and install the **Translation Manager Service** (deselect other Components).

5. On machines 1 and 2, open the `TranslationManager.xml` configuration file, by default located in your Tridion installation `\config` directory, and set the following configuration options:

    - `SmtpServer`—the location and connection details of the SMTP server which is used to send notification e-mails

- `TcmPollingInterval` the frequency with which the Translation Manager is polled for Translation Jobs waiting transferal to the translation management system.
- `TmsPollingInterval` the frequency with which the translation management system is polled to see whether Translation Jobs have been completed.

Note: When you install the Translation Manager, the installer prompts you to set the mandatory configuration options.

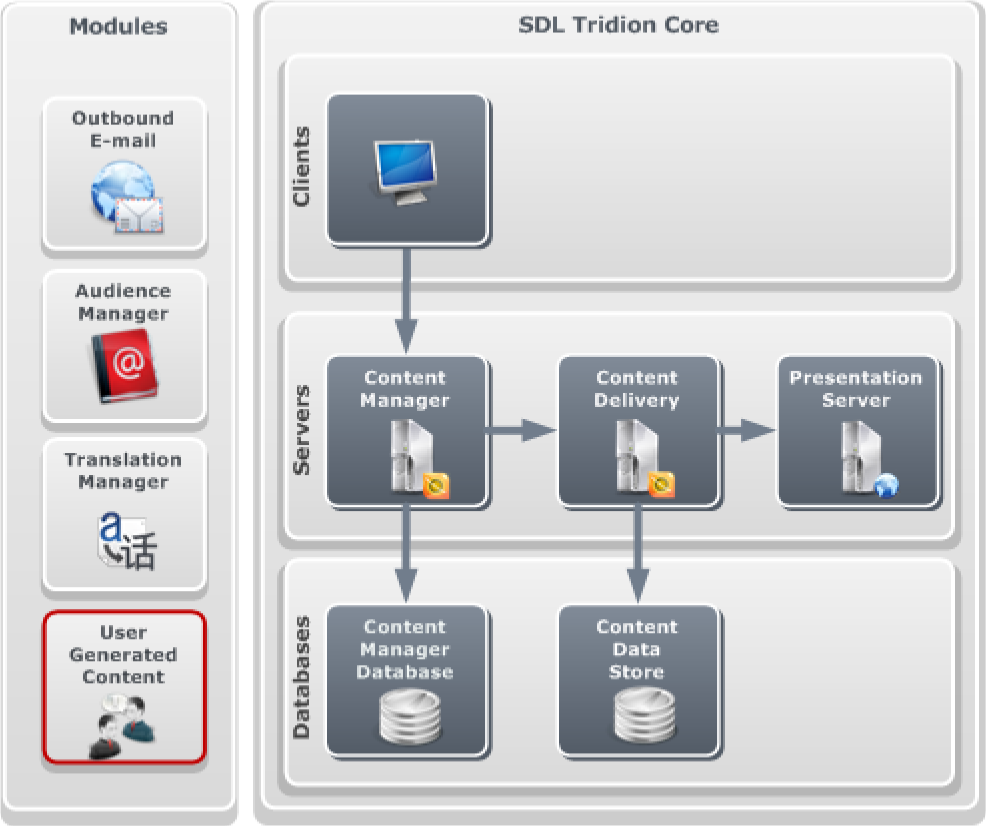
6. On machine 1, open the `TranslationManager.xml` configuration file, by default located in your Tridion installation `\config` directory, and set the various configuration options—for more information, see the Translation Manager documentation on the implementor's documentation portal.

7. Save and close the `TranslationManager.xml` configuration file.

8. Restart the Translation Manager Service to ensure your changes are enforced.

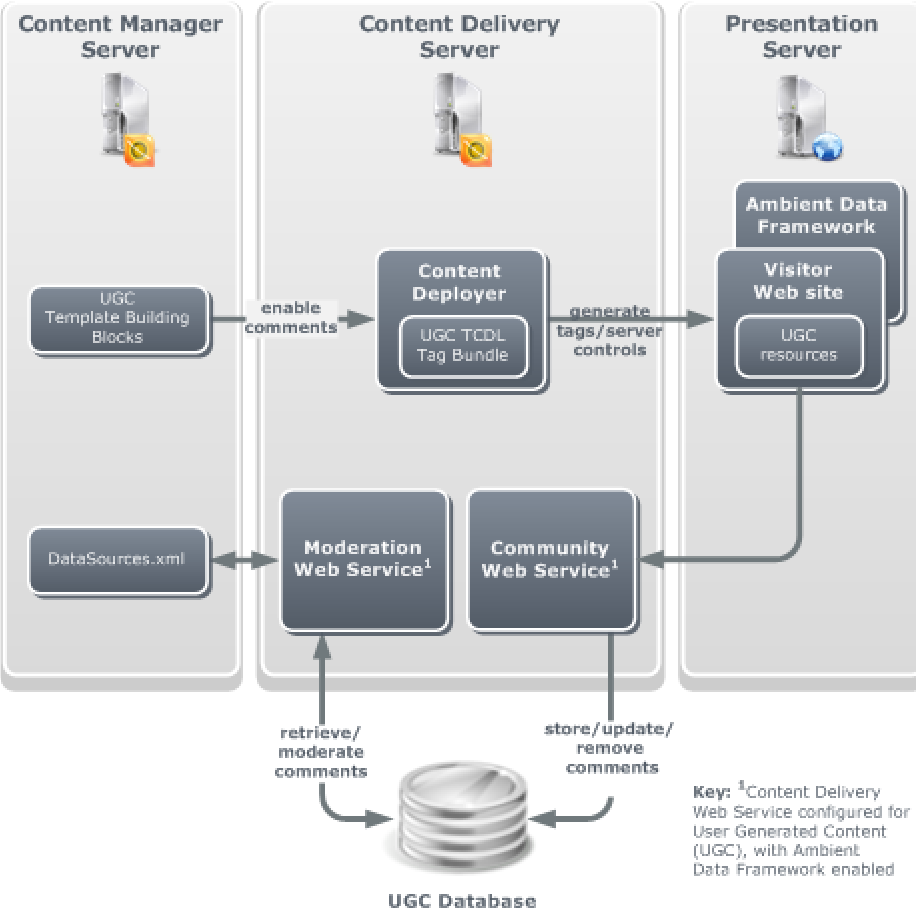
Chapter 14 Installing UGC

User Generated Content functionality allows visitors to your Web site to rate your content and leave comments and then manage these ratings and comments. User Generated Content is a module of User Generated Content. This section describes how to install User Generated Content and get it up and running.



14.1 UGC system diagram

User Generated Content consists of software components installed and configured on the Content Manager, Content Deployer, Content Delivery Web service, and on your visitor Web site. User Generated Content also requires you to create a database.



Content Deployer

Add a tag bundle to your Content Deployer so that it can process User Generated Content tags in published content.

User Generated Content database

Create a User Generated Content database to store comments and ratings.

Community Web Service

Create an instance of the Content Delivery Web Service to enable storage and retrieval of moderated comments and ratings with Ambient Data Framework set up and configured.

Moderation Web Service

Create an instance of the Content Delivery Web Service to enable storage and retrieval of submitted (visitor) comments and ratings with Ambient Data Framework set up and configured.

Visitor Web site on Presentation Server

Add library files and configurations to the (JSP or .NET) Web site to enable users to leave comments and rate content.

Content Manager

Set up and configure User Generated Content on the Content Manager server to enable implementers to add commenting and rating functionality to templates and to manage comments and ratings that Web site visitors submit.

14.2 Installing the UGC extension for Content Deployer

To add User Generated Content to your Content Deployer, add the `ugc_tcdl.jar` file to your Content Deployer installation and configure the `cd_deployer_conf.xml` configuration file.

Steps to execute

1. Go to the location where you have your Content Deployer running.
2. Open the Content Deployer configuration file, `cd_deployer_conf.xml` in a text editor.
3. Add the following element just above the closing tag `</TCDLEngine>`:

   ```
   <TagBundle Resource="ugc_tag_bundle.xml" />
   ```

4. Add User Generated Content functionality to the Content Deployer:

   a. Access the SDL Tridion installation media.

   b. Navigate to the `Content Delivery\roles\ugc\java\lib\` folder.

   c. Copy the file `ugc_tcdl.jar` to the following Content Deployer directory:

      - `lib` subdirectory (if installed as a Java process or Windows service)
      - `WEB-INF/lib/` subdirectory (if installed as a Java Web application)
      - `bin\lib\` subfolder (if installed as a .NET Web application)

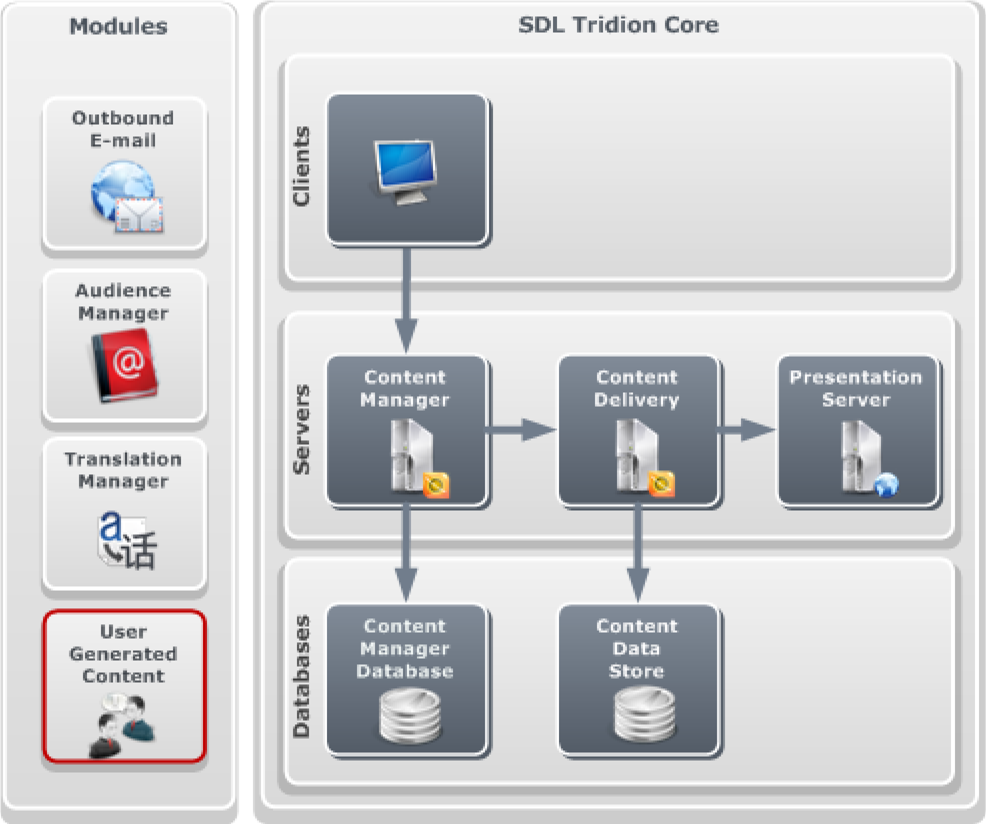
5. Restart the Content Deployer to apply the changes.

14.3 Creating the UGC database

User Generated Content needs a database (Oracle or SQL Server) to store ratings and comments.

14.3.1 User Generated Content prerequisites

This section lists the required and supported software for the User Generated Content database.



Content Manager side prerequisites

User Generated Content is fully integrated into the Content Manager and is installed as a part of the Content Manager installation, therefore Content Manager server prerequisites apply.

Content Delivery side prerequisites

User Generated Content is installed as a Server Role on the Content Delivery side of your installation, therefore Content Delivery prerequisites apply.

Database prerequisites on Content Delivery side

User Generated Content requires a database to store comments and ratings. The User Generated Content database supports the same Microsoft SQL Server and Oracle databases as the Content Data Store. **IBM DB2 database is not supported.**

The databases may run on any operating system that supports the specific DBMS you choose to use. Refer to your database vendor documentation to learn which operating systems are supported.

14.3.2 Creating the UGC database

You can use a PowerShell script to create a Microsoft SQL Server or Oracle database for UGC.

Requirements

To install SDL Tridion databases on a Windows machine requires Windows PowerShell 3.0 and Microsoft .NET Framework 4.0 or higher.

Steps to execute

1. Open Windows PowerShell from the Windows Start Menu.
2. In PowerShell, depending on your database navigate to one of the following folders on the installation media:
   - `Database\MSSQL\` or
   - `Database\Oracle\`
3. Type the following command to install the UGC database:

   ```
   & '.\Install User Generated Content database.ps1'
   ```

4. Follow the instructions in the PowerShell console to create the database.

Result

You have created the User Generated Content database.

14.4 Installing UGC in a .NET environment

This section explains how to install the Community Web service, Moderation Web service and Web site extension for User Generated Content in a .NET environment.

14.4.1 Installing the Community Web Service as a .NET Web application

User Generated Content requires you to install and configure two Web applications, one of which is used for adding comments (the Community Web Service). Note that this Web application also contains the Content Delivery Web service Server Role.

