OpenText™ Documentum™ Content Management

Transformation Services Administration Guide

Configure and administer OpenText Documentum Content Management (CM) Transformation Services applications and manage the transformation of the documents, audio/video, media, and XML formats.

EDCCT250400-AGD-EN-01

OpenText™ Documentum™ Content Management
Transformation Services Administration Guide
EDCCT250400-AGD-EN-01
Rev.: 2025-Oct-21

This documentation has been created for OpenText™ Documentum™ Content Management CE 25.4.
It is also valid for subsequent software releases unless OpenText has made newer documentation available with the product, on an OpenText website, or by any other means.

Open Text Corporation

275 Frank Tompa Drive, Waterloo, Ontario, Canada, N2L 0A1

Tel: +1-519-888-7111
Toll Free Canada/USA: 1-800-499-6544 International: +800-4996-5440
Fax: +1-519-888-0677
Support: https://support.opentext.com
For more information, visit https://www.opentext.com

Disclaimer

No Warranties and Limitation of Liability

Every effort has been made to ensure the accuracy of the features and techniques presented in this publication. However, Open Text Corporation and its affiliates accept no responsibility and offer no warranty whether expressed or implied, for the accuracy of this publication.

Table of Contents

Chapter 1

Overview

This guide provides step-by-step procedures for configuring and administering OpenText Documentum Content Management (CM) Transformation Services, and modifying plug-ins to extend the functionality of Transformation Services. This guide is intended for system operator or system administrator and assumes they have basic understanding of the Windows operating system.

1.1 Transformation Services

Transformation Services is a suite of Server products that perform transformations and analysis on repository content. The Transformation Services functionality is available through Documentum client applications. Transformation Services contains the following products:

- OpenText Documentum Content Management (CM) Transformation Services - Documents
- OpenText Documentum Content Management (CM) Transformation Services - Media
- OpenText Documentum Content Management (CM) Transformation Services - Audio/Video
- OpenText Documentum Content Management (CM) XML Transformation Services

OpenText Documentum Content Management (CM) Transformation Services provides the following functionalities:

Transformations

Transformation Services carries out a wide range of conversions of documents, media, and audio/video content (for example, Microsoft Office documents into PDF and HTML formats). In addition, you can apply watermarks, PDF overlays, headers, and footers to the transformed PDFs. Media files (images) can be transformed from one format into another with a wide range of options (such as changes in the resolution, orientation, and so on). Audio/video files can be transformed from one format into another with options such as updating frame rate, bit rate, resolution, encoding, and so on, in the target files.

Metadata Analysis and Attribution

Transformation Services extracts attributes from content. Examples of extracted attributes are height and width of image files, and author and subject of documents. The extracted attributes are stored as metadata or rendition attributes. Some rendition attributes can be mapped to object-level attributes.

Enhanced Content Previews
Transformation Services generates thumbnails and storyboards to enhance the previewing experience of Microsoft Office documents, PDF, images, and video files that are available in the repository.

Rendition Management
Transformation Services stores different formats of a repository file as renditions or related objects. Renditions are alternate formats of content that share the same object attributes and security. Related objects are complete objects on their own with independent metadata and security.

1.2 Transformation Services - Documents

Transformation Services - Documents is a server software that generates copies of documents in Portable Document Format (PDF), PDF text, PostScript, HyperText Markup Language (HTML), and so on. You can use Documentum client applications to request such a copy (called a rendition) of a document stored in a Documentum repository. The transformation capabilities of Transformation Services - Documents include PDF assembly, header/footer creation, security, textual watermarking, content overlays, and so on.

In addition, Transformation Services - Documents can provide PDF transformations from other file formats such as image and email formats.

Some Documentum client applications may not provide both PDF and HTML renditions; consult the documentation for the client you are using.

1.2.1 How Transformation Services - Documents works

The following sections detail how Transformation Services - Documents creates PDF and HTML renditions.

1.2.1.1 How Transformation Services - Documents creates PDF renditions

The following steps describe how Transformation Services - Documents produces PDF renditions in response to client requests:

1. In response to client requests (asynchronous transformation), a rendition request is sent to OpenText™ Documentum™ Content Management Server.
2. OpenText Documentum Content Management (CM) Server stores the request in a queue called dm_queue.
3. Transformation Services Server polls dm_queue at predefined intervals, and retrieves any pending requests.
4. When Transformation Services - Documents finds a PDF rendition request in the queue, it checks the format of the document and generates a PDF rendition based on the default profiles.

5. Documentum CM Server adds the PDF rendition to the source document in the repository.

6. After a rendition is added to the source document, users can view the rendition.

Figure 1-1 illustrates this rendition process.

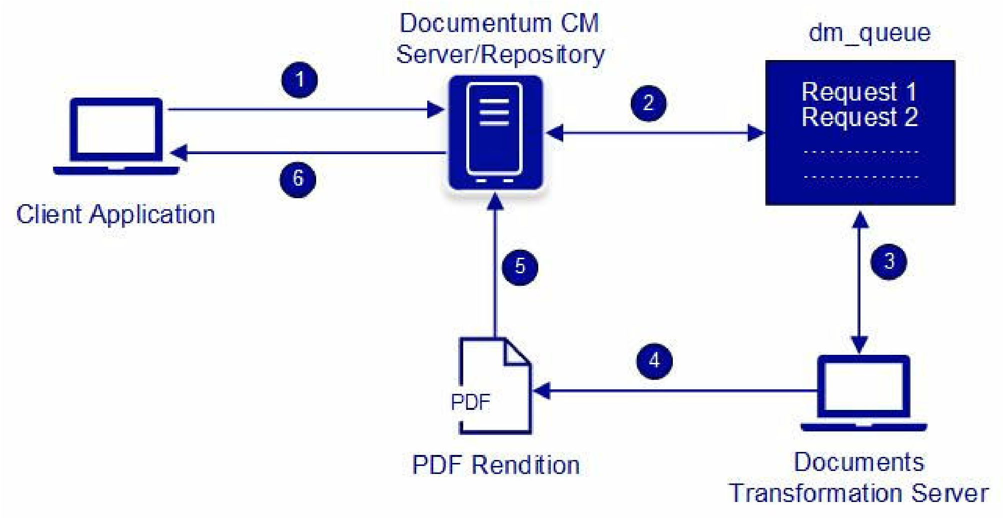


Figure 1-1: Processing a request for a PDF rendition in Transformation Services - Documents

For transformation of PDF documents greater than 1 GB using Doc4 plug-in, you need to increase the RAM size for Transformation Services server to 3 GB or higher as transformation of large PDF files requires higher memory.

To increase the RAM for Transformation Services server:

1. Stop the Transformation Services server services.

2. Open `regedit.exe`.

3. Navigate to `My Computer` > `HKEY_LOCAL_MACHINE` > `SYSTEM` > `CurrentContrlSet` > `Services` > `DocumentumCTS` > `Parameters`.

4. Modify `AppParameters`.

5. Increase the values of `-Xms` and `-Xmx` and then save it.

6. Start the Transformation Services server services.

The transformation time is higher for 1 GB or higher documents. You can enable the Partial Reading parameter to increase the transformation time. To enable Partial Reading parameter, navigate to *<Documentum Content Transformation Services_HOME>*`/config/doc4/` and open the `doc4.xml` file. Set the value of *<PARTIAL_READING>* parameter to `TRUE` or `YES` and restart Transformation Services.

Note: The maximum size of the document transformation supported is 1.5 GB.

1.2.1.2 How Transformation Services - Documents creates HTML renditions

The following steps describe how a Transformation Services - Documents server host produces HTML renditions in response to client requests:

1. A rendition request is sent to the Documentum CM Server.
2. Documentum CM Server stores the request in a queue called dm_queue.
3. Transformation Services Server polls dm_queue at predefined intervals, and retrieves any pending requests.
4. When Transformation Services - Documents finds an HTML rendition request in the queue, it checks the format of the document. Transformation Services - Documents invokes the corresponding plug-in to process the file.
5. The plug-in saves an HTML version of the document.
   - If the result is a single HTML file, Transformation Services - Documents sends this file to the Documentum CM Server.
   - If the result consists of multiple files (for example, HTML, graphics, and XML files), Transformation Services - Documents creates a ZIP file containing these files, and sends the ZIP file to Documentum CM Server. The ZIP file includes any folders the application creates to contain these files.
6. Documentum CM Server adds the HTML or ZIP file to the repository as a rendition of the source document.
7. After a rendition is added to the source document, users can view the rendition.

Figure 1-2 illustrates the rendition process.

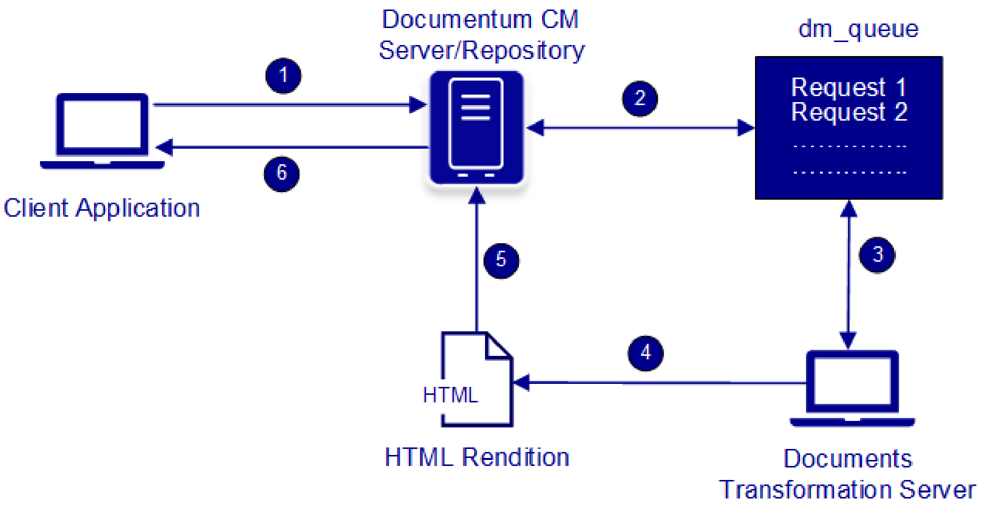


Figure 1-2: Processing a request for an HTML rendition in Transformation Services - Documents Server

1.2.2 Installed components

Figure 1-3 shows a typical Transformation Services - Documents installation on the Documentum platform.

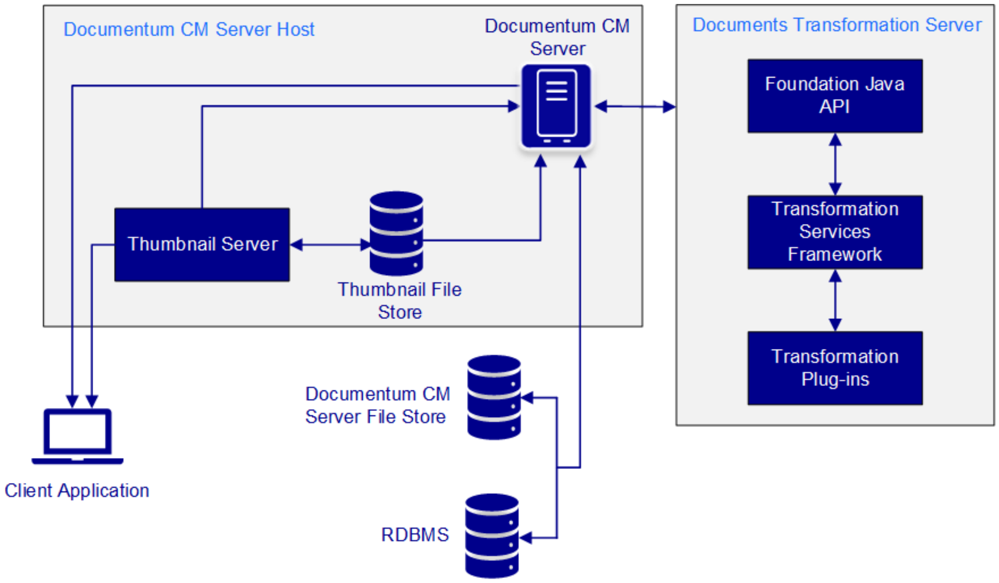


Figure 1-3: Typical Transformation Services - Documents installation

1.2.3 Transformation Services - Documents plug-ins

The Transformation Services - Documents plug-ins enable users to transform a variety of files from one format to another. For example, from DOC to PDF or HTML.

The standard Transformation Services - Documents plug-ins are as follows:

- `Doc2`: Transforms Adobe PDF files into the XML format. This transformation comes into use while using the xCP Viewer Search feature.
- `Doc3`: Transforms Adobe PostScript files into Adobe PDF files. In addition, it supports the creation of PDF-X, PDF-A flavours from a PDF source.
- `Doc4`: Creates PDF from PDF source files with advanced options such as overlay, header-footer, signature, and so on.
- `Doc5`: Supports page-reordering for PDF and multi-page TIFF files.

  Note: To support TIFF file, Ghostscript installation is mandatory.

- `Doc6`: Transforms Microsoft Word, Excel, PowerPoint, and Project, text files, and so on, to PDF format. In addition, it transforms PDF to HTML format.
- `Doc7`: Supports mail attachment extraction and storage as related objects.
- `Doc8`: Transforms PDF files to PS format.
- `Doc9`: Transforms PDF files to HTML format.
- `Doc11`: Blazon server transforms Microsoft Office (PowerPoint, Word, Excel, Visio, Project) and AutoCAD (DWG, DXF, DGN, DWF, DWFX) files to PDF. It is an optional plug-in for customers who require high fidelity output and choose to install Blazon server. If you enable Doc11, it is chosen as the plug-in for transformation of Microsoft Office files. Else, Doc6 is the default plug-in for Microsoft Office formats.

  Note: To use this plug-in, install the supported version of Blazon Server (Job Processor component) on the Transformation Services - Documents server. The product *Release Notes* on OpenText My Support (support.opentext.com) contains the information about the supported versions.

- `Doc12`: Transformation Publication-Service of OpenText Intelligent Viewing is used as a rendition engine to transform the Microsoft Word, Excel, PowerPoint, and Project files, text files, and so on, to PDF format.
- `PDFStoryboard`: Generates thumbnails, low-resolution renditions, and storyboards for PDF files.
- `Image3`: Generates thumbnails, low-resolution renditions, and storyboards for PDF files.
- `XMP`: Extracts metadata from document (PDF) format.