Steps to execute

1. Ensure that all the prerequisites for a Content Delivery Server Role are met on your target system.
2. If this is a 32-bit .NET Web application running in combination with Java 6, then in the advanced system properties of your machine, select **Environment Variables** to see and edit your environment variables, and ensure that your `PATH` variable contains the `bin\` subfolder of the Java instance you are using. This location contains the file `MSVCR71.DLL`.
3. On your Windows machine, start up the Server Manager and do one of the following:
   - If this Web application runs under Windows 2012, start up the **Add Roles and Features Wizard**. Step through the wizard until you reach the **Features** screen. Now do one of the following:

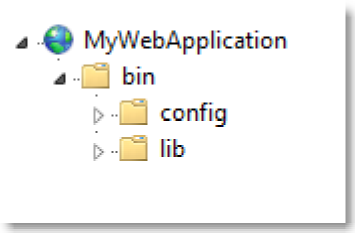
    - If you use .NET 4.5, expand the item **.NET Framework 4.5 Features** and the subitem **WCF Services**.
    - Alternatively, if you use .NET 3.5, expand the item **.NET Framework 3.5 Features**.
  - Alternatively, if this Web application runs under Windows 2008 R2 SP1, under **Features Summary**, select **Add Features**. In the **Add Features** dialog that opens, expand the item **.NET Framework 3.5.1 Features** and the subitem **WCF Activation**.

  If the option **HTTP Activation** is not listed as installed, select it. Then click **Next** and **Install** to install the feature.

4. Create the following Web application directory structure:

   a. In the root folder, create a `bin\` subfolder.

   b. In the `bin\` folder, create subfolders `config\` and `lib\`.



5. Access the SDL Tridion installation media and navigate to the `Content Delivery\roles\` folder.

6. Access the `ugc\dotNET` subfolder.

7. Copy all DLLs in this folder to the `bin\` folder of your Web application.

8. Do one of the following:

   - To install a 32-bit .NET Web application, access the `x86\` subfolder.
   - To install a 64-bit .NET Web application, access the `x86_64\` subfolder.

9. Copy all DLLs in this folder to the `bin\` folder of your Web application.

10. On the installation media, navigate to the `Content Delivery\roles\ugc\java\` folder.

11. Copy all the JAR files from the `lib\` subfolder, and all the JAR files from the `third-party-lib\` subfolder, to the `bin\lib\` folder of your Web application.

12. If you store published content in a database, download the JDBC driver (JAR file) required for your specific database vendor and copy it into the `bin\lib` folder of your Web application:

Database vendor	JDBC Driver	File to download
Microsoft SQL Server	Microsoft JDBC Driver 4.0 for SQL Server	`sqljdbc4.jar`
Oracle	Oracle JDBC driver 11.2.0.3	`ojdbc6.jar`

13. Navigate to `\Content Delivery\resources\configurations\`.

14. Copy `logback.xml` from that folder to the `bin\config` folder of your Web application.

15. Copy and rename the following configuration files to the `bin\config` folder of your Web application:

Corresponding configuration file	Renamed configuration file
`cd_ambient_conf_sample.xml`	`cd_ambient_conf.xml`
`cd_dynamic_conf_sample.xml`	`cd_dynamic_conf.xml`
`cd_storage_conf_sample.xml`	`cd_storage_conf.xml`
`cd_link_conf_sample.xml`	`cd_link_conf.xml`
`cd_ugc_conf_sample.xml`	`cd_ugc_conf.xml`
`cd_webservice_conf_sample.xml`	`cd_webservice_conf.xml`

16. Open `cd_ambient_conf.xml` for editing.

17. Remove the comment tags around the following elements:

    `WhiteList`

    This element contains an `IPAddresses` subelement which contains one or more `Ip` elements. Ensure that `IPAddresses` contains only one `Ip` subelement, and set its value to the IP address of your Presentation Server.

    `GloballyAcceptedClaims`

    Uncomment this element and its subelement `Claim` with its `Uri` set to the value `taf:tracking:id`.

    These elements specify which forwarded Claims, coming from which IP address, are accepted by the Community Web service.

18. Save and close `cd_ambient_conf.xml`.

19. Copy `cd_licenses.xml`, your Content Delivery license file available from SDL Tridion Customer Support, to the `bin\config` directory of your Web application.

    If you want to name the file differently or put it in a different location, you need to update the `License` element in your configuration files.

20. Configure logging.

21. Configure the Storage Layer and optionally encrypt sensitive strings.

22. Open `Web.config` for editing in the root folder of this Web application.

23. To enable the Ambient Data Framework, do the following:

    Application Pool in Classic Mode

    Add the following inside the `<httpModules>` inside the `<system.web>` section (create an `<httpModules>` element if it does not yet exist):

```
<add type="Tridion.ContentDelivery.AmbientData.HttpModule" name="AmbientFrameworkModule" />
```

    Application Pool in Integrated Mode

    Add the following inside the `<modules>` inside the `<system.webServer>` section (create a `<modules>` element if it does not yet exist):

```
<add type="Tridion.ContentDelivery.AmbientData.HttpModule" name="AmbientFrameworkModule"
 preCondition="managedHandler" />
```

24. Save and close `Web.config`.

25. Disable recycling of the Application Pool of this Web application by doing the following:

    a. Open IIS Manager.

    b. From the tree on the left, select the current machine and navigate to the Web application you just created.

    c. Click **Advanced settings** to open a dialog with the application's properties, and note down the value of **Application Pool**.

    d. Close the dialog.

    e. From the tree on the left, under the current machine, now select the **Application Pools** node.

    f. From the list on the right, select the Application Pool you just noted down.

    g. Select **Recycling** in the **Edit Application Pool** area on the right.

    h. In the dialog that opes, deselect **Regular time intervals**, then click **Next** and **Finish** to commit your change.

    i. Restart IIS.

26. In IIS Manager, navigate to the Content Delivery Web service .NET Web application and select it. In the **IIS** area in the content area, double-click **Modules** to see a list of installed Managed Modules. If you see an item in this list called **WebDAVModule**, select it and click **Remove** on the right hand side of the screen to remove it.

27. Close IIS Manager.

28. Restart IIS.

14.4.2 Installing the Moderation Web Service as a .NET Web application

User Generated Content requires you to install and configure two Web applications, one of which is used for moderating comments (the Moderation Web Service). Note that this Web application also contains the Content Delivery Web service Server Role.

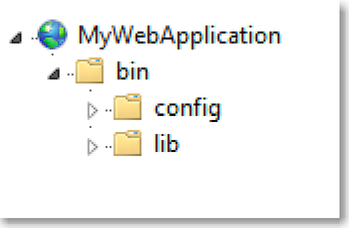
Steps to execute

1. Ensure that all the prerequisites for a Content Delivery Server Role are met on your target system.
2. If this is a 32-bit .NET Web application running in combination with Java 6, then in the advanced system properties of your machine, select **Environment Variables** to see and edit your environment variables, and ensure that your `PATH` variable contains the `bin\` subfolder of the Java instance you are using. This location contains the file `MSVCR71.DLL`.
3. On your Windows machine, start up the Server Manager and do one of the following:
   - If this Web application runs under Windows 2012, start up the **Add Roles and Features Wizard**. Step through the wizard until you reach the **Features** screen. Now do one of the following:
     - If you use .NET 4.5, expand the item **.NET Framework 4.5 Features** and the subitem **WCF Services**.
     - Alternatively, if you use .NET 3.5, expand the item **.NET Framework 3.5 Features**.
   - Alternatively, if this Web application runs under Windows 2008 R2 SP1, under **Features Summary**, select **Add Features**. In the **Add Features** dialog that opens, expand the item **.NET Framework 3.5.1 Features** and the subitem **WCF Activation**.

   If the option **HTTP Activation** is not listed as installed, select it. Then click **Next** and **Install** to install the feature.
4. Create the following Web application directory structure:

   a. In the root folder, create a `bin\` subfolder.

   b. In the `bin\` folder, create subfolders `config\` and `lib\`.



5. Access the SDL Tridion installation media and navigate to the `Content Delivery\roles\` folder.

6. Access the `ugc\dotNET` subfolder.

7. Copy all DLLs in this folder to the `bin\` folder of your Web application.

8. Do one of the following:
    - To install a 32-bit .NET Web application, access the `x86\` subfolder.
    - To install a 64-bit .NET Web application, access the `x86_64\` subfolder.

9. Copy all DLLs in this folder to the `bin\` folder of your Web application.

10. On the installation media, navigate to the `Content Delivery\roles\ugc\java\` folder.

11. Copy all the JAR files from the `lib\` subfolder, and all the JAR files from the `third-party-lib\` subfolder, to the `bin\lib\` folder of your Web application.

12. If you store published content in a database, download the JDBC driver (JAR file) required for your specific database vendor and copy it into the `bin\lib` folder of your Web application:

Database vendor	JDBC Driver	File to download
Microsoft SQL Server	Microsoft JDBC Driver 4.0 for SQL Server	`sqljdbc4.jar`
Oracle	Oracle JDBC driver 11.2.0.3	`ojdbc6.jar`

13. Navigate to `\Content Delivery\resources\configurations\`.

14. Copy `logback.xml` from that folder to the `bin\config` folder of your Web application.

15. Copy and rename the following configuration files to the `bin\config` folder of your Web application:

Corresponding configuration file	Renamed configuration file
`cd_ambient_conf_sample.xml`	`cd_ambient_conf.xml`
`cd_dynamic_conf_sample.xml`	`cd_dynamic_conf.xml`
`cd_storage_conf_sample.xml`	`cd_storage_conf.xml`
`cd_link_conf_sample.xml`	`cd_link_conf.xml`
`cd_ugc_conf_sample.xml`	`cd_ugc_conf.xml`
`cd_webservice_conf_sample.xml`	`cd_webservice_conf.xml`

16. Open `cd_ambient_conf.xml` for editing.

17. Find the `<Security>` section and enclose it in comments. This element is only needed if you intend to set up OAuth authentication.

18. Save and close `cd_ambient_conf.xml`.

19. Copy `cd_licenses.xml`, your Content Delivery license file available from SDL Tridion Customer Support, to the `bin\config` directory of your Web application.

    If you want to name the file differently or put it in a different location, you need to update the `License` element in your configuration files.

20. Configure logging.

21. Configure the Storage Layer and optionally encrypt sensitive strings.

22. Open `Web.config` for editing in the root folder of this Web application.

23. To enable the Ambient Data Framework, do the following:

    Application Pool in Classic Mode

    Add the following inside the `<httpModules>` inside the `<system.web>` section (create an `<httpModules>` element if it does not yet exist):

```
<add type="Tridion.ContentDelivery.AmbientData.HttpModule" name="AmbientFrameworkModule" />
```

    Application Pool in Integrated Mode

    Add the following inside the `<modules>` inside the `<system.webServer>` section (create a `<modules>` element if it does not yet exist):

```
<add type="Tridion.ContentDelivery.AmbientData.HttpModule" name="AmbientFrameworkModule"
 preCondition="managedHandler" />
```

24. Save and close `Web.config`.

25. Disable recycling of the Application Pool of this Web application by doing the following:

    a. Open IIS Manager.

    b. From the tree on the left, select the current machine and navigate to the Web application you just created.

    c. Click **Advanced settings** to open a dialog with the application's properties, and note down the value of **Application Pool**.

    d. Close the dialog.

    e. From the tree on the left, under the current machine, now select the **Application Pools** node.

    f. From the list on the right, select the Application Pool you just noted down.

    g. Select **Recycling** in the **Edit Application Pool** area on the right.

    h. In the dialog that opes, deselect **Regular time intervals**, then click **Next** and **Finish** to commit your change.

    i. Restart IIS.

26. In IIS Manager, navigate to the Content Delivery Web service .NET Web application and select it. In the **IIS** area in the content area, double-click **Modules** to see a list of installed Managed Modules. If you see an item in this list called **WebDAVModule**, select it and click **Remove** on the right hand side of the screen to remove it.

27. Close IIS Manager.

28. Restart IIS.

14.4.3 Adding the UGC Web site extension to a .NET Web site

Install the User Generated Content Web site extension by adding a .NET Web application to the live Web site.

Steps to execute

1. Ensure that all the prerequisites for a Content Delivery Server Role are met on your target system.

2. If this is a 32-bit .NET Web application running in combination with Java 6, then in the advanced system properties of your machine, select **Environment Variables** to see and edit your environment variables, and ensure that your `PATH` variable contains the `bin\` subfolder of the Java instance you are using. This location contains the file `MSVCR71.DLL`.

3. On your Windows machine, start up the Server Manager and do one of the following:

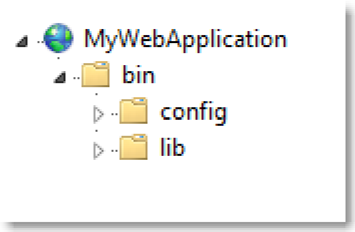
   - If this Web application runs under Windows 2012, start up the **Add Roles and Features Wizard**. Step through the wizard until you reach the **Features** screen. Now do one of the following:
     - If you use .NET 4.5, expand the item **.NET Framework 4.5 Features** and the subitem **WCF Services**.
     - Alternatively, if you use .NET 3.5, expand the item **.NET Framework 3.5 Features**.
   - Alternatively, if this Web application runs under Windows 2008 R2 SP1, under **Features Summary**, select **Add Features**. In the **Add Features** dialog that opens, expand the item **.NET Framework 3.5.1 Features** and the subitem **WCF Activation**.