During an installation of Transformation Services - Documents, all standard plug-ins are installed.

Notes

- When you import DWG format files, ensure to select AutoCAD DWG format.
- Transformation Services relies on one or more rendition engines to meet content transformation use-cases. Transformation Services has proven integration with OpenText Blazon as an optional add-on. Customers with loss-less Microsoft Office conversion use-cases are encouraged to adopt Blazon.

1.3 Transformation Services - Media

Transformation Services - Media is the server software that integrates with Documentum CM Server to perform analysis and transformation activities for media file formats. Thumbnail, storyboard, and low-resolution renditions are generated automatically. Transformations can be triggered automatically by certain events, such as when an object is versioned, or as requested by users.

1.3.1 Installed components

Figure 1-4 shows a typical Transformation Services - Media installation on the Documentum platform.

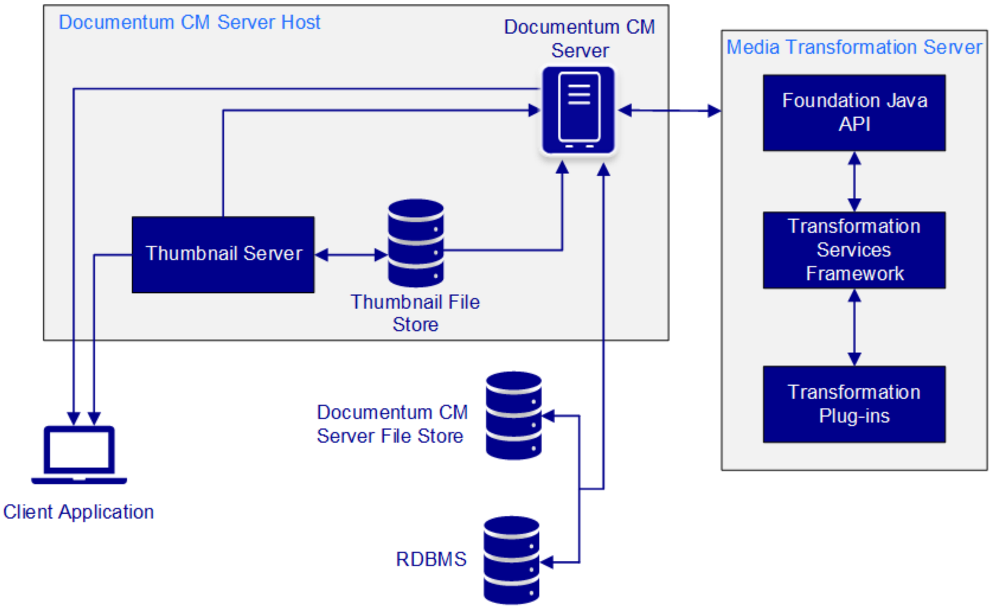


Figure 1-4: Typical Transformation Services - Media installation

1.3.2 Transformation Services - Media plug-ins

The Transformation Services - Media plug-ins identify and extract media-specific properties, such as height, width, color mode, and compression, and generate thumbnails and low-resolution renditions of objects. In addition, the plug-ins enable the transformation of media files from one format to another (for example, from TIFF to JPEG) and provide users with the ability to perform transformations such as resizing, flipping, rotating an image, and so on.

The standard Transformation Services - Media plug-ins are as follows:

- `PowerPoint1`: Generates thumbnails, low-resolution renditions, storyboards, and slide objects for PowerPoint files. This plug-in is available Out-Of-The-Box (OOTB). In addition, it is capable of the assembly feature.
- `PowerPoint2`: Generates thumbnails, low-resolution renditions, storyboards, and slide objects for PowerPoint files. It is an optional plug-in provided only to those for customers who require high fidelity. In addition, it is capable of the assembly feature.

  Note: To use this plug-in, install Microsoft PowerPoint 2007 SP3 (32-bit) on the Transformation Services - Media server that is installed on Windows Server 2008 R2 SP1 (64-bit) operating system.
- `Image1`: Processes images and other file formats. Generates thumbnails, generates low-resolution renditions, extracts properties, and performs transformations such as resize and transformTo.
- `Image2`: Processes images and other file formats. Generates thumbnails, generates low-resolution renditions, extracts properties, and performs transformations such as resize.
- `Image3`: Processes images and other file formats. Generates thumbnails, generates low-resolution renditions, and extracts properties of large image files. In addition, this plug-in generates thumbnails, generates low-resolution renditions, extracts properties, and creates storyboards for PDF files.
- `Image4`: Processes images and other file formats.
- `PDFStoryboard`: Generates thumbnails, low-resolution renditions, and storyboards for PDF files.
- `XMP`: Extract metadata from image formats and PDF files.
- `EXIF`: Extracts EXIF metadata from image files that have embedded EXIF metadata.

During an installation of Transformation Services - Media, all standard plug-ins are installed.

1.4 Transformation Services - Audio/Video

Transformation Services - Audio/Video provides transformation capability for audio and video file formats.

1.4.1 Installed components

Figure 1-5 shows a typical Transformation Services - Audio/Video installation on the Documentum platform.

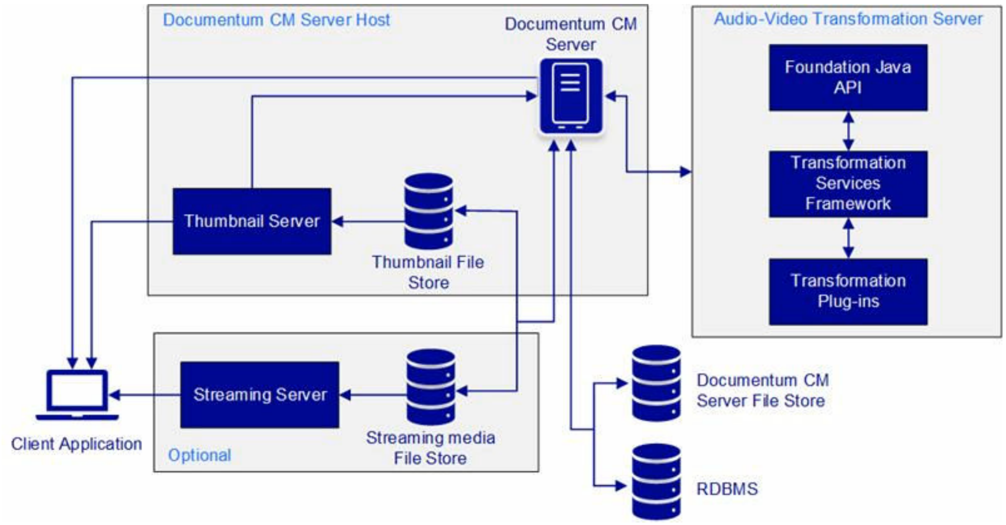

Documentum CM Server Host
Documentum CM Server
Audio-Video Transformation Server
Foundation Java API
Transformation Services Framework
Transformation Plug-ins
Thumbnail Server
Thumbnail File Store
Streaming Server
Streaming media File Store
Optional
Client Application
Documentum CM Server File Store
RDBMS

Figure 1-5: Typical Transformation Services - Audio/Video installation

1.4.2 Transformation Services - Audio/Video plug-ins

The Transformation Services - Audio/Video plug-ins identify and extract media-specific properties and generate thumbnails, storyboards, and low-resolution renditions of objects. In addition, the plug-ins enable the transformation of files from one format to another (for example, from AVI to MP4).

Video1 is the standard Transformation Services - Audio/Video plug-in. It identifies and extra Transformation Services - Media-specific properties and generates thumbnails, storyboards, and low-resolution renditions of objects. In addition, this plug-in facilitates the conversion of one video format to another.

During an installation of Transformation Services - Audio/Video, all standard plug-ins are installed.

1.5 Documentum XML Transformation Services

XML Transformation Services is server software that generates copies of documents in Portable Document Format (PDF), HyperText Markup Language (HTML), and Compiled HTML Help (CHM). You can use Documentum client applications to request such a copy (called a rendition) of a document stored in a Documentum repository.

Designed primarily for technical publication organizations, XML Transformation Services is part of a seamless approach to storing, managing, and transforming XML content in either DocBook or DITA formats.

1.5.1 How XML Transformation Services works

The following sections detail how XML Transformation Services creates renditions in PDF, HTML, and CHM formats.

1.5.1.1 How XML Transformation Services creates PDF renditions

The following steps describe how XML Transformation Services produces PDF renditions in response to client requests:

1. The client application sends the rendition request to the Documentum CM Server.

   Note: XML Transformation Services modifies the document object, so it cannot process locked (checked out) documents.

2. Documentum CM Server stores the request in a queue called dm_queue.
3. Transformation Services Server polls dm_queue at predefined intervals, and retrieves any pending requests.
4. When XML Transformation Services finds a PDF rendition request in the queue, it checks the input_doctype_family name token to distinguish between a DITA and DocBook file. The request is passed to the appropriate transform handler.
5. The transform handler generates a PDF rendition based on what the default profiles specify.
6. Documentum CM Server adds the PDF renditions to the object in the repository.

   An object represents a parent file, with all its associated graphics, text, and renditions, in the repository. Each time a transformation is performed, an additional rendition is created for the source file.

7. After a rendition has been added to the object, users can view the rendition from the client application.

Figure 1-6 illustrates this rendition process.

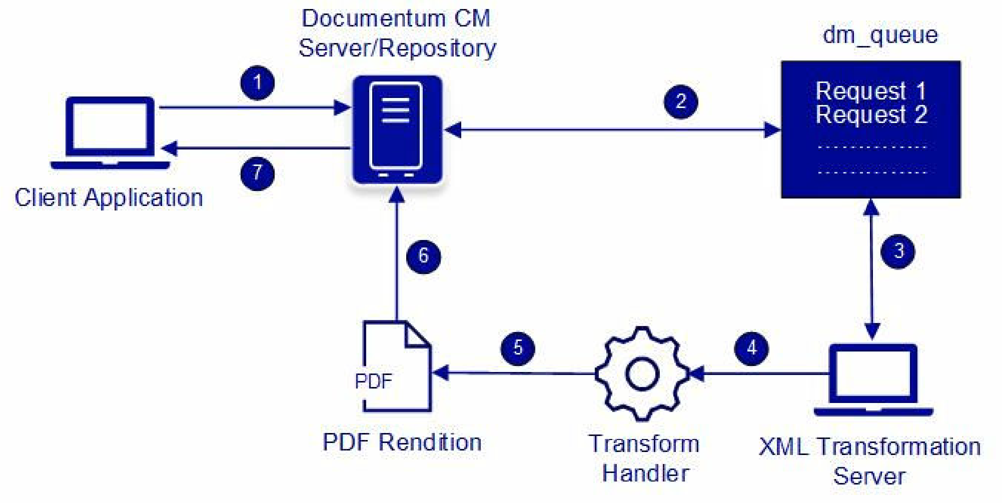


Figure 1-6: XML Transformation Services processing a request for a PDF rendition

1.5.1.2 How XML Transformation Services creates HTML renditions

The following steps describe how XML Transformation Services produces HTML renditions in response to client requests:

1. The client application sends the rendition request to the Documentum CM Server.

   Note: XML Transformation Services cannot process documents that are locked (checked out from the repository).

2. Documentum CM Server stores the request in a queue called dm_queue.
3. Transformation Services Server polls dm_queue at predefined intervals, and retrieves any pending requests.
4. When XML Transformation Services finds an HTML rendition request in the queue, it checks the input_doctype_family name token to distinguish between a DITA and DocBook file. The request is passed to the appropriate transform handler.
5. XML Transformation Services invokes the corresponding application to open the file and then saves an HTML version.

   - If the result is a single HTML file, XML Transformation Services sends this file to the Documentum CM Server.
   - If the result consists of multiple files (for example, HTML, graphics, and XML files), XML Transformation Services creates a ZIP file containing these files, and sends the ZIP file to Documentum CM Server. The ZIP file includes any folders the application creates to contain these files.

6. Documentum CM Server adds the HTML or ZIP file to the repository as a rendition of the document object.

   An object represents a parent file, with all its associated graphics, text, and renditions, in the repository. Each time a transformation is performed, an additional rendition is created for the source file.

7. After a rendition has been added to the object, users can view the rendition from the client application.

Figure 1-7 illustrates this rendition process.

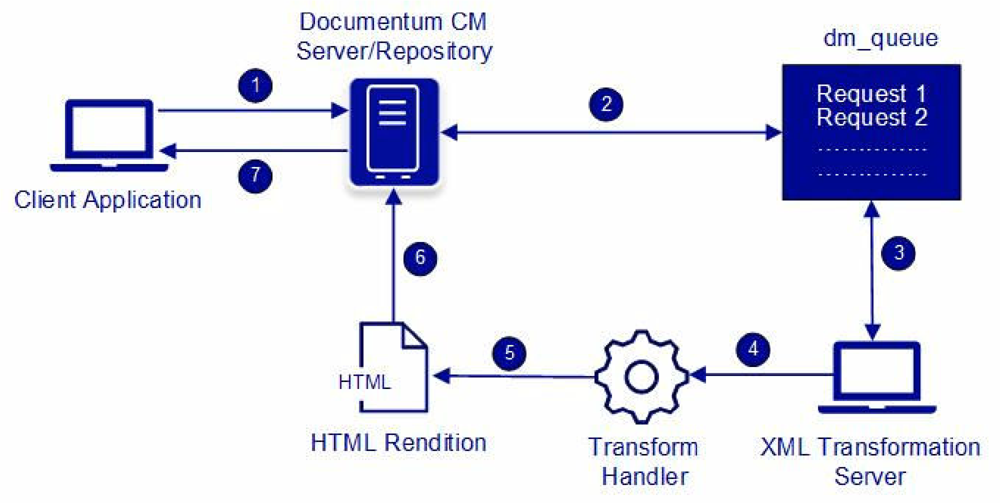


Figure 1-7: XML Transformation Services processing a request for an HTML rendition

1.5.1.3 How XML Transformation Services creates CHM renditions

The following steps describe how XML Transformation Services produces CHM renditions in response to client requests:

1. The client application sends the rendition request to the Documentum CM Server.

   Note: XML Transformation Services cannot process documents that are locked (checked out from the repository).

2. Documentum CM Server stores the request in a special queue called dm_queue.
3. Transformation Services Server polls dm_queue at predefined intervals, and retrieves any pending requests.
4. When XML Transformation Services finds a CHM rendition request in the queue, it checks the input_doctype_family name token to distinguish between a DITA and DocBook file. The request is passed to the appropriate transform handler.format of the document.

5. XML Transformation Services invokes the corresponding application to open the file and then saves an CHM version.

   - If the result is a single CHM file, XML Transformation Services sends this file to the Documentum CM Server.
   - If the result consists of multiple files (for example, HTML, graphics, and XML files), XML Transformation Services creates a ZIP file containing these files, and sends the ZIP file to Documentum CM Server. The ZIP file includes any folders the application creates to contain these files.

6. Documentum CM Server adds the CHM or ZIP file to the repository as a rendition of the document object.

   An object represents a parent file, with all its associated graphics, text, and renditions, in the repository. Each time a transformation is performed, an additional rendition is created for the source file.

7. After a rendition has been added to the object, users can view the rendition from the client application.

Figure 1-8 illustrates this rendition process.

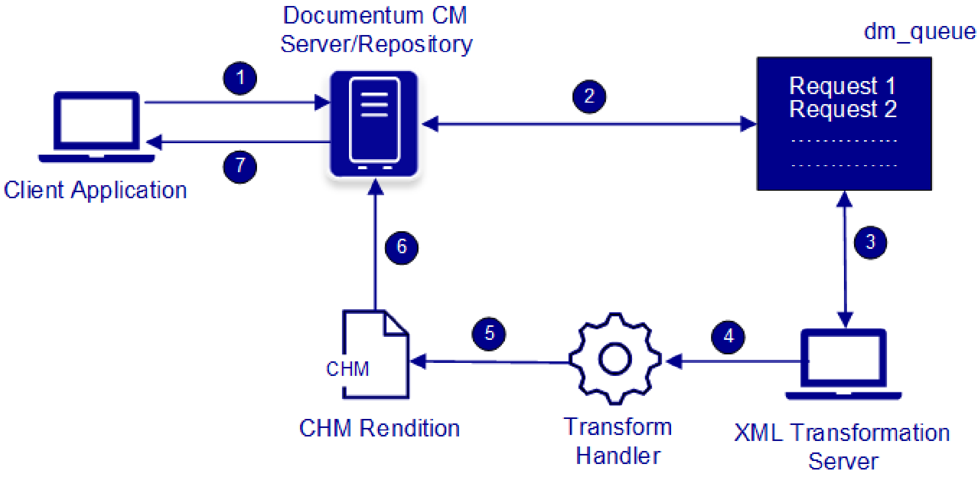


Figure 1-8: XML Transformation Services processing a request for a CHM rendition

1.5.2 XML Transformation Services plug-in

The XML Transformation Services plug-in enables users to transform a variety of files from one format to another (for example, from XML to PDF or HTML).

During an installation of XML Transformation Services, this plug-in is installed:

- XPUB - transforms XML (based on DocBook and DITA standards), DITA, and DITAMAP files to PDF, HTML, ZIP, or CHM; DITAMAP files to Java Help, Eclipse Help, RTF, and DocBook; and custom XML to PDF.

There may be prerequisite, third-party applications required for some of the standard plug-ins. The product *Release Notes* on My Support (support.opentext.com) provides more information.

1.5.3 Installed components

Figure 1-9 shows a typical XML Transformation Services installation on the Documentum platform.

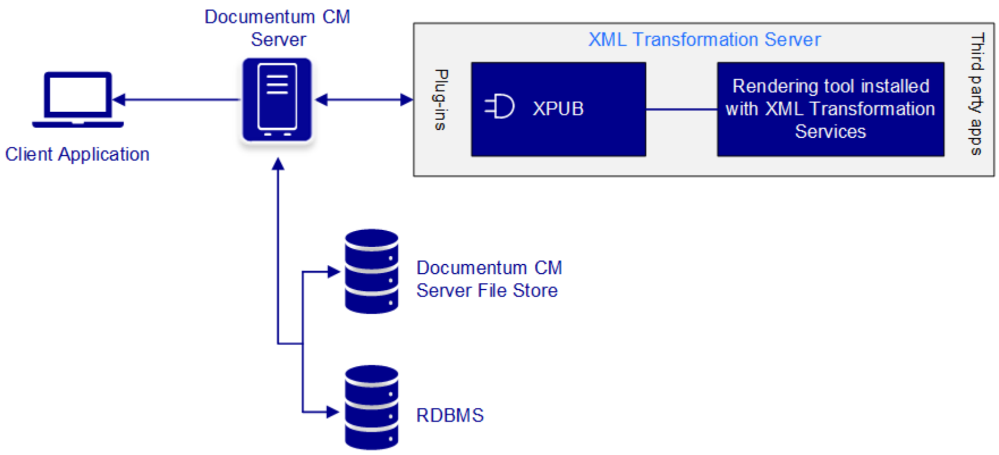

Documentum CM Server
XML Transformation Server
Plug-ins
XPUB
Rendering tool installed with XML Transformation Services
Third party apps
Client Application
Documentum CM Server File Store
RDBMS

Figure 1-9: Typical XML Transformation Services installation

1.6 Transformation Services transformations

Transforming is the act of changing a file in some way to create a new file.

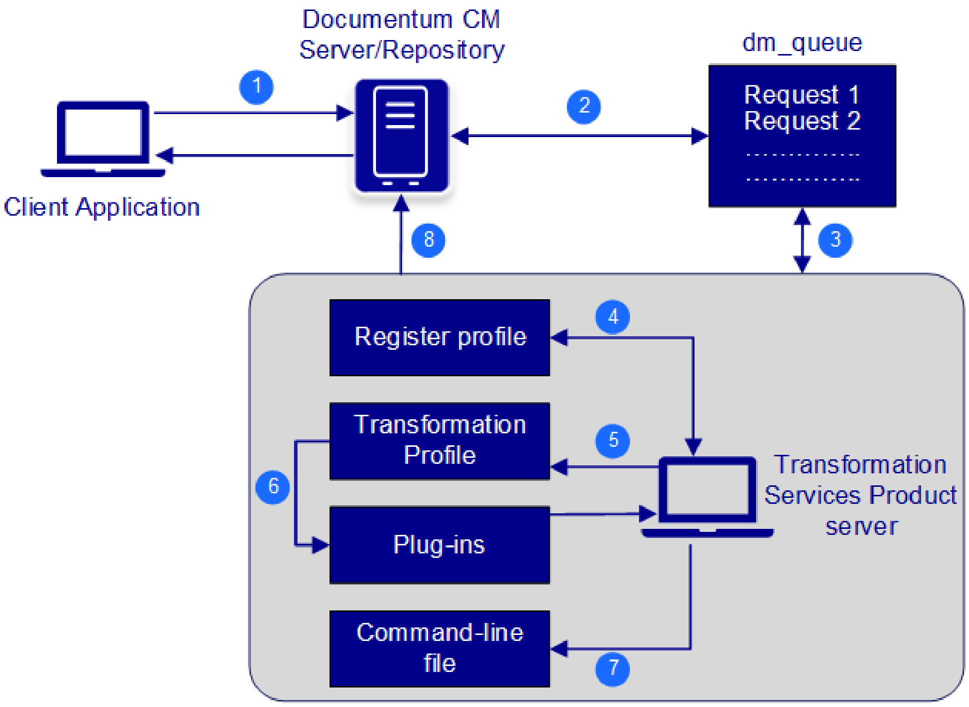
Transformations take place either as part of the registration process (described in "Automated transformations" on page 25) or as requested by users (described in "User transformations" on page 29).

1.6.1 Automated transformations

The process of executing transformations automatically upon importing or versioning the repository content is called Registration.

The overall process of registering a file in Transformation Services (depicted in Figure 1-10) is as follows:

1. A user imports or checks in a file to the repository.
2. Documentum CM Server creates a dmi_queue_item object of the event type dm_register_asset, which is added to the queue. Only content that is rich media enabled is passed to the queue.
3. The server checks the queue regularly; the default is every 10 seconds. When it finds queue items, it signs off a certain number (the default is 10 items). Both of these parameters are configurable through Documentum Administrator. "Administering and Configuring Transformation Services through Documentum Administrator" on page 31 provides more information.
4. The server starts processing the items using the register profile. It does not remove an item from the queue until the transformation is complete or has failed.
5. The server loads the transformation profile specified by the inner profile path in the register profile.
6. The server polls the plug-ins in the order specified by the transformation profile.
7. The server finds a plug-in to accept the transformation. If more than one plug-in can handle the transformation, the plug-ins are invoked in the order specified by the transformation profile.
8. The server uses the command line file to perform the transformation with the relevant plug-in.
9. The server updates the object in the repository, storing the thumbnails and low-resolution renditions (by using a OpenText™ Documentum™ Content Management Foundation Java API call) and storing media properties as attributes for each rendition (in the attributes content_attr_name and content_attr_value of dmr_content objects). It removes the dmi_queue item object from the queue.
10. The new object or renditions are available in the client application.



1. Source file is imported.
2. Rich Media enabled content is passed to queue.
3. Server signs off queue items.
4. Server processes content using register profile.
5. Server loads transformation profile specified by register.
6. Server polls plug-in(s) and finds plug-in(s) to accept transformation.
7. Server activates appropriate command-line file to perform transformation.
8. Rendition is added to content.

Figure 1-10: The registration process

The automatic transformation processes performed by Transformation Services is described in detail in the following sections.

1.6.1.1 Generating thumbnails

A thumbnail is an image that is used to represent an object in client applications. Thumbnails provide a visual cue for browsing media and enable users to identify objects quickly.

Transformation Services automatically generates a thumbnail for each object by passing the object to the appropriate plug-in. The plug-in extracts the object's properties and creates a new object by transforming the original object into a predefined thumbnail format (for example, a JPEG that is 100 x 100 pixels). Transformation Services sends the thumbnail back to Documentum CM Server as a rendition of the original object. The thumbnail's media properties are saved as attributes of that rendition.

Documentum CM Server stores thumbnails in a special file store that is shared with the OpenText™ Documentum™ Content Management Thumbnail Server, a dedicated server that delivers thumbnails directly to the browser.

1.6.1.2 Generating storyboards

Storyboards are low-resolution and low-bandwidth representations of video or multi-page objects (for example, PDFs), displayed as a sequence of JPEG thumbnails. Storyboards are generated automatically during registration (import or check in) of applicable file types.

For some video storyboards, each frame may include timecode information. The timecode is stored as a property of each JPEG image content object in the storyboard.

The product *Release Notes* on My Support (support.opentext.com) provides more information on the supported video formats for a Streaming Server.

For multi-page documents such as PowerPoint presentations, the storyboard contains a thumbnail of each slide in the presentation. Selecting one frame of an object's storyboard displays that frame in its related application as a JPEG image.

Storyboards are stored as renditions of the original dm_document object. Each JPEG image that makes up a storyboard is stored in an administrator-specified file path. *OpenText Documentum Content Management - Server Administration and Configuration Guide (EDCCS250400-AGD)* provides more information on storage of objects.

1.6.1.3 Generating low-resolution renditions

Low-resolution renditions are used to represent high-resolution objects. Low-resolution renditions are used by the client applications when a user wants to preview a high-resolution media file. Due to its smaller file size, a low-resolution rendition can be displayed more quickly than a file in its original format.

Content Transformation Server automatically generates low-resolution renditions for supported files upon registration. When the object is checked in, Content Transformation Server automatically invokes the appropriate plug-in to generate the rendition according to a predefined profile. For example, Content Transformation Server might generate a full-sized JPEG representation of the image, reduce it to a predetermined width and height, and save it back to the repository as a low-resolution rendition of the original object. Its media properties are stored as attributes of the rendition. Low resolution renditions are stored in a file path that is determined by the Documentum CM Server.

1.6.1.4 Extracting media properties

Transformation Services provides a range of plug-ins to support industry-standard file types. Some plug-ins recognize files and automatically extract media-specific properties, such as image height and width, bit rate, and audio sampling frequency.

During registration, Transformation Services first generates media properties for the high-resolution object. It then stores the properties for the object in two repeating attributes of dmr_content objects: content_attr_name and content_attr_value

Media properties are added to the object's indexed metadata (as a string value) and saved back to the repository. This provides the framework for client applications to expose media properties as part of an object's searchable metadata set. Transformation Services does the same for each rendition (thumbnail, low-resolution, and transcoded renditions).

The following table provides a sample of media properties extracted by Transformation Services for image, video, and audio files:

Table 1-1: Sample extracted media properties for Transformation Services

Image	Video	Audio
Width	Frame width	Sample width
JPEG quality	Bit rate	Sample rate
Compression	Frame rate	Number of samples
Format	Format	Format
GIF interleave	Number of frames	Number of frames
Color mode	Frame height	Number of channels
Height	Duration	Duration

1.6.1.5 Extracting XMP metadata

Adobe's Extensible Metadata Platform (XMP) embeds metadata (such as descriptions, titles, keywords, author, and copyright information), into the file itself. In XMP-enabled applications, metadata is captured during the creation process and embedded in the file.

Transformation Services has the ability to extract XMP metadata from supported formats upon import. In addition, Transformation Services can write XMP metadata to those same file formats.

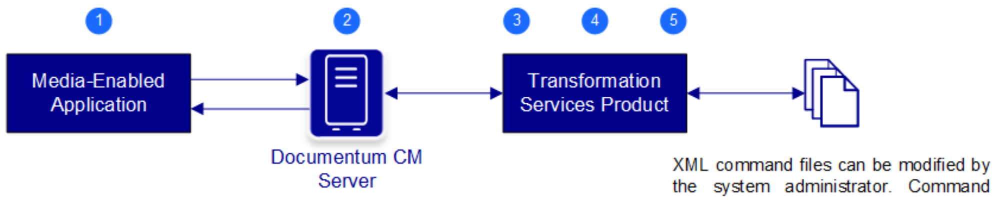
1.6.2 User transformations

User requests are processed similarly to register requests. The main difference is that processing is not automatic. A user must invoke a request for transformation.

The transformation process (depicted in Figure 1-11) is as follows:

1. The Documentum client applications (WDK clients such as Webtop, DA, Digital Asset Manager, and so on) get a list of available profiles and their required parameters by reading special system objects in the repository.
2. The user requests a transformation for an object by selecting a profile and entering parameters (if the profile requires parameters).
3. The application creates a transformation (TRANSCODE_CONTENT) request.
4. Documentum CM Server creates a queue object that requests the transformation. The request contains the profile ID, any parameter values required for a given operation (for example, height and width or angle of rotation), and the source and target formats.
5. The server polls the queue, finds objects for transformation, and matches the requested profile and source format to a particular plug-in.
6. The server locates the XML command line file for the profile, substitutes parameters as required, and invokes the plug-in to perform the transformation according to the selected profile and parameters.