   If the option **HTTP Activation** is not listed as installed, select it. Then click **Next** and **Install** to install the feature.

4. Create the following Web application directory structure:

   a. In the root folder, create a `bin\` subfolder.

   b. In the `bin\` folder, create subfolders `config\` and `lib\`.



5. Access the SDL Tridion installation media and navigate to the `Content Delivery\roles\` folder.

6. Access the `ugc\dotNET` subfolder.

7. Copy all DLLs in this folder to the `bin\` folder of your Web application.

8. Do one of the following:
   - To install a 32-bit .NET Web application, access the `x86\` subfolder.
   - To install a 64-bit .NET Web application, access the `x86_64\` subfolder.

9. Copy all DLLs in this folder to the `bin\` folder of your Web application.

10. On the installation media, navigate to the `Content Delivery\roles\ugc\java\` folder.

11. Copy all the JAR files from the `lib\` subfolder, and all the JAR files from the `third-party-lib\` subfolder, to the `bin\lib\` folder of your Web application.

12. If you store published content in a database, download the JDBC driver (JAR file) required for your specific database vendor and copy it into the `bin\lib` folder of your Web application:

Database vendor	JDBC Driver	File to download
Microsoft SQL Server	Microsoft JDBC Driver 4.0 for SQL Server	`sqljdbc4.jar`
Oracle	Oracle JDBC driver 11.2.0.3	`ojdbc6.jar`

13. Navigate to `\Content Delivery\resources\configurations\`.

14. Copy `logback.xml` from that folder to the `bin\config` folder of your Web application.

15. Copy and rename the following configuration files to the `bin\config` folder of your Web application:

Corresponding configuration file	Renamed configuration file
`cd_ambient_conf_sample.xml`	`cd_ambient_conf.xml`
`cd_dynamic_conf_sample.xml`	`cd_dynamic_conf.xml`

Corresponding configuration file	Renamed configuration file
`cd_storage_conf_sample.xml`	`cd_storage_conf.xml`
`cd_link_conf_sample.xml`	`cd_link_conf.xml`
`cd_ugc_conf_sample.xml`	`cd_ugc_conf.xml`
`cd_wai_conf_sample.xml`	`cd_wai_conf.xml`

16. Open the file `cd_dynamic_conf.xml` for editing.

17. In the `<TCDLEngine>` section, insert the following element just before the end tag:

    ```
    <TagBundle Resource="ugc_renderer_bundle.xml"/>
    ```

18. Save and close `cd_dynamic_conf.xml`.

19. Open `cd_ambient_conf.xml` for editing (if using).

20. Find the `<Security>` section and enclose it in comments. This element is only needed if you intend to set up OAuth authentication.

21. Save and close `cd_ambient_conf.xml`.

22. Copy `cd_licenses.xml`, your Content Delivery license file available from SDL Tridion Customer Support, to the `bin\config` directory of your Web application.

    If you want to name the file differently or put it in a different location, you need to update the `License` element in your configuration files.

23. Configure logging.

24. Configure the Storage Layer and optionally encrypt sensitive strings.

25. Disable recycling of the Application Pool of this Web application by doing the following:

    a. Open IIS Manager.

    b. From the tree on the left, select the current machine and navigate to the Web application you just created.

    c. Click **Advanced settings** to open a dialog with the application's properties, and note down the value of **Application Pool**.

    d. Close the dialog.

    e. From the tree on the left, under the current machine, now select the **Application Pools** node.

    f. From the list on the right, select the Application Pool you just noted down.

    g. Select **Recycling** in the **Edit Application Pool** area on the right.

h. In the dialog that opes, deselect **Regular time intervals**, then click **Next** and **Finish** to commit your change.

i. Restart IIS.

26. Restart the Web application.

27. Open `Web.config` for editing in the root folder of this Web application.

28. To enable the Ambient Data Framework, do the following:

Application Pool in Classic Mode

Add the following inside the `<httpModules>` inside the `<system.web>` section (create an `<httpModules>` element if it does not yet exist):

```
<add type="Tridion.ContentDelivery.AmbientData.HttpModule" name="AmbientFrameworkModule" />
```

Application Pool in Integrated Mode

Add the following inside the `<modules>` inside the `<system.webServer>` section (create a `<modules>` element if it does not yet exist):

```
<add type="Tridion.ContentDelivery.AmbientData.HttpModule" name="AmbientFrameworkModule"
 preCondition="managedHandler" />
```

29. Add a new `<appSettings>` node with the following contents:

```
<appSettings>
  <add key="ODataEndPoint.URL" value="http://mysite/ws/odata.svc/" />
  <add key="ODataEndPoint.ConnectionTimeout" value="20000" />
  <add key="ODataEndPoint.ReadTimeout" value="20000" />
</appSettings>
```

in which you make the following replacements:

- Change `http://mysite/ws/odata.svc/` into the URL of your Content Delivery Web service
- Change either of the timeout values to a different number of milliseconds if you expect this 20 second default to be inadequate for your implementation.

30. Add the UGC tag prefix to `web.config` by adding the following element in the `<controls>` section, located inside the `<pages>` section:

```
<add tagPrefix="ugc" namespace="Tridion.ContentDelivery.UGC.Web.UI"
assembly="Tridion.ContentDelivery.UGC" />
```

31. Save and close `Web.config`.

32. If you registered Server Controls, change all relevant Page Templates to add a Page directive at the top of every published Web page (`.ascx` pages):

```
<%@ Page Language="C#" %>
```

14.5 Installing UGC in a JSP environment

This section explains how to install the Community Web service, Moderation Web service and Web site extension for User Generated Content in a Java/JSP environment.

14.5.1 Installing the Community Web Service as a Java Web application

User Generated Content requires you to install and configure two Web applications, one of which is used for adding comments (the Community Web Service). Note that this Web application also contains the Content Delivery Web service Server Role.

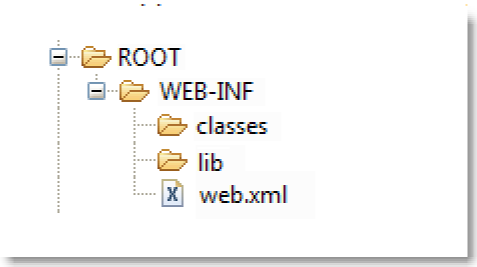
Steps to execute

1. Ensure that all the prerequisites for a Content Delivery Server Role are met on your target system.

2. Create the following Web application directory structure:

   a. In the root folder, create a `WEB-INF/` directory.

   b. In the `WEB-INF/` directory, create subdirectories `classes/` and `lib/`.



3. On the installation media, navigate to the `Content Delivery\roles\ugc\java\` folder.

4. Copy all the JAR files from the `lib\` subfolder, and all the JAR files from the `third-party-lib\` subfolder, to the `WEB-INF/lib/` folder of your Web application.

5. If you store published content in a database, download the JDBC driver (JAR file) required for your specific database vendor and copy it into the `WEB-INF/lib` directory of your Web application:

Database vendor	JDBC Driver	File to download
Microsoft SQL Server	Microsoft JDBC Driver 4.0 for SQL Server	sqljdbc4.jar
Oracle	Oracle JDBC driver 11.2.0.3	ojdbc6.jar

6. Navigate to \Content Delivery\resources\configurations\.

7. Copy logback.xml from that folder to the WEB-INF/classes/ subdirectory of your Web application.

8. Copy and rename the following configuration files to the WEB-INF/classes/ folder of your Web application:

Corresponding configuration file	Renamed configuration file
cd_ambient_conf_sample.xml	cd_ambient_conf.xml
cd_dynamic_conf_sample.xml	cd_dynamic_conf.xml
cd_storage_conf_sample.xml	cd_storage_conf.xml
cd_link_conf_sample.xml	cd_link_conf.xml
cd_ugc_conf_sample.xml	cd_ugc_conf.xml
cd_webservice_conf_sample.xml	cd_webservice_conf.xml

9. Copy cd_licenses.xml, your Content Delivery license file available from SDL Tridion Customer Support, to the WEB-INF/classes directory of your Web application.

   If you want to name the file differently or put it in a different location, you need to update the License element in your configuration files.

10. Configure logging.

11. Configure the Storage Layer and optionally encrypt sensitive strings.

12. Open cd_ambient_conf.xml for editing.

13. Remove the comment tags around the following elements:

    WhiteList

    This element contains an IPAddresses subelement which contains one or more Ip elements. Ensure that IPAddresses contains only one Ip subelement, and set its value to the IP address of your Presentation Server.

    GloballyAcceptedClaims

    Uncomment this element and its subelement Claim with its Uri set to the value taf:tracking:id.

    These elements specify which forwarded Claims, coming from which IP address, are accepted by the Community Web service.

14. Save and close `cd_ambient_conf.xml`.

15. Enable the Ambient Data Framework by adding the following to your `web.xml`:

    ```
    <filter>
      <filter-name>Ambient Data Framework</filter-name>
      <filter-class>com.tridion.ambientdata.web.AmbientDataServletFilter</filter-class>
    </filter>
    <filter-mapping>
      <filter-name>Ambient Data Framework</filter-name>
      <servlet-name>Content Delivery Web service</servlet-name>
    </filter-mapping>
    ```

    This enables the Ambient Data Framework for the Content Delivery Web service only. If you want to enable the Ambient Data Framework in the entire Web application, replace the `servlet-name` element with the following:

    ```
    <url-pattern>/*</url-pattern>
    ```

16. If your Web application server is JBoss, do the following:

    a. Locate the following files in the `WEB-INF/lib/` subdirectory and remove them:

       - `stax-api.jar`
       - `serializer.jar`
       - `xalan.jar`

    b. If you use JBoss 5.1, add the following JVM property-value pair to the `JAVA_OPTS` variable: `-Dorg.jboss.logging.provider=slf4j`

    c. If you use JBoss 6, create a file with the following contents called `jboss-deployment-structure.xml` and save the file in the `WEB-INF/` directory:

       ```
       <jboss-deployment-structure>
         <deployment>
           <exclusions>
             <module name="org.slf4j" />
             <module name="org.slf4j.impl" />
             <module name="com.sun.jersey" />
             <module name="com.sun.jersey.json" />
             <module name="com.sun.jersey.core" />
           </exclusions>
         </deployment>
       </jboss-deployment-structure>
       ```

    d. If you use JBoss 6, open `WEB-INF/web.xml` file, add the following section and save the file:

       ```
       <context-param>
           <param-name>resteasy.scan</param-name>
           <param-value>false</param-value>
       </context-param>
       <context-param>
            <param-name>resteasy.scan.resources</param-name>
            <param-value>false</param-value>
       </context-param>
       <context-param>
             <param-name>resteasy.scan.providers</param-name>
             <param-value>false</param-value>
       </context-param>
       ```

17. If your Web application server is WebLogic, do the following:

a. Remove the file `stax-api.jar` from your `WEB-INF/lib/` subdirectory.

b. Add a `weblogic.xml` file to the `WEB-INF` folder of your Web application that forces WebLogic to use the libraries contained in the Web application, rather than its own libraries. The contents of this `weblogic.xml` file must be as follows:

```
<weblogic-web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.bea.com/ns/weblogic/weblogic-web-app http://www.bea.com/ns/weblogic/weblogic-
web-app/1.0/weblogic-web-app.xsd"
  xmlns="http://www.bea.com/ns/weblogic/weblogic-web-app">
  <container-descriptor>
    <prefer-application-packages>
      <package-name>javax.persistence.**</package-name>
      <package-name>antlr.*</package-name>
      <package-name>org.slf4j.**</package-name>
      <package-name>org.apache.log4j.**</package-name>
      <package-name>org.apache.xml.serializer.**</package-name>
    </prefer-application-packages>
  </container-descriptor>
</weblogic-web-app>
```

18. Navigate to `Content Delivery\roles\webservice\java\` and copy `web.xml` to the root of your Java Web application.

19. Open the `web.xml` file and uncomment the following sections:

    ```
    Sample configuration for the Web service
    ```

    ```
    OAuth 2.0 security access filter
    ```

    ```
    Uncomment the following block to apply the filter for the UGC user
    ```

20. Save and close `web.xml`.

21. Restart the Web application.

14.5.2 Installing the Moderation Web Service as a Java Web application

User Generated Content requires you to install and configure two Web applications, one of which is used for moderating comments (the Moderation Web Service). Note that this Web application also contains the Content Delivery Web service Server Role.

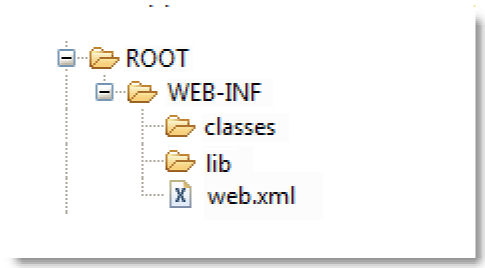
Steps to execute

1. Ensure that all the prerequisites for a Content Delivery Server Role are met on your target system.

2. Create the following Web application directory structure:

   a. In the root folder, create a `WEB-INF/` directory.

   b. In the `WEB-INF/` directory, create subdirectories `classes/` and `lib/`.



3. On the installation media, navigate to the `Content Delivery\roles\ugc\java\` folder.

4. Copy all the JAR files from the `lib\` subfolder, and all the JAR files from the `third-party-lib\` subfolder, to the `WEB-INF/lib/` folder of your Web application.

5. If you store published content in a database, download the JDBC driver (JAR file) required for your specific database vendor and copy it into the `WEB-INF/lib` directory of your Web application:

Database vendor	JDBC Driver	File to download
Microsoft SQL Server	Microsoft JDBC Driver 4.0 for SQL Server	`sqljdbc4.jar`
Oracle	Oracle JDBC driver 11.2.0.3	`ojdbc6.jar`

6. Navigate to `\Content Delivery\resources\configurations\`.

7. Copy `logback.xml` from that folder to the `WEB-INF/classes/` subdirectory of your Web application.

8. Copy and rename the following configuration files to the `WEB-INF/classes/` folder of your Web application:

Corresponding configuration file	Renamed configuration file
`cd_ambient_conf_sample.xml`	`cd_ambient_conf.xml`
`cd_dynamic_conf_sample.xml`	`cd_dynamic_conf.xml`
`cd_storage_conf_sample.xml`	`cd_storage_conf.xml`
`cd_link_conf_sample.xml`	`cd_link_conf.xml`
`cd_ugc_conf_sample.xml`	`cd_ugc_conf.xml`
`cd_webservice_conf_sample.xml`	`cd_webservice_conf.xml`

9. Open `cd_ambient_conf.xml` for editing.

10. Find the `<Security>` section and enclose it in comments. This element is only needed if you intend to set up OAuth authentication.

11. Save and close `cd_ambient_conf.xml`.

12. Copy `cd_licenses.xml`, your Content Delivery license file available from SDL Tridion Customer Support, to the `WEB-INF/classes` directory of your Web application.

    If you want to name the file differently or put it in a different location, you need to update the `License` element in your configuration files.

13. Configure logging.

14. Configure the Storage Layer and optionally encrypt sensitive strings.

15. Enable the Ambient Data Framework by adding the following to your `web.xml`:

    ```
    <filter>
      <filter-name>Ambient Data Framework</filter-name>
      <filter-class>com.tridion.ambientdata.web.AmbientDataServletFilter</filter-class>
    </filter>
    <filter-mapping>
      <filter-name>Ambient Data Framework</filter-name>
      <servlet-name>Content Delivery Web service</servlet-name>
    </filter-mapping>
    ```

    This enables the Ambient Data Framework for the Content Delivery Web service only. If you want to enable the Ambient Data Framework in the entire Web application, replace the `servlet-name` element with the following:

    ```
    <url-pattern>/*</url-pattern>
    ```

16. If your Web application server is JBoss, do the following:

    a. Locate the following files in the `WEB-INF/lib/` subdirectory and remove them:

       - `stax-api.jar`
       - `serializer.jar`
       - `xalan.jar`

    b. If you use JBoss 5.1, add the following JVM property-value pair to the `JAVA_OPTS` variable: `-Dorg.jboss.logging.provider=slf4j`

    c. If you use JBoss 6, create a file with the following contents called `jboss-deployment-structure.xml` and save the file in the `WEB-INF/` directory:

       ```
       <jboss-deployment-structure>
         <deployment>
           <exclusions>
             <module name="org.slf4j" />
             <module name="org.slf4j.impl" />
             <module name="com.sun.jersey" />
             <module name="com.sun.jersey.json" />
             <module name="com.sun.jersey.core" />
           </exclusions>
         </deployment>
       </jboss-deployment-structure>
       ```

    d. If you use JBoss 6, open `WEB-INF/web.xml` file, add the following section and save the file:

       ```
       <context-param>
           <param-name>resteasy.scan</param-name>
           <param-value>false</param-value>
       </context-param>
       <context-param>
            <param-name>resteasy.scan.resources</param-name>
            <param-value>false</param-value>
       </context-param>
       ```

```
<context-param>
      <param-name>resteasy.scan.providers</param-name>
      <param-value>false</param-value>
</context-param>
```

17. If your Web application server is WebLogic, do the following:

    a. Remove the file `stax-api.jar` from your `WEB-INF/lib/` subdirectory.

    b. Add a `weblogic.xml` file to the `WEB-INF` folder of your Web application that forces WebLogic to use the libraries contained in the Web application, rather than its own libraries. The contents of this `weblogic.xml` file must be as follows:

```
<weblogic-web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.bea.com/ns/weblogic/weblogic-web-app http://www.bea.com/ns/weblogic/weblogic-
web-app/1.0/weblogic-web-app.xsd"
  xmlns="http://www.bea.com/ns/weblogic/weblogic-web-app">
  <container-descriptor>
    <prefer-application-packages>
      <package-name>javax.persistence.**</package-name>
      <package-name>antlr.*</package-name>
      <package-name>org.slf4j.**</package-name>
      <package-name>org.apache.log4j.**</package-name>
      <package-name>org.apache.xml.serializer.**</package-name>
    </prefer-application-packages>
  </container-descriptor>
</weblogic-web-app>
```

18. Navigate to `Content Delivery\roles\webservice\java\` and copy `web.xml` to the root of your Java Web application.

19. Open the `web.xml` file and uncomment the following sections:

    ```
    Sample configuration for the Web service
    ```

    ```
    OAuth 2.0 security access filter
    ```

    ```
    Uncomment the following block to apply the filter for the UGC user
    ```

20. Save and close `web.xml`.

21. Restart the Web application.

14.5.3 Adding the UGC Web site extension to a JSP Web site

Install the User Generated Content Web site extension by adding a Java Web application to the live Web site.

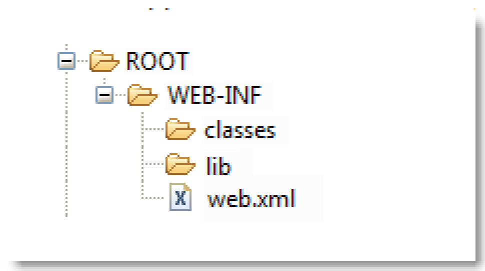
Steps to execute

1. Ensure that all the prerequisites for a Content Delivery Server Role are met on your target system.

2. Create the following Web application directory structure:

    a. In the root folder, create a `WEB-INF/` directory.

    b. In the `WEB-INF/` directory, create subdirectories `classes/` and `lib/`.



3. On the installation media, navigate to the `Content Delivery\roles\ugc\java\` folder.

4. Copy all the JAR files from the `lib\` subfolder, and all the JAR files from the `third-party-lib\` subfolder, to the `WEB-INF/lib/` folder of your Web application.

5. If you store published content in a database, download the JDBC driver (JAR file) required for your specific database vendor and copy it into the `WEB-INF/lib` directory of your Web application:

Database vendor	JDBC Driver	File to download
Microsoft SQL Server	Microsoft JDBC Driver 4.0 for SQL Server	`sqljdbc4.jar`
Oracle	Oracle JDBC driver 11.2.0.3	`ojdbc6.jar`

6. Navigate to `\Content Delivery\resources\configurations\`.

7. Copy `logback.xml` from that folder to the `WEB-INF/classes/` subdirectory of your Web application.

8. Copy and rename the following configuration files to the `WEB-INF/classes/` folder of your Web application:

Corresponding configuration file	Renamed configuration file
`cd_ambient_conf_sample.xml`	`cd_ambient_conf.xml`
`cd_dynamic_conf_sample.xml`	`cd_dynamic_conf.xml`
`cd_storage_conf_sample.xml`	`cd_storage_conf.xml`
`cd_link_conf_sample.xml`	`cd_link_conf.xml`
`cd_ugc_conf_sample.xml`	`cd_ugc_conf.xml`
`cd_wai_conf_sample.xml`	`cd_wai_conf.xml`

9. Open the file `cd_dynamic_conf.xml` for editing.

10. In the `<TCDLEngine>` section, insert the following element just before the end tag:

```
<TagBundle Resource="ugc_renderer_bundle.xml"/>
```

11. Save and close cd_dynamic_conf.xml.

12. Open cd_ambient_conf.xml for editing (if using).

13. Find the <Security> section and enclose it in comments. This element is only needed if you intend to set up OAuth authentication.

14. Save and close cd_ambient_conf.xml.

15. Copy cd_licenses.xml, your Content Delivery license file available from SDL Tridion Customer Support, to the WEB-INF/classes directory of your Web application.

    If you want to name the file differently or put it in a different location, you need to update the License element in your configuration files.

16. Configure logging.

17. Configure the Storage Layer and optionally encrypt sensitive strings.

18. Open cd_ambient_conf.xml for editing.

19. Save and close cd_ambient_conf.xml.

20. Open cd_ugc_conf.xml in a plain-text or XML editor.

21. Configure User Generated Content by editing the following elements:

    ODataEndPoint

    Set the URL attribute of this tag to the URL of the Content Delivery Web service that JSP tags must post submitted ratings and comments to.

    Security

    Set the attributes of this optional element, used for Web service authentication, as follows:

Attribute	Value
Endpoint	Set to the URL of the Access Token Server.
ClientId	Set to the client ID (user name) to authenticate.
ClientSecret	Set to the client secret (that is, password) to authenticate.

    You specified these parameters when you implemented authentication for the Content Delivery Web service.

22. Save and close cd_ugc_conf.xml.

23. From the installation media folder Content Delivery\resources\tld\, copy ugc.tld to the WEB-INF/lib/ subdirectory.

24. Enable the Ambient Data Framework by adding the following to your web.xml:

```
<filter>
  <filter-name>Ambient Data Framework</filter-name>
  <filter-class>com.tridion.ambientdata.web.AmbientDataServletFilter</filter-class>
```

```
</filter>
<filter-mapping>
  <filter-name>Ambient Data Framework</filter-name>
  <servlet-name>Content Delivery Web service</servlet-name>
</filter-mapping>
```

This enables the Ambient Data Framework for the Content Delivery Web service only. If you want to enable the Ambient Data Framework in the entire Web application, replace the `servlet-name` element with the following:

```
<url-pattern>/*</url-pattern>
```

25. If your Web application server is JBoss, do the following:

    a. Locate the following files in the `WEB-INF/lib/` subdirectory and remove them:

       - `stax-api.jar`
       - `serializer.jar`
       - `xalan.jar`

    b. If you use JBoss 5.1, add the following JVM property-value pair to the `JAVA_OPTS` variable: `-Dorg.jboss.logging.provider=slf4j`

    c. If you use JBoss 6, create a file with the following contents called `jboss-deployment-structure.xml` and save the file in the `WEB-INF/` directory:

```
<jboss-deployment-structure>
  <deployment>
    <exclusions>
      <module name="org.slf4j" />
      <module name="org.slf4j.impl" />
      <module name="com.sun.jersey" />
      <module name="com.sun.jersey.json" />
      <module name="com.sun.jersey.core" />
    </exclusions>
  </deployment>
</jboss-deployment-structure>
```

    d. If you use JBoss 6, open `WEB-INF/web.xml` file, add the following section and save the file:

```
<context-param>
    <param-name>resteasy.scan</param-name>
    <param-value>false</param-value>
</context-param>
<context-param>
     <param-name>resteasy.scan.resources</param-name>
     <param-value>false</param-value>
</context-param>
<context-param>
      <param-name>resteasy.scan.providers</param-name>
      <param-value>false</param-value>
</context-param>
```

26. If your Web application server is WebLogic, do the following:

    a. Remove the file `stax-api.jar` from your `WEB-INF/lib/` subdirectory.

    b. Add a `weblogic.xml` file to the `WEB-INF` folder of your Web application that forces WebLogic to use the libraries contained in the Web application, rather than its own libraries. The contents of this `weblogic.xml` file must be as follows:

```
<weblogic-web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.bea.com/ns/weblogic/weblogic-web-app http://www.bea.com/ns/weblogic/weblogic-
web-app/1.0/weblogic-web-app.xsd"
  xmlns="http://www.bea.com/ns/weblogic/weblogic-web-app">
  <container-descriptor>
    <prefer-application-packages>
      <package-name>javax.persistence.**</package-name>
      <package-name>antlr.*</package-name>
      <package-name>org.slf4j.**</package-name>
      <package-name>org.apache.log4j.**</package-name>
      <package-name>org.apache.xml.serializer.**</package-name>
    </prefer-application-packages>
  </container-descriptor>
</weblogic-web-app>
```

27. Restart the Web application.

14.6 Configuring Moderation Web Service and Community Web Service

User Generated Content requires you to configure the User Generated Content Community Web Service and the User Generated Content Moderation Web Service.

14.6.1 Configuring UGC

To configure User Generated Content, edit your User Generated Content configuration file for both User Generated Content Web services to set a scoring threshold and a whitelist for comments.

Steps to execute

1. Access the SDL Tridion installation media and navigate to the `Content Delivery\resources\configurations\` folder.
2. Navigate to the root directory of your Community Web Service.
3. Copy the `cd_ugc_conf.xml` to one of the following locations:
   - In a JSP Web application, in the `WEB-INF/classes/` directory.
   - In a .NET Web application, in the `bin\config\` subfolder.
4. Open `cd_ugc_conf.xml` in a plain-text or XML editor.
5. In the `<Comment>` section, configure the following subelements:

   `ScoreMinimumThreshold`

   Indicates the minimum score that a comment must have in order to be removed from the published Web page and offered to a moderator. This property defaults to the value "-10". Comment score depends on the amount of positive votes ('likes' or 'upvotes') and negative votes ('dislikes' or 'downvotes') that the comment has received. Every upvote raises the comment score by 1; every downvote lowers it by 1.