   The transformation occurs in the background.
7. When the transformation is complete, the server sends the transformed media back to Documentum CM Server and updates the original object with the new rendition and its associated media properties, which are stored as attributes of the rendition.
8. The server removes the transformation request item from the queue.



XML command files can be modified by the system administrator. Command files are specific to each plug-in, since the capabilities of plug-ins vary.

1. Application request transformation after getting the list of profiles and their required parameters.
2. Documentum CM Server queues profile name and parameter list.
3. Transformation Services server reads from queue and matches the requested profile and source format to a particular plug-in.
4. Transformation Services server locates the XML command file for the profile.
5. Transformation Services server substitutes parameters accordingly, and involve the plug-in.

Figure 1-11: The transformation process

1.6.2.1 User transformation profiles

Transformation Services provides a set of predefined profiles for most common operations and allows administrators to extend or combine profiles and create new profiles to meet their specific requirements. The following are examples of some profiles that a Transformation Services administrator might create:

- `Create small JPEG`: Resizes an image to fit within 200 x 500 pixels and save it as a JPEG rendition.
- `Resize H x W JPEG`: Resizes an image to a specified height and width and save it as a JPEG rendition.
- `Add text layer GIF`: Adds a predefined text layer to an image and save it as a GIF rendition.

Chapter 2

Administering and Configuring Transformation Services through Documentum Administrator

Many administration and configuration tasks for Transformation Services products can be performed using the Transformation Services Administration component of Documentum Administrator. You must have access to Documentum Administrator and the repositories to which the Transformation Services product is configured.

The Transformation Services Administration component appears in Documentum Administrator if at least one instance is installed and configured on a repository to which it is connected. Perform this verification by executing the Documentum Query Language/OpenText Documentum Content Management (CM) Foundation Java API call to find the presence of at least one `cts_instance_info` object in the repository.

The following Transformation Services configuration and administration tasks can be performed through Documentum Administrator client, in the Transformation Services Administration Node:

- Change the user for repository login
- Change the polling interval: The polling interval is the amount of time in seconds that the instance will wait between polls.

  When document processing (transformation) is requested, Documentum CM Server creates queue items and appends them to the dm_mediaserver queue or dm_autorender_win31 queue, depending on the type of request. Transformation Services use a queue polling mechanism to look for items in the queue to process. When Transformation Services finds items in any of these queues, it proceeds with processing.
- Change the logging level: The logging level value controls how much information will be recorded in the Transformation Services log files, which the product uses. Log files can be used to troubleshoot the system.
- Change the system operator user: The system operator is the name of the user who receives messages from an instance of Transformation Services.

  If Transformation Services fails to process a particular item, it queues an event (dm_mediaserver_error) to the Inbox of the repository user who is defined as the system operator (sysOpUser). The event appears as an error message in the sysOpUser's Inbox.
- Change the system notification setting: The notification setting controls whether success notifications should be sent to each individual user requesting a transformation through Transformation Services.

- Change the maximum number of queue items: This value controls how many items the Transformation Services instance adds for processing each time it polls the queue.
- Change the queue item expiry: This value controls the amount of time an item will be sitting on a queue before being deleted from the queue.
- Change the Transformation Services user: This task can be done through Documentum Administrator. Alternatively, this task can be done using the setPassword utility.

  Note: The system administrator must also change the password on the server. “Changing the administrator password” on page 48 provides detailed information.

- View log files: Log files are created for each plug-in. The contents and detail level of each log file depend on the log file setting you have chosen for the Transformation Services instance.
- View details of an instance: Details include the Transformation Services that is configured for a repository, the version number of each product, the name of the host machine for each product, its current status (running or stopped), the time it was last started, the number of queued items for the instance, and the number of items processed by the instance.
- Control a Transformation Services instance: This includes starting, stopping, and refreshing an instance.

  Note: Performing any of the preceding actions apply to Transformation Services running on the host. This is because Transformation Services on a host uses the Transformation Services in Windows service. Starting and stopping Transformation Services Windows service can also be done manually, on the host machine. “Starting and stopping Transformation Services” on page 35 provides the instructions.

OpenText Documentum Content Management - Administrator User Guide (EDCAC250400-UGD) provides more information about the administration tasks.

2.1 Starting and stopping the Transformation Services Administration Agent

To use the Transformation Services Administration component in Documentum Administrator, it is necessary to have the Transformation Services Admin Agent running on the Transformation Services host machine. The Admin Agent can be controlled either from the Windows Start menu or by running a batch file.

To start or stop the Transformation Services Admin Agent through the Start menu:

1. Log in to the Transformation Services host.

2. Go to the Windows Services utility located at **Start > Control Panel > Administrative Tools > Services**.
3. Select **Documentum Content Transformation Services Admin Agent**.
4. To start the Admin Agent, click **Start**.

   To stop the Admin Agent, click **Stop**.

To start or stop the Transformation Services Admin Agent by running the batch file:

1. Log in to the Transformation Services host.
2. Navigate to `C:\Documentum\CTS\AdminAgent\bin\`.
3. To start the Admin Agent, run the `startup.bat` file.

   To stop the Admin Agent, run the `shutdown.bat` file.

Chapter 3

Configuration and Administration tasks and tips

This chapter contains configuration and administration tasks for Transformation Services products. Many configuration and administration tasks are performed through Documentum Administrator.

3.1 General configuration and administration tasks

The following sections contain procedures for general configuration and administration tasks that are performed with Transformation Services products.

3.1.1 Starting and stopping Transformation Services

When you restart the Transformation Services host, the Transformation Services server will start automatically. In addition, start and stop Transformation Services manually, using the following procedure. For example, this may be necessary after modifying a transformation profile.

To start or stop Transformation Services:

1. Log in to the Transformation Services host.
2. Go to the Windows services utility located at **Start > Control Panel > Administrative Tools > Services**.
3. Select **Documentum Content Transformation Services**.
4. To start the product, click **Start**.

   To stop the product, click **Stop**.

Note: A background service, called Documentum Content Transformation Monitor Services, restarts Transformation Services if it stops unexpectedly. Since the monitor is set to run automatically by default, you do not typically need to access it. It is located under Windows Services.

3.1.2 Adding or removing repositories using Transformation Services configurator

During installation, Transformation Services is configured to communicate with single or multiple repositories. Run the Configurator (accessible through the Windows Start menu) to add or remove a repository. *OpenText Documentum Content Management - Transformation Services Installation Guide (EDCCT250400-IGD)* provides information about the Configurator and how to run it.

Note: You must have SuperUser privileges for a repository to be able to add or remove support for Transformation Services.

3.1.3 Configuring separate Transformation Services instances to handle real-time and asynchronous requests

For improving the performance of Transformation Services while handling both asynchronous requests and real-time requests, you must have dedicated Transformation Services instances for handling each kind of request. If there are two Transformation Services instances, for example, `CTS_Realtime` and `CTS_Async`, and you need to configure them to handle real-time and asynchronous requests only, perform the following steps:

1. The `CTS_Realtime` instance should be tuned to not ping the queue and not pick up any of the queue items.

   a. Login to the `CTS_Realtime` machine and navigate to the *`<Documentum Content Transformation Services_HOME>`*`\config\` folder.

   b. Open the `CTSServerService.xml` file in a text editor.

   c. Comment out the full `QueueProcessorContext` sections, where `queueItemName="dm_mediaserver"` for all of the repositories that are configured with `CTS_Realtime` instance. There will be one *`<QueueProcessorContext>`* for one repository.

   d. Update the other *`<QueueProcessorContext>`* and rename the queueItemName `dm_autorender_win31`to `dm_autorender_win31_do_not_ping` for all of the repositories that are configured with `CTS_Realtime` instance.

   e. Save the file and restart the Transformation Services.

2. The applications sending real-time requests to Transformation Services need to be configured such that the requests do not go into `CTS_Async`.

   a. Login to the Application Server machine hosting your application.

   b. Navigate to `Documentum Content Transformation Services preferences.xml` and open it in a text editor.

   c. Use this configuration element to specify a list of semicolon-separated Transformation Services instances.

```
ServerProperty Key="CTS_SkipList" Description="list of Content Transformation
Services instances
(semicolon separated) to skip (scalability)" Value="CTS_Async;
<other-cts-instances>"/>
```

Note: Use the host name as it appears in the `cts_instance_info` table. To retrieve the host name, run the following DQL command:

```
select hostname from cts_instance_info
```

With these changes, `CTS_Realtime` does not pick up any of the queue items and be dedicated only for the real-time requests. The `CTS_Async` does not receive any of the real-time requests and processes only the queue items. This approach can be scaled up and applied to configure multiple Transformation Services instances for handling either real-time or asynchronous requests.

3.1.4 Integrating a Streaming Server with Transformation Services

This section describes how to integrate a Streaming Server with Transformation Services and Documentum CM Server. It also provides an example of how to test the Streaming Server integration.

Supported streaming file format families depend upon the Streaming Server(s) that you integrate with Transformation Services.

3.1.4.1 Prerequisites

Transformation Services does not include a Streaming Server. Streaming Server requirements must be followed as per the documentation for those products. Transformation Services is agnostic regarding streaming video servers and does not explicitly certify specific products or versions. Streaming Servers must be purchased from their vendors.

Before integrating one or more Streaming Servers, they should be installed on their own host, where possible. The Streaming Server should not be installed on the Documentum CM Server host. In many cases, the intensive operations of the Streaming Server may degrade the performance of the Documentum CM Server. There may be some exceptions to this, such as Windows Media Server. Refer to the documentation of the streaming server product for further details.

3.1.4.2 Setting up a Streaming Server

To integrate a Streaming Server with Documentum CM Server:

1. Create a second dm_filestore object to store Windows Media streaming formats, if applicable. “Setting up streaming storage areas” on page 38 provides detailed information.

2. Set the default_storage attribute in the streaming format objects to the streaming storage area, as described in “Setting default_storage for the format objects” on page 39.

3. Ensure that the Streaming Server has access to the storage area. Follow the instructions in “Ensuring Streaming Server access to storage areas” on page 41.

4. Create a mount point (or a VirtualDirectory for Adobe Flash Media Server) on the Streaming Server for the storage area, as described in “Creating mount points or VirtualDirectory” on page 41.

5. Set the base_url attribute for the storage area. “Setting the storage area's base_url attribute” on page 41 provides detailed information.

6. Restart the Streaming Server.

7. If using Windows Media Server, it should be initialized at least once prior to using with Transformation Services.

3.1.4.3 Setting up streaming storage areas

Each Streaming Server may support different file formats. To use two Streaming Servers, you require a second streaming storage area. The files handled by each server must be stored in separate areas. This means that in addition to the default store (streaming_store_01), you must create another streaming store (for example, streaming_store_02).

Note: It is important that you set the media_type to 2 in the dm_filestore object.

The *Storage Management* chapter in the *OpenText Documentum Content Management - Server Administration and Configuration Guide (EDCCS250400-AGD)* provides more information on creating storage areas.

To create a second streaming storage area:

1. *OpenText Documentum Content Management - Server Administration and Configuration Guide (EDCCS250400-AGD)* provides instructions to create a dm_location object. Using DQL statements, your setup should appear similar to the following:

```
create dm_location object
set object_name = ‘streaming_storage_02',
set path_type = ‘directory',
set file_system_path =
‘C:\DOCUMENTUM\data\<repository>\streaming_storage_02',
set mount_point_name = ‘'
go
```

2. *OpenText Documentum Content Management - Server Administration and Configuration Guide (EDCCS250400-AGD)* provides the instructions to create a dm_filestore object. Ensure that you set the media_type to `2`. Using DQL statements, your setup should appear similar to the following:

```
create dm_filestore object
set name = ‘streaming_store_02',
set root = ‘streaming_storage_02',
set is_public = true,
set require_ticket = false
set media_type = 2
go
```

3.1.4.4 Setting default_storage for the format objects

The default_storage attribute is used by the Documentum client to identify the storage file path for content. For streaming content, you must set the attribute to the object ID of the streaming content storage area.

Only Digital Asset Manager and Web Publisher use the default_storage setting in format objects. If a user saves a document with streaming content using any other Documentum client, the content is saved to the default storage area for the object type.

Note: File formats supported by each server may change or differ from those listed in the following sections. Consult your Streaming Server documentation for the current list of supported file formats.

3.1.4.4.1 Finding file format names

In the following DQL statements for setting the default_storage for format objects, file formats are entered, not file extensions. File format names can be found by running the following DQL statement:

```
select * from dm_format where dos_extension = 'extension'
```

For example, the following DQL statement finds the name of the `mov` extension:

```
select * from dm_format where dos_extension = 'mov'
```

The result of the mov query is one record, where quicktime is the proper format name for the mov extension:

r_object_id	name	description	dos_extension
2701e306800001bf	quicktime	QuickTime Movie	mov

3.1.4.4.2 Windows Media formats

Even if you are using Real Helix Universal server exclusively, you must also identify a separate storage area for Windows Media file formats. Use the following DQL statement to update the formats recognized by the Windows Media storage area:

```
UPDATE dm_format OBJECTS
SET default_storage=(SELECT r_object_id FROM dm_filestore
    where name ='<storage_area_name>')
WHERE name in (‘<format_name>', ‘<format_name>')
```

For example, if the Windows Media file formats are to be stored in streaming_store_01, the statement should appear as follows:

```
UPDATE dm_format OBJECTS
SET default_storage=(SELECT r_object_id FROM dm_filestore
    WHERE name ='streaming_store_01')
WHERE name in (‘wmv', ‘wma', ‘wmx', ‘wax', ‘asf')
```

3.1.4.4.3 Real Media formats

Use the following DQL statement to update the formats recognized by the Real Media storage area:

```
UPDATE dm_format OBJECTS
SET default_storage=(SELECT r_object_id FROM dm_filestore
    WHERE name ='<storage_area_name>')
WHERE name in (‘<format_name>', ‘<format_name>')
```

For example, if the Real Helix server formats are to be stored in streaming_store_02:

```
UPDATE dm_format OBJECTS
SET default_storage=(SELECT r_object_id FROM dm_filestore
    WHERE name ='streaming_store_02')
WHERE name in (‘ra', ‘ram', ‘rm', ‘rmm', ‘rnx', ‘rv', ‘quicktime')
```

3.1.4.4.4 Darwin formats

Use the following DQL statement to update the formats recognized by the Real Media storage area:

```
UPDATE dm_format OBJECTS
SET default_storage=(SELECT r_object_id FROM dm_filestore
    WHERE name ='<storage_area_name>')
WHERE name in (‘<format_name>', ‘<format_name>')
```

For example, if the Darwin formats are to be stored in streaming_store_03:

```
UPDATE dm_format OBJECTS
SET default_storage=(SELECT r_object_id FROM dm_filestore
    WHERE name ='streaming_store_03')
WHERE name in (‘quicktime')
```

3.1.4.4.5 Flash formats

Use the following DQL statement to update the formats recognized by the Adobe Flash Media Server storage area:

```
UPDATE dm_format OBJECTS
SET default_storage=(SELECT r_object_id FROM dm_filestore
    WHERE name ='<storage_area_name>')
WHERE name in (‘<format_name>', ‘<format_name>')
```

For example, if the Flash formats are to be stored in streaming_store_04:

```
UPDATE dm_format OBJECTS
SET default_storage=(SELECT r_object_id FROM dm_filestore
    WHERE name ='streaming_store_04')
WHERE name in (‘flv', ‘f4v')
```

3.1.4.5 Ensuring Streaming Server access to storage areas

On a Windows host, the Streaming Server must run under the System account or as the Documentum CM Server installation owner, to ensure access to the streaming content storage areas.