   `WhiteList`

This section lists XHTML tags that you want to allow to be read and processed in a comment (any tags not in this list are ignored). Represent each XHTML tag as a `Tag` element with its `Name` attribute set to the name of the tag you want to allow. For example, `<Tag Name="em" />` lets commenters use the `<em>` tag in a comment.

Also, you can force certain attributes of a tag to have a certain value by creating a subelement of the format `<Attribute Name="ATTRNAME" EnforcedValue="ATTRVALUE">` where `ATTRNAME` is the attribute you want to set a value for, and `ATTRVALUE` is the enforced value for that attribute, which overwrites any existing value.

Specifically for the a tag, you can also enforce only certain protocols (say, only links to `http://` resources, not `https://` resources), create a subelement `<Attribute Name="href">` with a subelement `<Protocol Name="http"/>` where `http` is the protocol you want to allow.

The following is an example of an XHTML whitelist. It only allows bold, italic and underlined text; soft returns in text; and links to HTTP resources only, which should not be followed by search engines and should open in a new tab or window.

```
<WhiteList>
  <Tag Name="a">
    <Attribute Name="href">
      <Protocol Name="http"/>
    </Attribute>
    <Attribute Name="rel" EnforcedValue="nofollow"/>
    <Attribute Name="target" EnforcedValue="_blank"/>
  </Tag>
  <Tag Name="b"/>
  <Tag Name="br"/>
  <Tag Name="i"/>
  <Tag Name="u"/>
</WhiteList>
```

6. Save and close `cd_ugc_conf.xml`.

7. Restart the Web application.

8. Repeat the steps to configure the Moderation Web Service.

14.6.2 Enabling processing of UGC TCDL tags

You need to add a tag bundle to your dynamic configuration file `cd_dynamic_conf.xml` to enable your the Community Web Service and Moderation Web Service to process User Generated Content TCDL tags at runtime.

Context

The default Template Building Blocks for User Generated Content create TCDL tags that are handled differently depending on the `Target Language` setting in the Publication Target:

- if the Target Language is set to `JSP` or `ASP.NET`, User Generated Content tags are transformed into JSP or ASP.NET code during publishing that require no further processing.

- if the Target Language is set to REL, User Generated Content tags are published as is (TDCL) to the Web site that require processing at request time.

Steps to execute

1. Access the following directory where your Community Web Service is running:
   - the WEB-INF/classes/ directory in a JSP Web application.
   - the bin\config\ subfolder in a .NET Web application.
2. Open the file cd_dynamic_conf.xml in a plain-text or XML editor.
3. In the <TCDLEngine> section, insert the following element just before the end tag:

   ```
   <TagBundle Resource="ugc_renderer_bundle.xml"/>
   ```
4. Save and close cd_dynamic_conf.xml.
5. Repeat the steps to configure the Moderation Web Service.

Result

You have enabled the processing and rendering ofUser Generated Content TCDL tags at request time.

14.6.3 Configuring storage of UGC comments and ratings

Configure in cd_storage_conf.xml where User Generated Content comments, ratings user data and item statistics are stored.

Steps to execute

1. Access the following directory where your Community Web Service is running:
   - the WEB-INF/classes/ directory in a JSP Web application.
   - the bin\config\ subfolder in a .NET Web application.
2. Open cd_storage_conf.xml in a plain-text or XML editor.
3. In the Storages element, add a new Storage element and configure it to point to the User Generated Content database you created. The default name suggested for the database was Tridion_UGC.
4. In the StorageBindings element inside the Storages element, add the following:

   ```
   <Bundle src="ugc_dao_bundle.xml" />
   ```
5. In the ItemTypes element, add the following new Item elements:

   ```
   <Item typeMapping="Comment" cached="false" storageId="STORAGE_ID" />
   <Item typeMapping="Rating" cached="false" storageId="STORAGE_ID" />
   <Item typeMapping="UGCUser" cached="false" storageId="STORAGE_ID" />
   ```

```
<Item typeMapping="UGCItemStats" cached="false" storageId="STORAGE_ID" />
```

where `STORAGE_ID` is the ID of the new `Storage` element you just created. Caching is initially set to `false` to keep the Web site as up-to-date as possible. Consult the SDL Tridion core documentation to learn about turning caching on and configuring caching time.

6. Save and close `cd_storage_conf.xml`.

7. Access your Moderation Web Service and copy and paste the `cd_storage_conf.xml` file you just saved into the `WEB-INF/classes/` subdirectory (JSP) or `bin\config\` subfolder, overwriting the existing `cd_storage_conf.xml`.

14.6.4 Configuring the UGC cartridge for the Community Web service

The User Generated Content cartridge for the Ambient Data Framework takes care of, amongst others, the Content Delivery side of a User Generated Content Audience Manager integration. The Community Web Service requires two Claim Processors.

Steps to execute

1. Access the SDL Tridion installation media and navigate to the `Content Delivery\resources\configurations\` folder.

2. Access your Community Web Service Web application and navigate to the `WEB-INF/classes/` subdirectory (JSP) or `bin\config\` subfolder (.NET).

3. Copy the file `ugc_ambient_cartridge_sample.xml` from the installation media to this location.

4. Rename `ugc_ambient_cartridge_sample.xml` to `ugc_ambient_cartridge.xml`.

5. Open `ugc_ambient_cartridge.xml` for editing.

6. Ensure that the following Claim Processors (`ClaimProcessorDefinition` elements) are set as described:

   - The `ClaimProcessorDefinition` element that has an `ImplementationClass` attribute ending in `AllowAnonymousPostClaimProcessor` must be present and not enclosed in comment tags.
   - The `ClaimProcessorDefinition` element that has an `ImplementationClass` attribute ending in `PostAllowedByOwnerClaimProcessor` must be present and not enclosed in comment tags.
   - The `ClaimProcessorDefinition` element that has an `ImplementationClass` attribute ending in `PostAllowedByEveryoneClaimProcessor` must be absent or enclosed in comment tags.

7. Save and close `ugc_ambient_cartridge.xml`.

8. Open the file `cd_ambient_conf.xml`.

9. In the `<Cartridges>` section, add the following `Cartridge` element:

```
<Cartridge File="ugc_ambient_cartridge.xml" />
```

10. Save and close `cd_ambient_conf.xml`.

11. Restart the Community Web Service Web application.

14.6.5 Configuring the UGC cartridge for the Moderation Web service

The User Generated Content cartridge for the Ambient Data Framework takes care of, amongst others, the Content Delivery side of a User Generated Content Audience Manager integration. The Community Web Service requires two Claim Processors.

Steps to execute

1. Access the SDL Tridion installation media and navigate to the `Content Delivery\resources\configurations\` folder.

2. Access your Community Web Service Web application and navigate to the `WEB-INF/classes/` subdirectory (JSP) or `bin\config\` subfolder (.NET).

3. Open `ugc_ambient_cartridge.xml` for editing.

4. Ensure that the following Claim Processors (`ClaimProcessorDefinition` elements) are set as described:
    - The `ClaimProcessorDefinition` element that has an `ImplementationClass` attribute ending in `AllowAnonymousPostClaimProcessor` must be present and not enclosed in comment tags.
    - The `ClaimProcessorDefinition` element that has an `ImplementationClass` attribute ending in `PostAllowedByEveryoneClaimProcessor` must be present and not enclosed in comment tags.
    - The `ClaimProcessorDefinition` element that has an `ImplementationClass` attribute ending in `PostAllowedByOwnerClaimProcessor` must be absent or enclosed in comment tags.

5. Save and close `ugc_ambient_cartridge.xml`.

6. Copy the file `ugc_ambient_cartridge_sample.xml` from the installation media to this location.

7. Rename `ugc_ambient_cartridge_sample.xml` to `ugc_ambient_cartridge.xml`.

8. Open the file `cd_ambient_conf.xml`.

9. In the `<Cartridges>` section, add the following `Cartridge` element:

```
<Cartridge File="ugc_ambient_cartridge.xml" />
```

10. Save and close cd_ambient_conf.xml.

11. Restart the Moderation Web Service Web application.

14.6.6 Restarting the Web services

When you have set up and configured the Community Web Service and Moderation Web Service, you need to apply the changes you made by restarting the Web applications that contain these Web services.

14.7 Configuring UGC on your Web site

On the machine that serves Web site content to visitors, perform some configuration to enable visitors to submit comments and ratings.

14.7.1 Configuring the UGC cartridge on the Web site

The User Generated Content cartridge for the Ambient Data Framework takes care of, amongst others, the Content Delivery side of a User Generated Content-Audience Manager integration. The Web site requires one Claim Processor.

Steps to execute

1. Access the SDL Tridion installation media and navigate to the `Content Delivery\resources\configurations\` folder.

2. Access your Web site and navigate to the `WEB-INF/classes/` subdirectory (JSP) or `bin\config\` subfolder (.NET).

3. Copy the file `ugc_ambient_cartridge_sample.xml` from the installation media to this location.

4. Rename `ugc_ambient_cartridge_sample.xml` to `ugc_ambient_cartridge.xml`.

5. Open `ugc_ambient_cartridge.xml` for editing.

6. Ensure that the following Claim Processors (`ClaimProcessorDefinition` elements) are set as described:

   - The `ClaimProcessorDefinition` element that has an `ImplementationClass` attribute ending in `AllowAnonymousPostClaimProcessor` must be present and not enclosed in comment tags.
   - The `ClaimProcessorDefinition` element that has an `ImplementationClass` attribute ending in `PostAllowedByEveryoneClaimProcessor` must be absent or enclosed in comment tags.

- The ClaimProcessorDefinition element that has an ImplementationClass attribute ending in PostAllowedByOwnerClaimProcessor must be absent or enclosed in comment tags.

7. Find the ClaimProcessorDefinition element that has an ImplementationClass element ending in PostAllowedByEveryoneClaimProcessor, and enclose this element in comment tags.

8. Save and close ugc_ambient_cartridge.xml.

9. Open the file cd_ambient_conf.xml.

10. In the <Cartridges> section, add the following Cartridge element:

```
<Cartridge File="ugc_ambient_cartridge.xml" />
```

11. Save and close cd_ambient_conf.xml.

12. Restart your Web site.

14.7.2 UGC Web site security

Rating and commenting functionality creates new entry points for Web site visitors, and brings with it the risk of compromising security. Specifically, the Web service end point and the Web site itself are potentially vulnerable for attack. This topic explains some of the security concerns surrounding User Generated Content, and how you can address them.

Note: This topic only addresses some security concerns you may encounter. In the final analysis, the security level of User Generated Content heavily depends on how secure you make it. For example, to implement the business rule that visitors can only edit their own comments, you must create a Web page that actively prevents users from editing other people's comments. SDL strongly recommends integrating your User Generated Content implementation with existing authorization and authentication frameworks.

Web service authentication (OAuth)

You can implement authentication for your Content Delivery Web service. If you already have authentication set up for the Web service, configure Web service end points in Content Manager.

Accessing the Web service through SSL (HTTPS connection)

If you have set up your Content Delivery Web service to be only accessible through HTTPS, configure User Generated Content on the Content Manager to access the Web service.

SQL injection in comment forms

User Generated Content uses Hibernate to prevent SQL statements in comment forms from being executed. Visitors cannot manipulate your database using commenting.

JavaScript injection or other Web code injection in comment forms

On your Web site, you can edit a whitelist of allowed HTML elements in submitted comments. Any tags not listed in this whitelist are stripped out of the comment. By default, the whitelist only lets commenters include links to HTTP resources, preventing JavaScript injection, and opens those links in a new tab or window. Of course, you can edit the whitelist to disallow any links of any kind, or even any HTML at all.

Comment flooding and mass rating or comment voting

By default, User Generated Content is not set up to prevent visitors from posting multiple comments on the same Component or Page, rating the same Component or Page multiple times, or downvoting or upvoting a comment multiple times. This makes it easier to demonstrate the functionality of the product, for example, to show how multiple downvotes cause a comment to 'drop off' the Web page.

However, this also leaves open the possibility for abuse. Malicious visitors can flood a Web page with comments, downvote a comment into nonexistence, or rate a Component or Page any way they want.

You can prevent this by activating one of the Content Validators that SDL Tridion ships with, or by building a new Content Validator yourself.

14.7.3 Setting up visitor account management

You need to manage visitor accounts to identify and manage people who visit your site and leave comments or submit ratings.

Motivation to manage visitor accounts

Visitor accounts can help prevent abuse of your commenting and rating system. If you set up visitor accounts, disable anonymous access.

For a variety of reasons, you may want to identify the Web site visitors who submit comments or ratings:

- To allow a visitor to edit or delete comments they wrote themselves (as demonstrated in the sample JSP and ASP.NET Web pages included on the installation media)
- To prevent a visitor from rating an item more than once
- To prevent a visitor from flooding an item with comments
- To ban a visitors who acts abusively or otherwise behaves improperly

Anonymous users can still be identified through cookies, which means that in a completely anonymous environment, you can still allow visitors to edit or delete the comments they submitted themselves. However, you cannot prevent intentional abuse by visitors who delete their own cookies.

If you decide to set up accounts for visitors, SDL strongly encourages you to disable anonymous access altogether, because it may compromise the security of your login system.

Visitor account management

Set up visitor account management in User Generated Content as a standalone feature, integrated with Audience Manager, or integrated with an external system.

User Generated Content ships with two sample Web pages, an ASP.NET page and a JSP page, which demonstrate how to interact with User Generated Content for identification and authentication of users. In these samples, the comment is submitted along with a user name and e-mail address supplied by the user.

You can find these sample pages on the installation media in the folder `Content Delivery\resources\samples\ugc\`, subfolder `java\` or `dotNet\`.

You can manage visitor accounts as follows

Standalone setup

> If Audience Manager is not set up for your SDL Tridion installation, all visitor accounts are managed in the User Generated Content database itself. Visitors are identified using cookies only.

Integrated with Audience Manager

> If Audience Manager is set up for your SDL Tridion installation, you can integrate with its Contacts.

Integrated with an external system

> This option requires you to handle the identification of visitors through your external system, and supply the unique external ID associated with a visitor account in that system to the User Generated Content database.

Integrating UGC on a JSP Web site

Integrate User Generated Content with Audience Manager Contacts, or with your custom contacts stored in your custom user management system, from a JSP Web site.

Steps to execute

1. Access the SDL Tridion installation media and navigate to the `Content Delivery\roles\` folder.

2. Navigate to the folder `Content Delivery\resources\samples\ugc\java`.

3. Open `changeProfile.jsp` the file in a plain-text editor. This file is an updated version of the sample `changeProfile.jsp` page that SDL Tridion ships with User Generated Content code has been added to it.

4. Open your own JSP Web page that your visitors use to log in and to change their profile, in a plain-text editor.

5. Copy the following import statement from `changeProfile.jsp` and insert it at the top of your page:

```
<%@ page import="com.tridion.ambientdata.claimstore.ClaimStore,
                 com.tridion.ambientdata.web.WebContext,
                 com.tridion.broker.StorageException,
                 com.tridion.storage.StorageManagerFactory,
                 com.tridion.storage.ugc.UGCTypeMapping,
                 com.tridion.storage.ugc.UGCUser
                 com.tridion.storage.ugc.dao.UserDAO,
                 com.tridion.ugc.odata.claimprocessor.BasePostClaimProcessor,
                 org.codehaus.jettison.json.JSONException,
                 org.codehaus.jettison.json.JSONObject,
                 java.util.List" %>
```

6. Add the following static method from `changeProfile.jsp` to the page. This method is set up to integrate with Audience Manager. To integrate it with your custom external user management system, check the comments marked CUSTOM.

```
/**
 * Utility method to synchronize a User Generated Content user with a Contact.
 *
 * @param contact           The contact to synchronize with.
 * @throws JSONException    If the UGCUser externalId could not be created.
 * @throws StorageException On any error during database access.
 */
static void syncUGCUserWithContact(Contact contact) throws JSONException,
 StorageException {
  JSONObject externalIdJSON = new JSONObject();
  for (Object field : contact.getDetails().getIdentificationFields().values()) {
    ExtendedDetail extendedDetail = (ExtendedDetail) field;
    externalIdJSON.put(extendedDetail.getFieldName(), extendedDetail.getValue());
  }
  ClaimStore claimStore = WebContext.getCurrentClaimStore();
  UserDAO userDAO = (UserDAO)
 StorageManagerFactory.getDefaultDAO(UGCTypeMapping.UGC_USER.getConfigurationName());
  List <UGCUser> ugcUsers = userDAO.findByExternalId(externalIdJSON.toString());
  if (ugcUsers != null && !ugcUsers.isEmpty()) {
    // Get first 'Contact-linked' User Generated Content user.
    UGCUser ugcUser = ugcUsers.remove(0);
    // CUSTOM: To integrate with your custom external user management system,
    // remove the following line, which integrates with Audience Manager.
    claimStore.put(BasePostClaimProcessor.USER_CLAIM, ugcUser.getId());
    // CUSTOM: To integrate with your custom external user management system,
    // uncomment the following line, where "userId" is a String that matches the ID
 column
    // in the User Generated Content Users table:
    // claimStore.put(URI.create("taf:claim:contentdelivery:webservice:user"), userId);

    //
    // INSERT CODE HERE if remaining users need to be aggregated
    //
  } else {
    // Else: Current user claim is leading.
    if (claimStore.contains(BasePostClaimProcessor.USER_CLAIM)) {
      String userId = claimStore.get(BasePostClaimProcessor.USER_CLAIM, String.class);
      UGCUser user = userDAO.findByPrimaryKey(userId);
      if (user != null) {
        // Link current tracked User Generated Content user to Contact.
        user.setExternalId(externalIdJSON.toString());
      } else {
        // Create a new 'Contact-linked' User Generated Content user.
        user = new UGCUser(userId, null, null, externalIdJSON.toString());
      }
      userDAO.store(user);
    }
  }
}
```

7. In your Web page, find the places in your code where the user has successfully logged in, and where the user has changed his or her profile. In this location, make the call to the method you just added, catching any exceptions thrown:

```
// Sync User Generated Content User
try {
  syncUGCUserWithContact(contact);
} catch (StorageException e) {
  addToErrorMessage(errorBuffer, "Unable to set current Contact as current User
 Generated Content user. " + e.getMessage());
} catch (JSONException e) {
  addToErrorMessage(errorBuffer, "Problem reading Contact's identification fields. " +
 e.getMessage());
}
```

8. Save and close your JSP Web page.

Integrating UGC on a .NET Web site

Integrate User Generated Content with Audience Manager Contacts, or with your custom contacts stored in your custom user management system, from a .NET Web site.

Steps to execute

1. Access the SDL Tridion and bavigate to the folder `Content Delivery\resources\samples\ugc\dotNet`.

2. Open the file `ChangeProfile.aspx.cs` in a plain-text editor. This file is an updated version of the sample `ChangeProfile.aspx.cs` file that SDL Tridion ships with. User Generated Content code has been added to it.

3. Open your own ASP.NET Web page (or underlying `.aspx.cs` file) that your visitors use to log in and to change their profile, in a plain-text editor.

4. From `ChangeProfile.aspx.cs`, copy the following import statements to the top of your .NET page:

   ```
   using System;
   using System.Configuration;
   using Tridion.OutboundEmail.ContentDelivery;
   using Tridion.OutboundEmail.ContentDelivery.Profile;
   using Tridion.ContentDeliveru.UGC.WebService;
   ```

5. Add the following static method from `ChangeProfile.aspx.cs` to your page (on the sample page, it is added to the region called "Utility methods". This method is set up to integrate with Audience Manager. To integrate it with your custom external user management system, check the comments marked CUSTOM.

```
private static void SyncUGCUserWithContact(Contact contact)
{
  WebServiceClient WebServiceClient = new WebServiceClient();

  //Modify to fit Audience Manager identification fields
  string externalId = "{\"IDENTIFICATION_KEY\":\"" + contact.ExtendedDetails["IDENTIFICATION_KEY"] +
    "\"IDENTIFICATION_SOURCE\":\"" + contact.ExtendedDetails["IDENTIFICATION_SOURCE"] + "}";

  XmlNamespaceManager nsmgr = new XmlNamespaceManager(new NameTable());
  nsmgr.AddNamespace("m", "http://schemas.microsoft.com/ado/2007/08/dataservices/metadata");
  nsmgr.AddNamespace("d", "http://schemas.microsoft.com/ado/2007/08/dataservices");

  XmlDocument users = new XmlDocument();
  users.LoadXml(WebServiceClient.DownloadString("/Users?$filter=ExternalId eq " + externalId));

  XmlElement firstUser = (XmlElement)users.SelectSingleNode)"feed/entry[1]/content.m:properties", nsmgr);
  if (firstUser != null)
  {
    // Get first 'Contact-linked' User Generated Content user.
    string userId = firstUser.SelectSingleNode("d:Id", nsmgr).InnerText;
    // CUSTOM: To integrate with your custom external user management system,
    // remove the following line, which integrates with Audience Manager.
    AmbientDataContext.CurrentClaimStore.Put(UserClaim, userId);
    // CUSTOM: To integrate with your custom external user management system,
    // uncomment the following line, where "userId" is a String that matches the ID column
    // in the User Generated Content Users table:
    // AmbientDataContext.CurrentClaimStore.Put(new Uri("taf:claim:contentdelivery:webservice:user"), userId);

    //
    // INSERT CODE HERE if remaining users need to be aggregated.
    //

  }
  else
  {
    // Else: Current user claim is leading.
    if (AmbientDataContext.CurrentClaimStore.Contains(UserClaim))
    {
      XmlDocument.existingUser = new XmlDocument();
      existingUser.LoadXml(WebServiceClient.DownloadString("/Users(Id=" +
        AmbientDataContext.CurrentClaimStore.Get<string>(UserClaim) + ")"));

      XmlElement userElement = (XmlElement)existingUser.SelectSingleNode("entry/content/m:properties", nsmgr);
      if (userElement != null)
```

```
      {
        // Link current tracked User Generated Content user to Contact.
        User user = new User(
          userElement.SelectSingleNode("d:Id", nsmgr).InnerText,
          userElement.SelectSingleNode("d:Name", nsmgr).InnerText,
          userElement.SelectSingleNode("d:EmailAddress", nsmgr).InnerText,
          externalId);

        WebServiceClient.UploadString("/Users(Id=" + externalId + ")", "PUT", "{d:" + user.ToJSON(0) + "}");
      }
      else
      {
        // Create new 'Contact-linked' User Generated Content user.
        User user = new User(
          AmbientDataContext.CurrentCalimStore.Get<string>(UserClaim),
          null,
          null,
          externalId);

        WebServiceClient.UploadString("/Users", "POST", "{d:" + user.ToJSON() + "}");
      }
    }
  }
}
```

6. In your Web page, find the places in your code where the user has successfully logged in, and where the user has changed his or her profile. In this location, make the call to the method you just added by calling `SyncUGCUserWithContact(contact)`.

7. Save and close your ASP.NET Web page.

14.8 Configuring UGC on Content Manager

User Generated Content is installed on the Content Manager server when you run the Content Manager installer. You need to configure User Generated Content on the Content Manager to add comments and rating controls to your Components and Pages.

14.8.1 Enabling commenting and rating in a Template

Enable commenting and rating in a Template to let users see and submit comments and ratings on Web pages that were rendered with such a Template, or on Web pages that contain Component Presentations rendered with such a Template.

Context

You need to decide which Component Template(s) and Page Template(s) you want to enable ratings and/or comments. Pages or Components rendered with these templates will then display the following information on the published Web page:

- the current average rating
- an interface to submit a rating
- all the comments posted so far (paginated if needed)
- a form to provide a new comment

Steps to execute

1. Start Template Builder.
2. Open one of the Templates in which you want to enable commenting and rating.
3. If this is the first Template for which you want to enable commenting and ratings, select **Tools** > **Create Default Building Blocks** .

   New Template Building Blocks are added to the **Building Blocks** area on the left. In Content Manager Explorer, new items will also appear in the `Building Blocks\Default Templates` folder of this Publication
4. Insert the Template Building Block **Enable User Generated Content Processing** before the Template Building Block that generates your output (typically your Dreamweaver Template Building Block) in the Template.
5. Modify the Template Building Block that generates your output so that it displays ratings and comments and allows visitors to set ratings and add comments.

   The **Enable User Generated Content Processing** makes a number of TCDL tags available that you can use to do this. You can use the **Default UGC Dreamweaver Template Design** Template Building Block as example of how to work with comments and ratings. Note however that this Template Building Block does not produce any output.
6. Save and close your template.
7. Repeat the same procedure for all other Component Templates and Page Templates for which you want to enable commenting and rating.

Next steps

The templates now add ratings and comments to Components or Pages they render, unless a user explicitly configures those Components or Pages not to show ratings and/or comments.

14.8.2 Configuring Moderation Web service end points

Configure the end points of the Moderation Web Service to give the Content Manager access to this Web service to retrieve and store comments and ratings. If you want to aggregate comments and ratings from multiple Content Delivery instances, you can configure multiple end points.

Steps to execute

1. On the Content Manager server machine, access the Content Manager root location (defaults to `C:\Program Files (x86)\Tridion\`). From there, navigate to the subfolder `web\WebUI\Models\UGC\Configuration\` and open the `DataSources.xml` file you find there in a plain-text or XML editor.
2. In this location, find the `<DataSource>` element and set the value of the subelements as follows:

`<Key>`

A number that uniquely identifies the Moderation Web Service end point.

`<Match>`

A regular expression that specifies which items in Content Manager should use this data source. If you use only one data source, leave this element set to its default value, which is `^(ugc:|oe:|tcm:)+`. But if the Moderation Web Service only provides access to comments and ratings for items from, say, a Publication with ID 14, change the string to `(tcm:0-14-1$)|(tcm:14-\d*$)|(tcm:14-\d*-64$)|(oe:14-\d*-196616$)|(ugc:)`.

`<Url>`

The URL of the Moderation Web Service.

`<TimeOut>` (optional, defaults to 100000)

The time (in milliseconds) after which Content Manager considers the Content Delivery Web service to be unreachable. For example, a value of 4000 creates a 4-second timeout limit.

`<Threshold>` (optional, defaults to 0)

Set this number to the same value as the value of the `CommentScoreMinimumThreshold` element in the User Generated Content configuration file `cd_ugc_conf.xml`. In this `DataSources.xml` file, the value is used to change the visualization of the comment score in various parts of the Content Manager Explorer GUI: a comment score that is equal to or lower than this `Threshold` value is colored red; all other comment scores are colored black.