3.1.4.6 Creating mount points or VirtualDirectory

You must create a mount point (or a VirtualDirectory for Adobe Flash Media Server) on the Streaming Server that points to the root of the streaming storage area. The root path is the value found in the file_system_path attribute of the file path object associated with the storage area. To obtain that value, run the following DQL statement:

```
SELECT file_system_path FROM dm_location a, dm_filestore f WHERE
f.name='<storage_area_name>' AND a.object_name=f.root
```

Substitute the storage area's name for *`<storage_area_name>`*.

The returned value is a full root path specification for the storage area's file path.

Use your Streaming Server's product documentation for instructions on how to create a mount point or virtual directory.

3.1.4.7 Setting the storage area's base_url attribute

For the Documentum client to retrieve a file from a streaming storage area, Documentum CM Server must provide the URL for the content file. The Documentum CM Server then sends that URL to the Streaming Server. The base of the URL returned by Documentum CM Server is defined in the base_url attribute of the streaming storage area's storage object. This attribute must be set manually for streaming content storage areas.

Refer to the documentation accompanying the Streaming Server to determine the base URL.

Note: For all base_url paths, you must include the ending forward slash (/).

For Windows Media Server:

- on its own, it is typically similar to:

```
mms://<host_name>/mountpoint1/
```

For Real Helix Universal Server, the base_url is typically similar to:

- Real file formats

```
rtsp://<host_name>:<port>/mountpoint1/
```

- Windows file formats (if you are not using Windows Media Server as well)

```
http://<host_name>:<port>/mountpoint2/
```

For Darwin Streaming Server:

- the base_url is typically similar to:

```
rtsp://<host_name>/
```

For Adobe Flash Media Server:

- the base_url is typically similar to:

```
rtmp://<host_name>/vod/mp4:<VirtualDirectory>/
```

For Windows Media Server:

- Use the following DQL statement to set the base_url attribute:

```
UPDATE dm_filestore object
set base_url='<mms://<machine name>/mountpoint1/>'
where name = '<storage_area_name>'
```

For Real Helix Universal Server:

- Use the following DQL statement to set the base_url attribute:

```
UPDATE dm_filestore object
set base_url='rtsp://<machine name>:<port number>/<mount_point_name>/'
where name = '<storage_area_name>'
```

For Darwin:

- Use the following DQL statement to set the base_url attribute:

```
UPDATE dm_filestore object
set base_url='<rtsp://<machine name>'
where name = '<storage_area_name>'
```

For Adobe Flash Media Server:

- Use the following DQL statement to set the base_url attribute:

```
UPDATE dm_filestore object
set base_url='rtmp://<host_name>/vod/mp4:<VirtualDirectory>/'
where name = '<storage_area_name>'
```

3.1.4.8 Testing the Streaming Server

When you are finished integrating the Streaming Server with Documentum CM Server and Transformation Services, it is important that you test the Streaming Server's functionality.

You can test the configuration of your Streaming Server through IAPI calls.

3.1.4.8.1 Real Helix on Windows

The example in this section shows a test session for Real Helix Universal Server on Windows.

To test the Real Helix server on Windows:

1. Type the text in bold, when prompted:

```
/iapi32

Please enter a repository name (docubase) : dmtestdb
Please enter a user (Documentum): <user>
Please enter password for Documentum :<password>

         Documentum iapi - Interactive API interface
        (c) Copyright Documentum, Inc., 1992-2001
        All rights reserved.
        Client Library Release 5.1.0.64 Win32

Connecting to Server using repository dmtestdb
[DM_SESSION_I_SESSION_START]info:  "Session
011e9a8b80040107 started
for user Documentum."

Connected to Documentum Server running Release
5.1.0.64 Win32.Oracle
Session id is s0
API> create,c,dm_document
...
091e9a8b80037d00
API> set,c,l,object_name
SET> Streaming test
...
OK
API> set,c,l,a_storage_type
SET> streaming_store_01
...
OK
API> setfile,c,l,C:\Program Files\Real\RealServer\
Content\real8video.rm,rm
...
OK
API> save,c,l
...
OK
API> apply,c,l,GET_FILE_URL,FORMAT,S,rm
...
q0
API> next,c,q0
...
OK
API> dump,c,q0
...
USER ATTRIBUTES

result : T
```

```
base_url: rtsp://eng178:554/mountpoint1/
store  : streaming_store_01
path : 001e9a8b\80\00\00\42.rm
ticket  :
SYSTEM ATTRIBUTES

APPLICATION ATTRIBUTES

INTERNAL ATTRIBUTES

API> close,c,q0
...
OK
API>
```

2. Concatenate the value for base_url and path (for example, `rtsp://eng178:554/mountpoint1/001e9a8b\80\00\00\42.rm`) and paste the string into the Real Player.

   The video represented by the URL begins to stream.

3.1.5 Defining file formats and DOS extensions

Define file formats and DOS file extension pairs in the repository for all source document types that you want Transformation Services to recognize that are not standard, out of the box pairs. In addition, if the DOS file extension differs from the Documentum format name, update the `FormatMapperService.xml` config file on the Transformation Services host.

OpenText Documentum Content Management - Administrator User Guide (EDCAC250400-UGD) provides information and procedures to add file formats to your system.

The following procedure is required only if the DOS file format differs from the Documentum format name in Documentum CM Server. “Documentum format mapping” on page 44 lists some examples of the format mapping used in Documentum systems. Use these format names in DQL.

Table 3-1: Documentum format mapping

Format	Documentum name
Bitmap image (BMP)	bmp
Encapsulated PostScript (EPS)	eps
GIF image	gif
JPEG image	jpg
TIFF image	tif

To update the FormatMapperService config file:

1. Navigate to *`<Content Transformation Services_HOME>`*`\config\`.

2. Open the `FormatMapperService.xml` file in a text editor.

3. Add the format to the `CTSHandlerList` section, using the following syntax:

   ```
   <FormatMapper CTSFormat="<dos extension>" DocumentumFormat="<format name in
   Documentum Server>"/>
   ```

   For example, the entry for JPEG files is:

   ```
   <FormatMapper CTSFormat="jpeg" DocumentumFormat="jpg"/>
   ```

4. Save and close the `FormatMapperService.xml` file.

3.1.6 Configuring Inbox notifications

When transformation requests fail, the Transformation Services server sends notifications to the Inbox of the repository user. By default, success notifications are not sent. However, these default behaviors can be configured using the procedures in this section.

To modify the configuration of success notifications:

1. Navigate to *`<Documentum Content Transformation Services_HOME>`*`\config\`.

2. Open the `CTSServerService.xml` file in a text editor.

3. To turn on notifications for successful transformations, set the *`notifySuccessMessage`* parameter to `YES` as follows:

   ```
   <CTSServer AttributeName="notifySuccessMessage" AttributeValue="YES"/>
   ```

   To turn off notifications for successful transformations, set the *`notifySuccessMessage`* parameter to `NO` as follows:

   ```
   <CTSServer AttributeName="notifySuccessMessage" AttributeValue="NO"/>
   ```

   Note: There are two *`<QueueProcessorContext>`* tags for each repository that is configured. Modify the *`notifySuccessMessage`* parameter within each *`<QueueProcessorContext>`* tag.

4. Save and close the `CTSServerService.xml` file.

To modify the configuration of failure notifications:

1. Navigate to *`<Documentum Content Transformation Services_HOME>`*`\config\`.

2. Open the `CTSServerService.xml` file in a text editor.

3. To turn on notifications for failure transformations, set the *`notifyFailureMessage`* parameter to `YES` as follows:

   ```
   <CTSServer AttributeName="notifyFailureMessage" AttributeValue="YES"/>
   ```

   To turn off notifications for failure transformations, set the *`notifyFailureMessage`* parameter to `NO` as follows:

   ```
   <CTSServer AttributeName="notifyFailureMessage" AttributeValue="NO"/>
   ```

Note: There are two *`<QueueProcessorContext>`* tags for each repository that is configured. Modify the *`notifyFailureMessage`* parameter within each *`<QueueProcessorContext>`* tag.

4. Save and close the `CTSServerService.xml` file.

3.1.6.1 Inbox notification behavior

The following table details the behavior for administrative and regular users based on the type of queue item and notification options selected:

Table 3-2: Inbox notification behavior for admin and regular users

Legacy calls (dm_autorender_win31 queue items)		
Success Notification	Admin user	Notified based on the entry in `CTSServerService.xml` for *`notifySuccessMessageAdmin`*.
	Regular user	Notified based on the entry in `CTSServerService.xml` for *`notifySuccessMessage`*.
Failure Notification	Admin user	Notified based on the entry in `CTSServerService.xml` for *`notifyFailureMessageAdmin`*.
	Regular user	Notified based on the entry in `CTSServerService.xml` for *`notifyFailureMessage`*.
Regular Queues (dm_mediaserver queue items)		
Success Notification	Admin user	Notified based on the entry in `CTSServerService.xml` for *`notifySuccessMessageAdmin`*.
	Regular user	Notified based on the entry in `CTSServerService.xml` for *`notifySuccessMessage`*.
Failure Notification	Admin user	Notified based on the entry in `CTSServerService.xml` for *`notifyFailureMessageAdmin`*.
	Regular user	Notified based on the entry in `CTSServerService.xml` for *`notifyFailureMessage`*.

If the client requests for notification through the queue item, using the *`-notifier_user`* parameter in message attribute, then this notification setting takes precedence over settings from `CTSServerService.xml` for that particular transformation request.

3.1.7 Changing the Transformation Services user

In certain situations, it may be desirable to change the Transformation Services user account after Transformation Services has been installed and configured. This may be necessary for security reasons.

It is not necessary to uninstall and reinstall Transformation Services to make this change. Ensure that the new user has the Admin privilege or same access as the existing user for any rendering software (for example, Ghostscript is used by Transformation Services).

3.1.8 Configuring multiple domains for Transformation Services products

The multiple domains feature processes user requests to multiple domains. This feature is enabled only if the repository is already in domain-required mode.

To configure users for multiple domains using Documentum Administrator:

1. Connect to the repository that is configured for the Transformation Services instance.

2. Click the **Administration** node.

3. Click the **Documentum Content Transformation Services** node.

4. Click the **CTS Instances** node.

5. Select the Transformation Services instance from the list of available instances.

6. Click **Tools > Documentum Content Transformation Services > Configure > Users**.

7. The **CTS Administrative Users** page displays a list with pre-configured administrative users and their domains for the selected Transformation Services instance. Click **Add**.

   The **Content Transformations Services User Details** page appears.

8. Select the required super user from the list of options.

   Note: The required dm_user should already be added to the repository.

9. Type the correct password.

10. Type the correct domain.

11. Click **OK** to return to the **CTS Administrative Users** page.

12. Click **OK** to save the changes.

To change the password of an existing domain user configuration using Documentum Administrator:

1. From the **CTS Administrative Users** page, select the required domain user. Click **Edit**.

   The **Content Transformations Services User Details** page appears with the **Password** fields enabled.

2. Type the new password.

3. Click **OK** to return to the **CTS Administrative Users** page.

4. Click **OK** to save the changes.

To remove a domain user configuration using Documentum Administrator:

1. From the **CTS Administrative Users** page, select the domain user configuration you want to remove. Click **Delete**.

2. Click **OK** to save the changes.

3.1.9 Changing the administrator password

Run the `setPassword` utility to change the Content Transformations Services administrator's password. The password must be changed on Documentum CM Server also.

1. Stop Transformation Services.

2. Change the password on the Documentum CM Server.

3. Open a command prompt window.

4. Type the following command:

   ```
   > cd <Documentum Content Transformation Services_HOME>\docbases\<repository name>
   \CTSServerScript\bin
   > setPassword <new_password>
   ```

5. Restart the Transformation Services.

Note: It is also possible to change the password in Documentum Administrator instead of running the `setPassword` utility.

3.1.10 Using the Transformation Services Reporting tool

The Transformation Services Reporting tool provides detailed information about transformation type, volume, and error events. Data such as Transformation Services performance, plug-in usage, and requests by users assists in monitoring and analyzing transformation usage and can assist with load balancing strategies. Reporting can be logged for a configurable time frame and then rolled over to be archived. The resulting information is retained in the repository and is viewable as a CSV formatted file (a report format using comma-separated values) through Digital Asset Manager. Users can view their own Transformation Services reporting data. Administrators have access to all data.

Usage tracking is configured in Documentum Administrator. The Transformation Services Reporting tool reports on the number of requests in a given time frame according to the following options:

- Number of successful transformations
- Number of requests that could not be handled
- Cumulative error report
- Number of errors per format and per specific transformation request
- Cumulative total file sizes of input and output per given time frame
- Total requests per specific user

3.1.10.1 Enabling and disabling the Transformation Services Reporting tool

The Transformation Services Reporting tool is turned off by default. Usage tracking can be switched on or off.

1. Login to Documentum Administrator.
2. In the tree pane, select **Transformation Services > CTS Reporting Configuration**.
3. Select the following:
    - Reporting Configuration (ON/OFF) check box, to select whether the data is collected.
    - Purging Configuration (ON/OFF) option, to delete any data collected.
    - Archiving Configuration (ON/OFF) option, to select whether data should be archived.
4. If you have enabled **Reporting Configuration and Archiving Configuration**, provide the information for the following fields:
    - **Number of days**: Archiving Interval to set how often the report should be archived – everyday, every two days, and so on.

  - **Size of transformation request table**: Archiving DataSize, to set the number of transformations that should be recorded at a time before the data is archived and for the compilation to start again.
  - **Archiving monitor interval in seconds**: Archiving Monitor Interval, to set the length of time in seconds the report should be updated.

5. Click **OK**.

3.1.10.2 Viewing the Transformation Services Reporting tool report

When the `Documentum Content Transformation Services Reporting` tool is enabled, the following information is recorded for each transformation request:

- The object ID of the source document
- The object ID of the queue item
- The user name who sends the requests
- The format of the source document SOURCE_FORMAT = “source_format”
- The format of target TARGET_FORMAT = “target_format”
- The transformation type, it could be either user created or auto/import
- The profile name used for the transformation request
- The target object type, it could be rendition or related object
- The parameters send in the request
- The queue item added time
- The queue item signed off time
- The transformation completed time
- The source file size
- The target file size
- The status, it could be SUCCESS, FAILED, or UNHANDLED
- The Transformation Services instance doing the transformation
- Any message resulting from a transformation execution

1. In `Digital Asset Manager`, go to **Tools > Transformation report > View**.

   The **CTS Transformation Report** page appears.

2. Type information for the following fields:

   - Report name
   - Start date
   - End date

3. Click **OK**. The **Select Folder** screen appears.

4. Select the destination folder to save the retrieved report. Click **OK**. A CSV report opens with the requested information.

3.1.11 Managing the transformation queue

The Queue Management feature provides a mechanism for monitoring and administering transformation request queues. This feature allows users to view or delete their own pending transformation requests. Administrators can manage the queues for all users and change the priority of transformation requests when required.

The Queue Management feature is accessed through the Transformation node in Digital Asset Manager.

Administrators can manage the queue by deleting items or changing the priority of items. Users can view their own transformation requests and delete their own transformation requests, but cannot change a queue item priority.

3.1.11.1 Viewing transformation requests in the queue

Users can monitor the status of their transformation requests by viewing the queue. If a user is dissatisfied with the priority level of their transformation request(s), they can ask their administrator to change an item's priority on their behalf.

1. Login to Digital Asset Manager.
2. In the tree view, select the **Transformations** node. Click the **Transformations** link. A list of current queue items appears on the right pane.

   Transformation requests are displayed in the order of priority. The priority level of an item is indicated as follows:

   - !!! denotes a high priority item.
   - !! denotes a medium priority item.
   - ! denotes a low priority item.
3. (For administrators only) Choose the items to view by selecting an option from the **Show list** box:

   - **Show All** displays all users transformation requests.
   - **Show Mine** displays only your transformation requests.
4. To sort transformation requests in ascending or descending order, click the header field on each of the columns.

3.1.11.2 Changing a queue item's priority

Administrators can change the priority of an item in the queue.

1. Login to `Digital Asset Manager`.
2. In the tree view, select the **Transformations** node. The transformation queue appears in the main pane.
3. Find the target item in the queue.
4. Select the item, right-click, and select **View** from the context menu.

   The transformation details of this item are displayed.
5. Change the numeric value in the **Priority** box. The highest priority is 10. The higher the numeric value, the higher the priority level. When the priority level is changed, the queue position of the object changes automatically.
6. Click **OK**. The page returns to the list of transformation queue. Verify if the queue position has changed. The color code of the priority item should change to a higher priority. "Viewing transformation requests in the queue" on page 51 provides the information on queue item priority level color codes.

3.1.11.3 Viewing details of a queue item

When a pending item is in the queue, the following transformation details are available depending on which fields are configured:

- **Description** gives the name of the transformation being performed, such as Rotate.
- **Parameters** lists any user-defined parameters for this transformation, if applicable. For a transformation such as Rotate, for example, the angle of rotation is a parameter.
- **From** gives the name of the user who requested the transformation.
- **Received** is the date and time the transformation was requested.
- **Source** is the source (or input) file name being transformed.
- **Priority** is given to a queue item with the highest numeric value.
- **Target** is the target (or output) file name for the transformation. If the request is to create a rendition, the Target field shows `None`.
- **Queue position** gives the specific order number of this item in the queue.

1. Login to Digital Asset Manager.
2. In the tree view in the left pane, select the **Transformations** node. The transformation queue appears in the main pane.
3. Find the target item in the queue.

4. Select the item, right-click, choose **View** from the context menu. The transformation details of this item are displayed.

3.1.11.4 Deleting a pending queue item

Administrators can delete any items in the queue. Users can only delete their own queue items. The deletion is not immediate for users. Items are marked, then polled and then deleted.

Note: The queue item cannot be deleted if Transformation Services has already picked up the queue for processing.

1. Go to the transformation queue by selecting the **Transformations** node. The transformation queue appears in the main pane.
2. Find the item(s) to delete from the queue.

   To find items quickly, configure the display as follows:

   - Sort alphabetically by document name by clicking the header on the **Document** column.
   - Sort the requests alphabetically by user name by clicking the header on the **Sent by** column.
   - Toggle between all users' transformation requests or your own transformations using the Show list box.
3. Select the item(s), and right-click with your mouse.

   Note: Select a range of items using the Shift key, or select multiple items using the Ctrl key.
4. Select **Delete** from the context menu. A delete confirmation page appears for each file you selected.
5. Select **OK** to confirm a deletion, or click **Cancel** to abort a deletion request.

If the queue item is pending, it is deleted and removed from the transformation queue. If the queue item is being processed, an error will show saying `Transformation request is being processed, can't delete`.

3.1.11.5 Configuring additional queue management fields

Customize the queue management fields by adding additional columns.

1. Login to Digital Asset Manager.
2. Go to the **Transformations** node. The right pane shows the default column fields.
3. Click **Column Preferences**.

   The screen displays the Preferences: Display Settings on the left and the default column setting on the right.
4. Select an additional item from the left list **Select attributes to display**. Click the button to add to the right list **Selected attributes to display as column**. The right list shows the new attribute added. Click **OK**.

   The list view appears.
5. Verify if the list view has the additional columns.

3.1.12 Configuring the queue pre-processor mechanism

This procedure mainly applies if you have different Transformation Services instances polling the same repository but installed on separate hosts.

The queue pre-processor mechanism analyzes queue items and marks those that can be executed by the Transformation Services instance, preventing items from being prematurely expired by a Transformation Services instance that cannot handle them. In addition, this helps the regular queue processor thread not to perform the capability checking again during the execution. The pre-processor tasks run with their own sessions.

This functionality is controlled by two elements in the `CTSServerService config` file:

- allowQueuePreProcessing

  This element is used to pre-process queue items when Transformation Services starts up. If the value is set to `YES`, the pre-processor thread is created. By default, the value is set to `NO`, and there is no pre-processing mechanism.
- markerInterval

  This element sets the wait period between consecutive queue pre-processing jobs.

The expiration period and marking interval should be tuned based on each system configuration. On a system with more than one Transformation Services instance polling the same queue, the marker mechanism is not required and thus the default setting is appropriate. However, if there are different Transformation Services instances running against a repository, one Transformation Services instance might expire items (because it cannot handle them) when the instance that could handle them is busy. In this case, the pre-processing mechanism should be turned on for those instances that could process the items.

3.1.13 Controlling how quickly programs time out

When Transformation Services is processing a request, it utilizes third party plug-ins as part of the process. There may be occasions when one of these plug-ins fail, or simply runs longer than you want it to run. To control the resources consumed by a plug-in, Transformation Services includes an application timeout parameter. The value assigned to the parameter determines how long the product waits for a plug-in to complete its process before moving on to the next request. The timer starts when Transformation Services calls the plug-in.

For example, if Transformation Services calls a third party plug-in to print a PDF version of a document and that process does not complete within 30 seconds, Transformation Services abandons the request and checks the queue for the next request. When a rendition operation fails, Transformation Services sends a message reporting the failure to the Transformation Services log file and also to the requester’s Inbox.

The value for the application timeout parameter is controlled by the following related tags in the plug-in configuration file:

```
<APPLICATION_WAIT_INTERVAL>1200</APPLICATION_WAIT_INTERVAL>
<APPLICATION_MONITOR>true</APPLICATION_MONITOR>
```

The default value for *APPLICATION_WAIT_INTERVAL* is `1200`.

For example, if you want to configure *APPLICATION_WAIT_INTERVAL* to a higher value for any PowerPoint transformation, modify the tags of the `powerpoint1.xml` file (available in the *<Documentum Content Transformation Services_HOME>*`\CTS\config\powerpoint1\` folder) as follows:

```
<APPLICATION_WAIT_INTERVAL>2400</APPLICATION_WAIT_INTERVAL>
APPLICATION_MONITOR>true</APPLICATION_MONITOR>
```

The wait interval represents the amount of time (in seconds) between retries. The max_retries value multiplied by the wait_interval value gives the timeout parameter value.

```
<MAX_RETRIES>60</MAX_RETRIES>
<WAIT_INTERVAL>5</WAIT_INTERVAL>
```

For example, if you want to configure this value to 30 seconds, modify the values as follows:

```
<MAX_RETRIES>6</MAX_RETRIES>
<WAIT_INTERVAL>5</WAIT_INTERVAL>
```

Note: This task can also be performed through the Transformation Services Administration component in Documentum Administrator. “Administering and Configuring Transformation Services through Documentum Administrator“ on page 31 provides the instructions.

3.1.14 Caching Transformation Services capabilities on startup

An option is available to cache a Transformation Services instance's execution capabilities. Transformation Services can cache execution capability information of all the atomic profiles in the repositories in which a Transformation Services instance is configured against. This caching can be done either during the Transformation Services startup which can be configured in `CTSProfileService.xml`, or during the first load balancer call for getting the capabilities. Refer to the following tags added to the service configuration file. The default value is set to `true`, which means Transformation Services caches this capability information during startup:

```
<ProfileConfig ProfileConfigName="cacheCapabilityOnStartup" ProfileConfigValue="true"/>
```

The logging information related to this is logged to a separate log file configured in `log4j2.properties` CAPABILITYAppender. "Setting logging appenders" on page 56 provide more information.

While performing profiles across instances, if Transformation Services identifies a remote instance, capable of executing any of the inner atomic profiles of a nested profile, Transformation Services caches this information up to the profileRefreshInterval attribute value configured in the following tag. This value represents numbers of minutes.

```
<ProfileConfig ProfileConfigName="profileRefreshInterval" ProfileConfigValue="5"/>
```

3.1.15 Setting logging appenders

Separate logging appenders are added to `log4j2.properties` for logging the polling and capability caching information. Refer to the following entries in `log4j2.properties`:

```
appender.F37.type=RollingFile
appender.F37.name=POLLINGAppender
appender.F37.fileName=<Documentum Content Transformation Services_HOME>\\logs\
\Polling_log.txt
appender.F37.filePattern=<Documentum Content Transformation Services_HOME>\\logs\
\Polling_log.%d{yyyy-MM-dd}-%i
appender.F37.layout.type=PatternLayout
appender.F37.layout.pattern=%d{ABSOLUTE} %5p [%t] %c - %m%n
appender.F37.policies.type=Policies
appender.F37.policies.time.type=TimeBasedTriggeringPolicy
appender.F37.policies.time.interval=8
appender.F37.policies.time.modulate=true
appender.F37.policies.size.type=SizeBasedTriggeringPolicy
appender.F37.policies.size.size=10MB
appender.F37.strategy.type=DefaultRolloverStrategy
appender.F37.strategy.max=5
logger.R37.name=POLLINGAppender
logger.R37.level=DEBUG
logger.R37.additivity=false
logger.R37.appenderRef.F37.ref=POLLINGAppender
#-----------------------------------------------------------------

appender.F39.type=RollingFile
appender.F39.name=CAPABILITYAppender
appender.F39.fileName=<Documentum Content Transformation Services_HOME>\\logs\
\Capability_log.txt
appender.F39.filePattern=<Documentum Content Transformation Services_HOME>\\logs\
\Capability_log.%d{yyyy-MM-dd}-%i
appender.F39.layout.type=PatternLayout
appender.F39.layout.pattern=%d{ABSOLUTE} %5p [%t] %c - %m%n
appender.F39.policies.type=Policies
appender.F39.policies.time.type=TimeBasedTriggeringPolicy
appender.F39.policies.time.interval=8
appender.F39.policies.time.modulate=true
appender.F39.policies.size.type=SizeBasedTriggeringPolicy
appender.F39.policies.size.size=10MB
appender.F39.strategy.type=DefaultRolloverStrategy
appender.F39.strategy.max=5
logger.R39.name=CAPABILITYAppender
logger.R39.level=DEBUG
logger.R39.additivity=false
logger.R39.appenderRef.F39.ref=CAPABILITYAppender
```

The log files `Polling_log.txt` and `Capability_log.txt` corresponding to these appenders contain the logs related to polling and capability caching information respectively. This information is not logged to the main `CTS_log.txt` file. The log level can be set to `DEBUG` for retrieving more information.

3.1.16 Enabling performance and throughput logging

The ability to create a separate performance log file is available. The following entry in `log4j2.properties` enables this functionality:

```
ogger.performancelogging.name=performancelogging
logger.performancelogging.enabled=false
logger.performancelogging.file=<Documentum Content Transformation Services_HOME>\\logs\
\Performance_log.txt
```

The default value is set to `false`. Setting this value to `true` creates a `Performance_log.txt` in the *`<Documentum Content Transformation Services_HOME>`*`\logs\` folder.

Edit the value of the following entry to define the interval of rolling up performance log into storage repository:

```
logger.performancelogging.rollupInterval=24
```

The value format is an integer number +H (case insensitive). For example, for an interval of 12 hours, the value is 12H. If omitted, the value is a default value of 24H. When the interval value has been reached, the performance log is parsed and a performance and throughput report XML is generated. Then, a ZIP file, which contains the report XML and the related stylesheets, is uploaded to the storage repository as defined in `SessionService.xml`.

3.1.17 Processing requests for local content

Transformation Services includes a configuration element called processLocalContentOnly, which is useful in distributed environments to instruct Transformation Services to process requests for content residing in local (near) file stores and not pick up requests for content in distant files stores.