`<RatingMinimum>` (optional, defaults to 0)

The lowest rating that a visitor can give to a Component or Page (can be any integer, including negative numbers).

`<RatingMaximum>` (optional, defaults to 5)

The highest rating that a visitor can give to a Component or Page (can be any integer, including negative numbers).

`<OAuthEnabled>` (optional, defaults to `false`)

Set this element to the value `true` if you have configured authentication for the Moderation Web Service.

`<Locale>` (optional)

The locale used by Content Delivery, in the format `ll-cc`, where *ll* is the language and *cc* is the country. For example, `en-us`.

`<ClientId>` (optional)

If you authenticate access to the Web service, the ID of a client account listed in the Web service configuration file `cd_webservice_conf.xml`. Leave this element empty if not in use.

`<ClientSecret>` (optional)

If you authenticate access to the Web service, the password of the client account specified under `<ClientId>`. Leave this element empty if not in use.

`<EventSystemClientId>` (optional)

If you authenticate access to the Web service, the ID of a client account listed in the Web service configuration file `cd_webservice_conf.xml`. Leave this element empty if not in use. This ID is used for cases in which the Event System triggers a Web service call, for example, when the deletion of a Page or Component triggers the deletion of its associated Comments and Ratings. The value of this element can be the same as the value of `ClientId`.

`<EventSystemClientSecret>` (optional)

If you authenticate access to the Web service, the password of the client account specified under `<EventSystemClientId>`. Leave this element empty if not in use. This password is used for cases in which the Event System triggers a Web service call. The value of this element can be the same as the value of `ClientSecret`.

`<AccessTokenUrl>` (optional)

If you authenticate access to the Web service, the URL of your access token. Leave this element empty if not in use.

3. If you want to configure additional instances of the Moderation Web Service for the purpose of aggregating comments and ratings from multiple Web sites, add a new `<DataSource>` and specify its parameters.

4. Save and close `DataSources.xml`.

5. Restart IIS.

14.8.3 Granting UGC moderation rights

By default, only administrator-level users can moderate, modify or delete comments and ratings. To enable non-administrator users to manage comments and ratings, grant them User Generated Content moderation rights.

Context

Steps to execute

1. Open Content Manager Explorer.

2. In the **Publications** navigation pane, select the **Content Management** node.

3. In the list view, select the Publication you want to edit and in the Ribbon toolbar and click **Properties** .

   The Publication window opens.

4. Click the **Security** tab:

   a. On the left-hand side, select **Users and Groups**. Select the **Show all** check box to view Users as well as Groups.

b. On the right-hand side, select the **User Generated Content** check box to assign this **Right** to the selected Users or User Group.

5. Repeat the process for all Users you want to grant User Generated Content moderation rights.

6. Click Save and Close .

14.8.4 Configuring UGC for HTTPS

If your User Generated Content Web services use HTTPS, perform some additional configuration in the User Generated Content `web.config` file.

Requirements

To make User Generated Content work with HTTPS, you need to have a valid SSL certificate in place that is not expired, and was issued by a trusted certificate authority.

Steps to execute

1. On the Content Manager server, access the `web\WebUI\Models\UGC\` subfolder of the Content Manager root location (defaults to `C:\Program Files (x86)\Tridion\`). In this folder, open the file `web.config` for editing.

2. In this file, find the `service` element that has its `name` attribute set to the value `Tridion.UGC.Model.Services.UGC` and its `behaviorConfiguration` attribute set to `Tridion.Web.UI.ContentManager.WebServices.DeveloperBehavior`.

3. Inside this element, ensure the presence of an uncommented `endpoint` element that has its attributes set as follows:

```
<endpoint
  address=""
  behaviorConfiguration="Tridion.Web.UI.ContentManager.WebServices.AspNetAjaxBehavior"
  binding="webHttpBinding"

 bindingConfiguration="Tridion.Web.UI.ContentManager.WebServices.WebHttpsBindingConfig"
  contract="Tridion.UGC.Model.Services.UGC" />
```

4. Save and close `web.config` to commit your changes, and restart IIS.

14.8.5 Configuring UGC for single sign-on

If your Content Manager uses single sign-on, perform some additional configuration in the User Generated Content `web.config` file.

Steps to execute

1. On the Content Manager server, access the `web\WebUI\Models\UGC\` subfolder of the Content Manager root location (defaults to `C:\Program Files (x86)\Tridion\`). In this folder, open the file `web.config` for editing.

2. In this file, inside the `system.web` element contained in the `configuration` section, find the `identity` element and set its `impersonate` attribute to the value `false`.

3. Save and close `web.config` to commit your changes, and restart IIS.

14.8.6 Enabling logging for UGC

By default, the installer has set up User Generated Content to not produce any logs, because logging adversely affects performance. However, if you find yourself faced with an implementation that does not function as you would like, and you want to obtain logging, you can enable logging.

Steps to execute

1. Access the Content Manager root location (defaults to `C:\Program Files (x86)\Tridion\`).

2. Navigate to the subdirectory `web\WebUI\WebRoot\bin\`.

   You see a file `Tridion.UGC.Model.dll` in this location.

3. Back up `Tridion.UGC.Model.dll` to a safe location.

4. Copy the file `Tridion.UGC.Model.dll` from the subdirectory `trace\` to the current location.

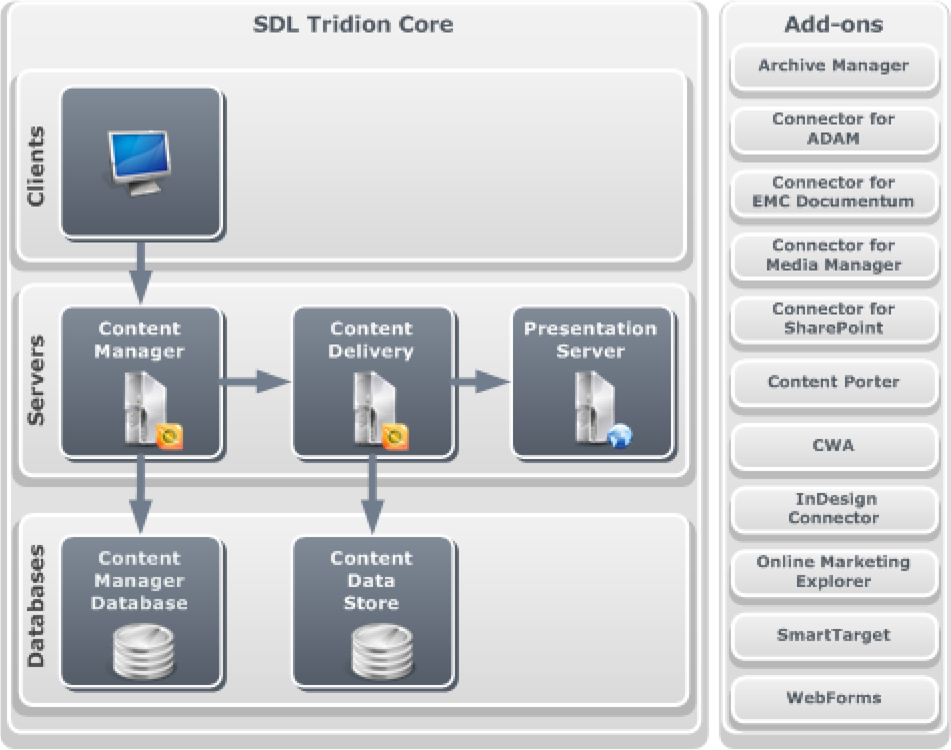
5. Perform an IIS reset.

Result

User Generated Content now writes log entries to the existing log file `Tridion.Web.Trace`, located in the `web\WebUI\WebRoot\` subfolder of the Content Manager root location. Remember to disable logging by switching the DLL files around once you have finished debugging User Generated Content. This will improve performance.

Chapter 15 Add-on products

Various add-on products are available that plug into SDL Tridion. An add-on is a product that is delivered separately to SDL Tridion and installed on top of the SDL Tridion platform. For information on how to install these products, refer to the documentation for these products.



15.1 Add-on product versions compatible with SDL Tridion 2013

This topic lists which versions of add-on products are known to be compatible with SDL Tridion 2013.

Add-on product	Compatibili version	Hotfix or other requirement	Link
Content Porter	2013	None	Content Porter 2013
Archive Manager	1.0 SP2	Hotfix 83265	Archive Manager 1.0 SP2
Connector for ADAM	1.0	External Content Library (ECL) 7.0 (part of suite installer)	

Add-on product	Compatibili version	Hotfix or other requirement	Link
Connector for Media Manager	1.1	External Content Library (ECL) 7.0 (part of suite installer)	
Online Marketing Explorer	2011 SP2	None	

Note: At the time of release, the compatibility requirements for all add-ons was not yet known. For the most up-to-date information about which add-ons are compatible with this release, contact SDL Customer Support.

Know issues with add-ons

The following add-ons have known issues associated with them:

Archive Manager 1.0 SP2

Archive Manager fails when a transactional unpublish occurs.

Chapter 16 Installing offline documentation

The online documentation portal for SDL Tridion and add-on products requires an active internet connection. To work offline, install the documentation locally on a Windows machine and download API documentation from SDL Tridion World.

Note: You can access online documentation portal at http://sdllivecontent.sdl.com/LiveContent/. Log on as **visitor**, password **sdltridion**.

16.1 Offline documentation portal prerequisites

To install the documentation portal locally, the following hardware and software requirements apply:

Disk space

You require approximately 300-400 MB of free disk space.

Operating Systems

You can install the offline documentation portal on a Windows 2008 or Windows 2003 machine.

You may have to modify your security settings as Windows 2008 or Windows 2003 machines typically restrict browsing on the machine itself for security reasons.

Java version

The SDL LiveContent installer requires Java 1.6, 32-bit version. Make sure the `JAVA_HOME` environment variable points to the 32-bit version. Open a command prompt and type `java –version`. If the version does not report the 32-bit version, change the `Path` environment variable to point to your Java `\bin` folder.

16.2 Installing the offline documentation portal powered by SDL LiveContent

The offline documentation portal installer installs version 5.2 of SDL LiveContent on your local machine and SDL Tridion and add-on product documentation.

Requirements

Read the Offline documentation portal prerequisites before running the installer.

The offline documentation portal installer installs version 5.2 of SDL LiveContent on your local machine. If you have an older version of SDL LiveContent, uninstall it before running the installer.

Steps to execute

1. Log on to the SDL Tridion World Web site at http://www.sdltridionworld.com.
2. Navigate to the documentation page for the current SDL Tridion release.
3. Download the offline documentation ZIP.
4. Unzip the installer files to a folder on your local machine. The ZIP file contains:
   - an installer executable
   - one or more compression files starting at`.Z01`
   - a `.pac` file package file
   - a PDF document of the installation procedure
5. Run the `install.exe` and follow the instructions on the screen.

   Note: When prompted for a license file, answer that you do not have one.

Result

The installer creates an offline version of the SDL LiveContent documentation portal on your local machine. The installer will take some time to install the package because SDL LiveContent needs to re-index all the documents it is loading. When finished:

1. Open a browser.
2. Enter `http://localhost:4444/LiveContent/` (the default SDL LiveContent address).
3. Log in as **admin**, password **admin**.
4. The **List Publications** contains documentation for the most recent version of each SDL product.

16.3 Updating the offline documentation portal

Update your offline documentation portal extracting the`.pac` file and running your installer again.

Steps to execute

1. Log on to the SDL Tridion World Web site at http://www.sdltridionworld.com.
2. Download the offline documentation ZIP.
3. Extract only the updated `.pac` and compression file (or files, starting at `.Z01`) to the folder on your local machine where you first downloaded the offline documentation ZIP, or extract all the files including the installer executable.
4. Delete any existing `.pac` and `.Z01` files.
5. Run the `install.exe` and follow the instructions on the screen.

Result

The installer installs the offline documentation. The installer will take some time to install the package because SDL LiveContent needs to re-index all the documents it is loading.

16.4 API documentation

The documentation portal contains all documentation except API documentation. To get the API documentation, you need to go to SDL Tridion World Web site and download Windows help files (.NET APIs) and Javadoc (Java APIs).

Steps to execute

1. Log on to http://www.sdltridionworld.com.
2. Go to the **Downloads** > **Documentation** > **SDL Tridion** 2013 section.
3. Download the SDL Tridion 2013 documentation.

Chapter 17 Running an unattended install of the Content Manager server

You can run the installer executable from the command prompt, specifying a number of switches so that it can run and complete its task unattended. Running an unattended install of the Content Manager server is useful if you need to install Content Manager on a large number of machines.

Requirements

If you want to perform an unattended install of the Content Manager server on a computer with User Account Control (UAC) enabled, it is recommended that you perform the installation from an elevated command prompt in order to avoid the display of a UAC Confirmation Dialogue.

Steps to execute

1. If you have previously run an unattended install on this machine, it is recommended to clean the installer logs first. To do so, navigate to the following location:
   - the subdirectory `SDL\Tridion\Logs\` of the path indicated by the `%PROGRAMDATA%` environment variable (defaults to `c:\ProgramData\`).
2. Delete all files in this location.
3. Open a command prompt, navigate to the folder `Content Manager\` on the SDL installation media, and enter a command in the following format:

   ```
   SDLTridion2013CM.exe -s [<param1>=<value1> [<param2>=<value2> [...]]]
   ```

   Note: The installer automatically reboots the machine as the final step of the installation process.