The configuration is available in *`<Documentum Content Transformation Services_HOME>`*`\config\CTSServerService.xml`. In each *`<QueueProcessorContext>`* node, the following line exists:

```
<CTSServer AttributeName="processLocalContentOnly" AttributeValue=""/>
```

Valid values are `YES` or `NO`. The default is `NO`.

3.1.18 Configuring Transformation Services for BOCS or ACS

If required, use the Branch Office Caching Services (BOCS) documentation to install BOCS and configure it for a global registry. The following section outlines the procedures required to configure Transformation Services for BOCS.

The following are limitations of BOCS with Transformation Services:

- Inbound operations (saving renditions and checking-in) are not supported.
- Saving renditions is not supported due to the Foundation Java API limitation (IDfImportOperation does not support adding renditions).
- Checking-in (version up) is not supported.
- XML documents are exported through Foundation Java API. Foundation Java API does not allow you to transfer XML contents through OpenText™ Documentum™ Content Management Branch Office Caching Services/ OpenText™ Documentum™ Content Management Accelerated Content Services even though the OpenText Documentum Content Management (CM) Branch Office Caching Services/OpenText Documentum Content Management (CM) Accelerated Content Services options are specified.

1. For all Transformation Services that use Accelerated Content Services or Branch Office Caching Services, update the `CTSServerService.xml` file located at *`<Documentum Content Transformation Services_HOME>`*`\config\`.

   The file may be updated at the master configuration level as follows:

   ```
   <CTSServerConfig>
   ...
   <BocsConfig networkLocationId="" allowBocsTransfer="true" preferAcsTransfer="true"
    allowSurrogateTransfer="true" processOnlyParked="false"/>
   ...
   </CTSServerConfig>
   ```

   Overwrite the master settings, using repository specific preferences, through the queue processor configuration:

   ```
   <QueueProcessorContext DocbaseName="my_repository">
   ...
   ```

```
<CTSServer AttributeName="networkLocationId" AttributeValue=""/>
<CTSServer AttributeName="allowBocsTransfer" AttributeValue=""/>
<CTSServer AttributeName="preferAcsTransfer" AttributeValue=""/>
<CTSServer AttributeName="allowSurrogateTransfer" AttributeValue=""/>
<CTSServer AttributeName="processOnlyParked" AttributeValue=""/>
...
</QueueProcessorContext>
```

These options are based on IDfAcsTransferPreferences:

- networkLocationId: Preferred network file path identifier, if multiple network file paths can be applied to the machine. The network file path identifiers need to be configured in the global registry repository. If this is not set, the first available network file path identifier is used.
- allowBocsTransfer: Specifies whether Branch Office Caching Services content transfer is allowed. The default value is `true`.
- preferAcsTransfer: Sets the accelerated content transfer preference. The default value is `true`.
- allowSurrogateTransfer: Specifies whether surrogate transfer is allowed. The default value is `true`.

Note: Before signing off the queue item, Transformation Services performs a canExecuteProfile check to verify if the content for a transformation is available. If any of the content that participates in the transformation is not available for a particular Transformation Services instance, it indicates that the content for a transformation is not available.

2. Update the export operations.

   If allowBocsTransfer or preferAcsTransfer is set and the ACS transfer is configured for the repository, all the export operations specifies the ACS preference options and try to get the transfer URLs for contents.

   If the transfer URLs are returned, the contents are downloaded through the URL protocols.

   Note: If allowBocsTransfer is set to `true`, then preferAcsTransfer must also be set to `true`, if the content has to be retrieved through Branch Office Caching Services. This is a Foundation Java API requirement.

3.1.19 Configuring queue management for Branch Office Caching Services installations

Use the processOnlyParked feature to ensure that Transformation Services polls queue items only from the content available at a specific Branch Office Caching Services file path.

1. Following are the requirements when the processOnlyParked is set to `true`:

    - allowBocstransfer = true
    - preferAcsTransfer = true
    - allowSurrogateTransfer = true
    - parkingServerName=“BOCS_ABC” where object_name of the Branch Office Caching Services config object associated with the Branch Office Caching Services server
    - networkLocationId=“BOCS_NETLOC” where network file path is served by the BOCS_ABC. Ensure that this networkLocation is server by BOCS_ABC

    Note: These settings can be specified at a global or master level (applicable to all repositories) or at a local level (specific repository).

2. To make the processOnlyParked settings global, navigate to the `//CTSCustomConfig/CTSServerConfig/BocsConfig/` xpath and perform the following modifications:

    ```
    <BocsConfig allowBocsTransfer="true"
    allowSurrogateTransfer="true"
    preferAcsTransfer="true"
    networkLocationId="BOCS_NETLOC"  <!--Network location served by the BOCS_ABC.
    Make sure that this networkLocation is server by BOCS_ABC-->
    processOnlyParked="true"
    parkingServerName="BOCS_ABC"/>  <!--object_name of the BOCS config object
    associated with the BOCS server-->
    ```

3. To make the processOnlyParked settings specific to a repository, add or modify the following tags in one of the QueueProcessorContext node in the repository xpath: `//QueueProcessorContext@DocbaseName=“My_Docbasename”`:

    ```
    <CTSServer AttributeName="allowBocsTransfer" AttributeValue="true"/>
    <CTSServer AttributeName="allowSurrogateTransfer" AttributeValue="true"/>
    <CTSServer AttributeName="networkLocationId" AttributeValue="BOCS_netLOC"/>
    <CTSServer AttributeName="preferAcsTransfer" AttributeValue="true"/>
    <CTSServer AttributeName="processOnlyParked" AttributeValue="true"/>
    <CTSServer AttributeName="parkingServerName" AttributeValue="BOCS_ABC"/>
    ```

    Note: Ensure that you replicate the same values in all QueueProcessorContext nodes for a repository or define these settings in one of the QueueProcessorContext nodes for a repository. If different set of values are specified in different QueueProcessorContext nodes for the same repository, then you get a different behavior than expected.

4. If local values are not defined, then global values are used as a fallback option.

3.1.20 Load balancing servers

The standard installation for Transformation Services consists of one repository with one server. Additional servers can be added to an active repository to enhance its performance. Load balancing can optimize performance by spreading transformation requests between multiple servers.

Load balancing can be achieved by adjusting settings in the `CTSserverservice.xml` file, in the *`<Documentum Content Transformation Services_HOME>`*`\config\` folder. By default, Transformation Services is equipped with a standard load balancing strategy.

The ideal load balancing strategy depends upon your environment and the specific servers in use. Contact OpenText Global Technical Services for load balancing guidance.

3.1.20.1 Configuring Transformation Services load balancer

The `preferences.xml` file (default file path is `C:\Documentum\CTS\config\`) allows configuration for various aspects of load balancer functionality. The following details what each line of the configuration file does.

This line is used to configure how Transformation Services should handle the request:

```
<LoadBalancer type="remote" URL=" http://host:port/services/transformation/LoadBalancer/"
 sendMode="remote"/>
```

- *`type=“local”`* - load balancer runs within the same JVM as the API client
- *`type=“remote” && sendMonde=“local”`* - call is made to the remote load balancer (specified in `URL=“http://...”`) to retrieve the Transformation Services instance URL and SBO sends the request
- *`type=“remote” && sendMonde=“remote”`* - call is made to the remote load balancer to select the instance and send the request

These two lines control failover retries for the case where a request is sent to Transformation Services but comes back with an error:

```
<ServerProperty Description="Allow a number of retries if a request sent to CTS
 fails" Key="FailoverRetries" Value="1"/>
<ServerProperty Description="Wait between failover retries (seconds)"
 Key="FailoverWait" Value="1”/>
```

Specify how frequent the load balancer should refresh its Transformation Services occupancy level cache:

```
<ServerProperty Key="CTSOccupancyPollingInterval" Description="Specify occupancy
 polling interval in seconds" Value="7"/>
```

Specify the number of repository connection retries (in case *`<Repositories>`* node is not configured):

```
<ServerProperty Key="ConnectionRetries" Description="Specify connection retries
 (in case Repositories section is not configured )" Value="10”/>
```

Control failover for the scenario where no Transformation Services instances are found that are available to execute a request (that is, an instance is busy or starting up, meaning there is chance in the near future it could become available):

```
<ServerProperty Key="AvailabilityRetries" Description="Number of retries when
 CTS instances are not available" Value="0"/>
<ServerProperty Key="AvailabilityWait" Description="Number of seconds to wait
 for rechecking availability" Value="0"/>
```

Use this configuration element to specify a list of semicolon separated Transformation Services instances (host name as it appears in `cts_instance_info`). It can be used to have one or more instances process only asynchronous requests or scale up the load balancer by assigning a specific set of Transformation Services instances to one load balancer and a different set of Transformation Services instances to another load balancer (Transformation Services instances from the same repository):

```
<ServerProperty Key="CTS_SkipList" Description="list of cts instances (semicolon
 separated) to skip (scalability)"  Value="CTS1;CTS3"/>
```

The *`InstanceSelector`* property specifies the selection algorithm used by the load balancer. The new value points to the occupancy based implementation:

```
<ServerProperty Key="InstanceSelector" Description="Specify an implementation class
 for instance selection "
 Value="com.emc.documentum.cts.lb.workers.OccupancyBasedSelector"/>
```

Note: The *`InstanceSelector`* value is used internally and should not be changed.

3.1.21 Configuring Transformation Services Server for SSL Mode

Transformation Services server and Transformation Services WebServices can be configured to run in SSL mode (that is, HTTPS instead of HTTP).

This is enabled at the application server level, where each application server has their own procedure to make it handle HTTPS. However, Transformation Services must be updated to accept the new SSL port.

For any real time requests, the Transformation Services WebServices server communicates to the Transformation Services server through HTTP. Transformation Services server package contains the Jetty application server. To configure this communication in SSL (HTTPS) mode, the following procedures are provided.

To configure Transformation Services WebServer to run in SSL mode, consult the publicly available reference material.

1. Verify the status of Jetty. The status of Jetty can be verified as follows:

   a. Start the Transformation Services server.

   b. Run the following query against the repository to get the Jetty running URL:

```
Select websrv_url, hostname from cts_instance_info
```

c. Access the attribute value of websrv_url from a browser to verify if it is running.

2. Configure Jetty to run in SSL mode.

   a. Update the `jetty.properties` file found on the Transformation Services server in *`<Documentum Content Transformation Services_HOME>`*`/jetty/base` with the following details:

      Note: Ensure that the port numbers used for SSL configuration are not the same as the existing Jetty port number (default port value is `9096`).

```
# Connector port to listen on
jetty.http.port=<ssl_port>
jetty.ssl.port=<ssl_port>
jetty.httpConfig.securePort=<ssl_port>
# KeyStore absolute file path
jetty.sslContext.keyStoreAbsolutePath=<keystore path>
# TrustStore absolute file path
jetty.sslContext.trustStoreAbsolutePath=<truststore path>
# KeyStore password
jetty.sslContext.keyStorePassword=<password>
# KeyManager password
jetty.sslContext.keyManagerPassword=<password>
# TrustStore password
jetty.sslContext.trustStorePassword=<password>
```

   b. In the `WebServerService.xml` file located at `<|`*`Documentum Home`*`|`*`CTS`*`|`*`config`*`|>`, set the value of `<EnableHttps>` to `true`.

```
<ServerProperty Key="EnableHttps" Description="Enable SSL configuration"
Type="String" Value="false"/>
```

   c. To verify the new SSL URL, run the following URL from the Transformation Services server browser: `https://`*`<localhost>`*: *`<ssl_port>`*`/cts/`

   d. If the URL is successful, update the cts_instance_info object in the repository to set the new URL to websrv_url attribute by running the following query:

```
Update cts_instance_info object set websrv_url=<new_ssl_url> where
hostname=<ctsserver_hostname_here>
```

   e. Restart the Transformation Services server.

   f. Restart Transformation Services WebServices.

3.1.22 Automating thumbnail or proxy creation for custom formats

To automate thumbnail or proxy creation for custom formats derived from an existing rich_media_enabled supported Transformation Services dm_format:

1. Run the following Documentum Query Language (DQL) to set the *`richmedia_enabled`* flag to `1` for the custom format:

   ```
   update dm_format object set richmedia_enabled = 1 where name='custom_format_name'
   ```

2. If the dm_format name and dos_extension are different for a custom format, the `FormatMapperService.xml` file (in `\CTS\config\` folder) should be updated as follows:

   ```
   <FormatMapper CTSFormat="dos_extension" DocumentumFormat="dm_format_name"/>
   ```

3. Check out the following system profiles from the repository:

   - register_legacy (provided that LEGACY is turned on as forClients in the `CTSProfileService.xml` file)
   - register_xcp (provided that xCP is turned on as forClients in the `CTSProfileService.xml` file)
   - register_d2 (provided that client is turned on as forClients in the `CTSProfileService.xml` file)
   - thumbnail (if it is an image format)
   - autoGenProxy (if it is an image format)
   - pdf_processing, thumbnail_pdfstoryboard, autoGenProxy_pdfstoryboard, and storyboard_pdfstoryboard (if the custom format is a flavor of PDF format)
   - powerpoint_registration (if the custom format is a flavor of PowerPoint format)

4. Add the custom format entry to the appropriate profiles.

5. Check in the profiles.

6. Restart the Transformation Services.

3.1.23 Performing bulk transformation requests for documents

Sometimes, it is useful to submit transformation requests in bulk. Such scenarios can occur when you have legacy documents that were ingested prior to deployment of a Transformation Services. In addition, you may want to resubmit a number of documents for a specific job because Transformation Services failed to process them initially and the job requests were removed.

Transformation Services provides a tool that can request transformation requests for a group of documents. This tool is deployed in `<Documentum Content Transformation Services_HOME>\docbases\<your_repository>\CTSServerScript\`.

Use `\config\script.xml` to configure the process, then use `\bin\script.bat` to run it. By default, the tool uses the `script.xml` but you can also modify the script batch file to use a different configuration file.

OpenText Documentum Content Management - Transformation Services Development Guide (EDCCT250400-PGD) provides details on how to use the batch script to submit bulk transformations.