   The following example is the most basic call (written on one line):

   ```
   SDLTridion2013CM.exe -s
   ```

   Running the command installs Content Manager server with default settings. It does not display a wizard or any graphical user interface, nor does it provide any feedback on screen.

   Note: For a full list of unattended install options, see Unattended install switches and parameters (see page 282)

4. You can verify that the unattended installation finished successfully from the Windows Start menu (see step 4) or from a command prompt (see step 5).

5. To verify that the unattended installation finished successfully:

    a. From the Windows Start menu, access the your list of installed software components in the Windows Control Panel.

    b. Select **SDL Tridion** 2013 and select **Change**.

    c. In the overview of packages and features, the ones that have their check box selected are installed. If all check boxes are selected *and* the **Next** button is enabled, installation was not successful.

6. To verify that the unattended installation finished successfully:

    a. Open a command prompt .

    b. Navigate to the path indicated by the `%PROGRAMDATA%` environment variable (defaults to `c:\ProgramData\` and then to the subdirectory `SDL\Chainer\Logs\`.

    c. Find files with names that start with `Install-` and end with a timestamp indication corresponding to the moment you started up the unattended installation, or later. The string between `Install-` and the timestamp indicates the specific software component of which this is an installation log. For example, `Install-SDL Tridion Template Builder 2013-2010-11-16T144320.log` is an installation log for Template Builder, dated 16 November 2013 at 2:14:43.20 PM.

    d. Open each of these files in turn and search for the string `return value 3`. If you do *not* find this string in the log file, the installation of that software component was successful. If you *do* find this string in the log file, the installation of that software component failed and its installation has been undone. Examine the text surrounding this string to get information on the nature of the failure.

17.1 Unattended install switches and parameters

This section describes the switches and parameters you can use when running an unattended install of the Content Manager Server.

The basic command syntax for running an unattended install is:

```
SDLTridion2013CM.exe -s [-log "logpath\logfilename.log" ] params
```

where:

- *logpath* is a path to a custom log location
- *logfilename* is a custom filename base for the log file
- *params* is a sequence of zero or more parameters of the form `<param>=<value>`.

If you omit the `-log` switch, the installer creates a log file in the following location:

- the subdirectory `SDL\Tridion\Logs\` of the path indicated by the `%PROGRAMDATA%` environment variable (defaults to `c:\ProgramData\`).

The following list shows the named parameters you can use for the Content Manager installation. If a parameter has a default value, it may be omitted. Default values correspond as much as possible to the defaults suggested during earlier installation steps. To specify an empty value, simply write, say, `Foo=` or `Foo=""`.

ACCEPT_EULA

Default value: none (mandatory parameter if you do not have all prerequisites installed)

Description: If set to `true`, automatically accepts all EULAs (End User License Agreements) of any third-party software that the installer will install.

Note: By setting this property to `true`, you indicate that you have read and understood all EULAs for any third-party software that the installer may install. For a full list of such software, refer to Content Manager server prerequisites (see page 25).

DB_NAME

Default value: `tridion_cm`

Description: Name of the Content Manager database

DB_PASSWORD

Default value: none (mandatory parameter)

Description: Password to access the Content Manager database

DB_SERVER

Default value: `(local)`

Description: Name of the Content Manager database server

DB_TYPE

Default value: `MSSQL`

Description: Type of the Content Manager database server. The only other possible value is `ORACLE`

DB_USER

Default value: `TCMDBUser`

Description: The name of the Content Manager database user

SYSTEM_ACCOUNT_NAME

Default value: `MTSUser`

Description: The identity of the user that runs the COM+ application. You created this user in Creating a system user (MTSUser) for Content Manager (see page 16)

SYSTEM_ACCOUNT_DOMAIN

Default value: the machine's domain name, or, if no domain exists, computer name

Description: The domain of the user specified under `SYSTEM_ACCOUNT_NAME`.

`SYSTEM_ACCOUNT_PASSWORD`

Default value: none (mandatory parameter)

Description: The password of the user specified under `SYSTEM_ACCOUNT_NAME`.

`TM_TRANSLATIONSYSTEM_TYPE`

Default value: `Translation Manager System`

Description: The translation system Translation Manager connects to (SDL TMS or SDL WorldServer). The only other possible value is `WorldServer`.

`TM_TRANSLATIONSYSTEM_ADDRESS`

Default value: empty

Description: The URL of SDL TMS or SDL WorldServer.

`TM_TRANSLATIONSYSTEM_PORT`

Default value: `80`

Description: The port of the URL of SDL TMS or SDL WorldServer.

`TM_TMS_PROXY_ENABLED`

Default value: empty (mandatory parameter if installing Translation Manager and if connecting)

Description: Set to `TRUE` if you want to enable a proxy setting for SDL TMS, to `FALSE` otherwise.

`TM_TMS_PROXY_ADDRESS`

Default value: empty (mandatory parameter)

Description: The address of the proxy server that SDL TMS.

`TM_TMS_PROXY_PORT`

Default value: empty (mandatory parameter)

Description: The port of the proxy server that SDL TMS uses.

`TM_WORLDSERVER_USERNAME`

Default value: empty

Description: The name of a user authorized to connect to SDL WorldServer.

`TM_WORLDSERVER_PASSWORD`

Default value: empty (mandatory parameter)

Description: The password for the user authorized to connect to SDL WorldServer.

`TM_DB_TYPE`

Default value: `MSSQL`

Description: Type of the Translation Manager database server. The only other possible value is `ORACLE`

`TM_DB_SERVER`

Default value: `(local)`

Description: The name of the database server that contains your Translation Manager database.

`TM_DB_NAME`

Default value: `Tridion_TranslationManager`

Description: The name of the Translation Manager database.

`TM_DB_USER`

Default value: `TMUser`

Description: The name of a user authorized to connect to the Translation Manager database.

`TM_DB_PASSWORD`

Default value: none (mandatory)

Description: The password of a user authorized to connect to the Translation Manager database.

`WEB_DESCRIPTION`

Default value: `"SDL Tridion 2011"`

Description: The description of the Content Manager Explorer Web site.

`WEB_IP`

Default value: empty, meaning all unassigned IP addresses

Description: The IP address of the Content Manager Explorer Web site.

`WEB_PORT`

Default value: `80`

Description: The port of the Content Manager Explorer Web site.

`WEB_HEADER`

Default value: empty, meaning no host header is provided

Description: The host header of the Content Manager Explorer Web site.

`LICENSE_PATH`

Default value: none; if you do not specify the license path here, you can specify it after installation, refer to Licenses (see page 4)

Description: The full path and filename of the Content Manager license file.

`CD_LICENSE_PATH`

Default value: none; if you do not specify the license path here, you can specify it after installation, refer to Licenses (see page 4)

Description: The full path and filename of the Content Delivery license file.

`OE_DB_NAME`

Default value: `tridion_cm_email`

Description: Name of the Outbound E-mail database

`OE_DB_PASSWORD`

Default value: none (mandatory)

Description: Password to access the Outbound E-mail database

`OE_DB_SERVER`

Default value: `(local)`

Description: Name of the Outbound E-mail database server

`OE_DB_TYPE`

Default value: `MSSQL`

Description: Type of the Outbound E-mail database server. The only other possible value is `ORACLE`.

`OE_DB_USER`

Default value: `TMSDBUser`

Description: The name of the Outbound E-mail database user.

`CdWindowsServices_SelectedFeatures`

Default value: `(All)`—all features

Description: Indicates whether to install Windows Services (Transport Service and Monitoring). The only other possible value is an empty string, meaning that no Windows Services are installed.

`cm_SelectedFeatures`

Default value: `(All)`—all features

Description: Indicates whether to install Content Manager features. The only other possible value is an empty string, meaning that Content Manager features are only installed if another, selected feature depends on them.

`CMECore_SelectedFeatures`

Default value: `(All)`—all features

Description: Indicates whether to install Content Manager Explorer core features. This does not install the GUI itself, but only the extensible GUI framework. The only other possible value is an empty string, meaning that Content Manager Explorer core features are only installed if another, selected feature depends on them.

`CMEGui_SelectedFeatures`

Default value: `(All)`—all features

Description: Indicates whether to install Content Manager Explorer features. This installs the GUI itself, including the underlying core framework. The only other possible value is an empty string, meaning that Content Manager Explorer features are only installed if another, selected feature depends on them.

`Documentation_SelectedFeatures`

Default value: `(All)`—all features

Description: Indicates whether to install Documentation. The only other possible value is an empty string, meaning that Documentation is not installed.

ExternalContentLibrary_SelectedFeatures

Default value: `(All)`—all features

Description: Indicates whether to install External Content Library. The only other possible value is an empty string, meaning External Content Library is not installed.

OutboundEmail_SelectedFeatures

Default value: `(All)`—all features

Description: Indicates whether to install Outbound E-mail features. The only other possible value is an empty string, meaning that Outbound E-mail features are only installed if another, selected feature depends on them.

SDLTridionUGC_SelectedFeatures

Default value: `(All)`—all features

Description: Indicates whether to install the User-Generated Content (UGC) application. The only other possible value is an empty string, meaning that the UGC application is only installed if another, selected feature depends on it.

SpellChecker_SelectedFeatures

Default value: `(All)`—all features

Description: Indicates whether to install spell-checking features in the Content Manager Explorer GUI. The only other possible value is an empty string, meaning that spell-checking features are only installed if another, selected feature depends on them.

TcmSearch_SelectedFeatures

Default value: `(All)`—all features

Description: Indicates whether to install search functionality for Content Manager Explorer. The only other possible value is an empty string, meaning that search functionality is only installed if another, selected feature depends on them.

TemplateBuilder_SelectedFeatures

Default value: `(All)`—all features

Description: Indicates whether to install the Template Builder application. The only other possible value is an empty string, meaning that the Template Builder application is only installed if another, selected feature depends on it.

TranslationManager_SelectedFeatures

Default value: `(All)`—all features

Description: Indicates whether to install the Translation Manager application. The only other possible value is an empty string, meaning that the Translation Manager application is only installed if another, selected feature depends on it.

INSTALLLOCATION

Default value: `C:\Program Files (x86)\Tridion\`.

Description: Indicates where to install Content Manager server.

WEBLOCATION

Default value: `C:\Program Files (x86)\Tridion\` web (on a 64-bit system).

Description: Indicates where to install the Content Manager Explorer Web site.

17.2 Unattended uninstall or repair

Add the `-repair` or `-uninstall` switch to the installer executable call to repair or uninstall your Content Manager Server. To uninstall specific Content Manager server software components, omit the `-uninstall` switch and specify one or more of the uninstall parameters.

The following command repairs your previously installed Content Manager server installation, without showing a user interface:

```
SDLTridion2013CM.exe -s -repair
```

Note: Performing a repair preserves your configuration settings for the product, but resets all custom settings applied to services, IIS and COM+.

The following command uninstalls your previously installed Content Manager server completely, without showing a user interface:

```
SDLTridion2013CM.exe -s -uninstall
```

You can also uninstall specific Content Manager server software components by leaving out the `-uninstall` switch, and instead specifying a specific parameter for each software component you want to uninstall, set to the value `(All)`.

For example, the following command uninstalls only Template Builder, without showing a user interface:

```
SDLTridion2013CM.exe -s TemplateBuilder_RemovedFeatures=(All)
```

17.3 Unattended uninstall switches and parameters

This section describes the switches and parameters you can configure when running an unattended uninstall of Content Manager server features.

Start by using the basic command syntax:

```
SDLTridion2013CM.exe -s [-log "<logpath>\<logfilename>.log" ]<params>
```

where

- *<logpath>* is a path to a custom log location
- *<logfilename>* is a custom filename base for the log file
- *<params>* is a sequence of zero or more parameters of the form `<param>=<value>`, denoting the software component(s) you want to uninstall.

If you omit the `-log` switch, the installer creates a log file in the following location:

- the subdirectory `SDL\Tridion\Logs\` of the path indicated by the `%PROGRAMDATA%` environment variable (defaults to `c:\ProgramData\`).

The following list shows the named parameters you can use for the uninstallation. Set the value of the parameter to the value `(All)`.

`CdWindowsServices_RemovedFeatures`

Uninstalls Windows Services (Transport Service and Monitoring).

`cm_RemovedFeatures`

Uninstalls Content Manager.

`CMECore_RemovedFeatures`

Uninstalls the framework for extending the GUI (not the user interface).

`CMEGui_RemovedFeatures`

Uninstalls Content Manager Explorer user interface.

`Documentation_RemovedFeatures`

Uninstalls the documentation.

`ExternalContentLibrary_RemovedFeatures`

Uninstalls External Content Library.

`OutboundEmail_RemovedFeatures`

Uninstalls Outbound E-mail.

`SDLTridionUGC_RemovedFeatures`

Uninstalls the User Generated Content.

`SpellChecker_RemovedFeatures`

Uninstalls spell-checking features in the Content Manager Explorer GUI.

`TcmSearch_RemovedFeatures`

Uninstalls search functionality for Content Manager Explorer.

`TemplateBuilder_RemovedFeatures`

Uninstalls Template Builder.

`TranslationManager_RemovedFeatures`

Uninstalls Translation Manager.