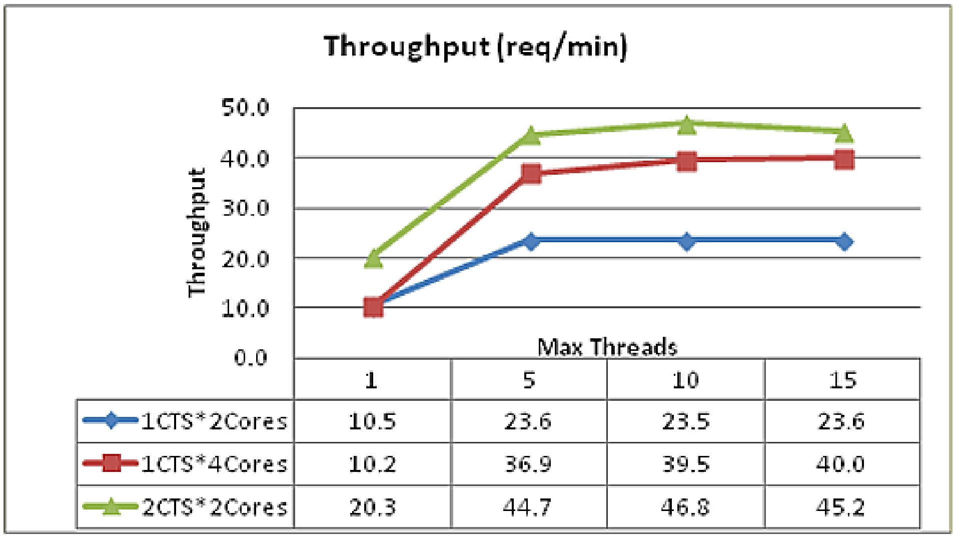
3.1.24 Improving the performance of Transformation Services

Note: The best practices and test results are derived or obtained after testing the product in the testing environment. Every effort is made to simulate common customer usage scenarios during performance testing, but actual performance results vary due to differences in hardware and software configurations, data, and other variables.

The following suggestions may be implemented to achieve a better throughput with Transformation Services when used in an asynchronous transformation scenario. Asynchronous refers to a scenario where a client submits a batch of transformation requests to the Documentum CM Server, and the Transformation Services servers subsequently fetches tasks from the Documentum CM Server queue asynchronously, to invoke the transformation process and import the renditions into the repository.

- Transformation Services delivers a better performance for transformation of documents when compared to the transformations in previous Transformation Services releases, such as Transformation Services - Media, Transformation Services - Documents, and Transformation Services - Audio/Video.
- Except for PDF to PDFA transformations which involve Ghostscript processing, Transformation Services stands to gain over Transformation Services - Documents the most in terms of performance.
- Better throughput (requests/minute) can be achieved with multithreaded Transformation Services configuration. The configured number of working threads supplied with Transformation Services is enough to derive a throughput improvement compared to a single threaded configuration.

- For a given configuration of CPU and memory, an increase in number of threads may not increase throughput linearly. It flattens out at some point where resource utilization becomes a bottleneck.
- To achieve more throughputs, more CPU cores can be added to a Transformation Services server.

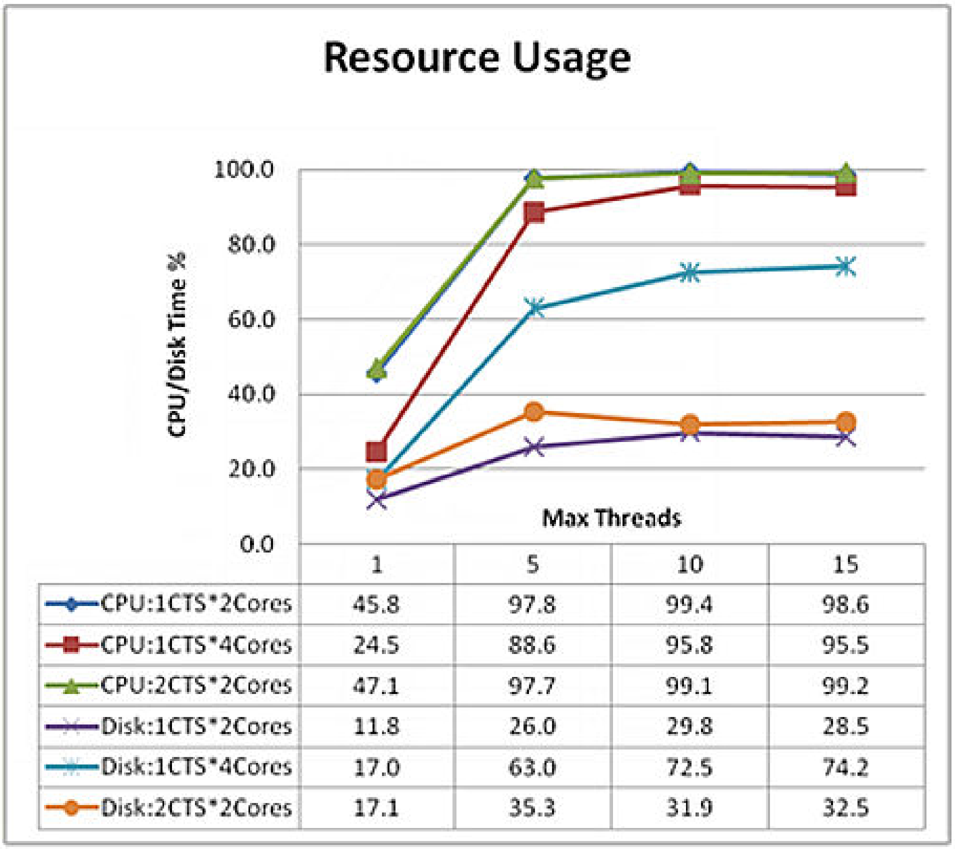


Max Threads	1	5	10	15
1CTS*2Cores	10.5	23.6	23.5	23.6
1CTS*4Cores	10.2	36.9	39.5	40.0
2CTS*2Cores	20.3	44.7	46.8	45.2

Figure 3-1: Throughput

- To achieve higher throughputs with large scale transformations and high availability environments, additional Transformation Services servers can be added.

- Better performance and throughput can be achieved when Transformation Services is deployed on dedicated servers, as with increased throughput comes increased resource utilization.



	1	5	10	15
CPU:1CTS*2Cores	45.8	97.8	99.4	98.6
CPU:1CTS*4Cores	24.5	88.6	95.8	95.5
CPU:2CTS*2Cores	47.1	97.7	99.1	99.2
Disk:1CTS*2Cores	11.8	26.0	29.8	28.5
Disk:1CTS*4Cores	17.0	63.0	72.5	74.2
Disk:2CTS*2Cores	17.1	35.3	31.9	32.5

Figure 3-2: Transformation Services resource usage

3.1.25 Transformation Services clean up job

Transformation Services provides a generic clean up job utility for cleaning up orphaned or invalid objects from the repository, provided that the deletion can be performed through a specific, configured DQL.

The following unused transformation related requests or response objects are examples of orphaned objects:

- During a transformation request using Transformation Services, a transformation request object of the dm_transform_request type is created in the repository. This object may remain in the repository if the *clearOnExit* flag on the CTSRequest object is not set to TRUE.
- Similarly, a dm_cts_response object is created in the repository during every transformation process, if the reporting configuration is enabled in the repository. These objects can get accumulated in the repository over time causing a disk space issue.

A Server Java method is implemented for the Transformation Services CleanUp Job. The Transformation DAR is responsible for deploying this server job and performing additional repository object updates. The job is made inactive Out-of-the-Box. If required, the job can be enabled through Documentum Administrator.

The execution of this method is made further configurable using an XML file. This job can be used to clean up objects, provided that the objects can be identified through a DQL. Out-of-the-Box, this job is used to delete orphaned objects from transformation or response objects.

3.1.25.1 Job configuration options

Jobs are made further configurable using `ctsJobs.xml`, which is deployed in the repository as part of DAR installation process. This configuration option is provided for the easy enhancement of this feature. The configuration file, `ctsJobs.xml`, is deployed into the `/System/Applications/Transformation/` repository folder as part of the DAR installation process.

One or more <*Job*> tags can be associated with any job to initiate more than one deletion process. For example, the same default cts_cleanup method or job can be used for deleting both dm_transform_request and dm_cts_response objects, where these objects are detected through different DQLs. Out-of-the-Box, the following DQLs are configured in `ctsJobs.xml` for deleting request or response objects.

- Cleanup DQL entry for deleting transformation request objects:

```
<Job name="cts_cleanup" description="delete the orphaned transformation
related request objects">
<ServerProperty Key="delegate_class" Description="delagate class"
Value="com.emc.documentum.cts.job.services.DefaultCleanUpService" />
<ServerProperty Key="dql" Value="select r_object_id from dm_transform_request
 where DATEDIFF(day, "r_creation_date", DATE(TODAY))>10" Description="DQL
to identify the object to be deleted" />
</Job >
```

- Cleanup DQL entry for deleting transformation response objects:

```
<Job name="cts_cleanup" description="delete the invalid transformation
related response objects">
<ServerProperty Key="delegate_class" Description="delagate class"
Value="com.emc.documentum.cts.job.services.DefaultCleanUpService" />
<ServerProperty Key="dql" Value="select r_object_id from dm_cts_response
 where DATEDIFF(day, "r_creation_date", DATE(TODAY))>10" Description="DQL
to identify the object to be deleted" />
</Job >
```

3.1.25.2 Transformation Services job Log4j configuration

You can update the following `log4j2.properties` file to make this log file for this job available on the Documentum CM Server:

```
$Documentum\wildfly<supported-version>\server\DctmServer_MethodServer\deployments
\ServerApps.ear\APP-INF\classes\log4j2.properties
```

Note: Ensure that the paths are resolved and accessible in the machine.

```
appender.FJobAppender.type=RollingFile
appender.FJobAppender.name= CTSJobsAppender
appender.FJobAppender.fileName=$DOCUMENTUM/wildfly<supported-version>/server/
```

```
DctmServer_MethodServer/logs/cts_jobs.log
appender.FJobAppender.filePattern=$DOCUMENTUM/wildfly<supported-version>/server/
DctmServer_MethodServer/logs/cts_jobs.%d{yyyy-MM-dd}-%i
appender.FJobAppender.layout.type=PatternLayout
appender.FJobAppender.layout.pattern=%d{ABSOLUTE} %5p [%t] %c - %m%n
appender.FJobAppender.policies.type=Policies
appender.FJobAppender.policies.time.type=TimeBasedTriggeringPolicy
appender.FJobAppender.policies.time.interval=1
appender.FJobAppender.policies.time.modulate=true
appender.FJobAppender.policies.size.type=SizeBasedTriggeringPolicy
appender.FJobAppender.policies.size.size=100MB
appender.FJobAppender.strategy.type=DefaultRolloverStrategy
appender.FJobAppender.strategy.max=5

logger.RJobAppender.name = com.emc.documentum.cts.job
logger.RJobAppender.level = INFO
logger.RJobAppender.additivity = false
logger.RJobAppender.appenderRef.FJobAppender.ref = CTSJobsAppender
```

3.1.26 Changing the hostname of the Transformation Services server machine

To change the name of the host machine when a Transformation Services product is installed and configured against a repository, perform the following configuration changes:

1. Stop all Transformation Services services.
2. Update the host name in the following fields of the Windows Registry entry:

   `agentURL` and `wsurl` in `HKEY_LOCAL_MACHINE\SOFTWARE\Documentum\CTS\`

   `Hostname` in `HKEY_LOCAL_MACHINE\SOFTWARE\Documentum\CTS\AdminAgent\`
3. Rename the key to reflect the new host name in the Windows Registry entry: `HKEY_LOCAL_MACHINE\SOFTWARE\Documentum\CTS\MTS\Docbases[repository name][host name]\`
4. Update the Transformation Services configuration file located at `C:\Documentum\CTS\config\CTSServerService.xml`.
   - Take a backup of the `CTSServerService.xml` file.
   - Search and replace references to the old host name with the new host name in the `CTSServerService.xml` file.
5. Update the cts_instance_info object with the following DQL command:

   ```
   UPDATE cts_instance_info OBJECT SET hostname = '[newName]', SET agent_url =
   'http://[newName]:9095/CTSAgent/CTSAgent', SET websrv_url = 'http://[newName]:
   9096/cts/' WHERE hostname = '[oldname]'
   ```
6. Change the host name and restart the Transformation Services host.

3.1.27 Resolving issues with email content containing double byte characters

When you import emails with double byte characters in the subject, content, and attachment names, it is seen that in the subject and attachment names, the double byte characters are corrupted and appear as `???`. For the renditions to be created without character corruption, it is recommended that you install Arial Unicode Microsoft font on the Transformation Services server machine.

3.1.28 Localization in Transformation Services

Transformation Services does not support the localization of the attributes in the email message. Therefore, it does not translate the attributes in the email message like, From, To, Subject, Send date, Receive date, Attachment, CC, and BCC to the system locale.

3.1.29 PDF compliance not working

For PDF/A 1b and PDF/X-3:2002 compliance, the supported version of Ghostscript is required. The product *Release Notes* on My Support (support.opentext.com) provides information about the supported version.

To enable PDF/A 1b compliance, install Ghostscript, edit `<GS>/lib/PDFA_def.ps`, and update the following line:

Change:

```
/ICCProfile (ISO Coated sb.icc) % Customize.
```

To:

```
/ICCProfile (C:/PROGRA~1/gs/gs<supported-version>/iccprofiles/srgb.icc) % Customise.
```

Note: To enable PDF/X-3:2002 compliance, install the supported version of Ghostscript, edit `<GS>/lib/PDFX_def.ps` file, and update the following line:

Change:

```
/ICCProfile (ISO Coated sb.icc) def % Customize or remove.
```

To:

```
%/ICCProfile (ISO Coated sb.icc) def % Customize or remove.
```

In a few environments with Ghostscript 9.50 or later, the system fonts are not found. In such cases, perform the following steps to specify the path and direct Ghostscript to query the font:

1. Check out the `to_pdfa_doc3.xml` command line profile from the repository.
2. Add the `-sFONTPATH` parameter and the value of the fonts file path after the `dNOPAUSE` parameter line.

For example, `<PARAMETER name="-sFONTPATH=C:/Windows/Fonts"/>`.

3. Check in the `to_pdfa_doc3.xml` command line profile to the repository.
4. Restart the Transformation Services.

3.1.30 Configuring WebServices in a single sign-on environment

Single sign-on (SSO) environment is enabled with OpenText Directory Services (OTDS) authentication.

3.2 Configuration tasks specific to Transformation Services - Documents

This section contains procedures for the document-based configuration tasks that are performed with Transformation Services - Documents.

3.2.1 Creating thumbnails and storyboards for MSG assets

To create thumbnails, preview, and storyboards for the MSG format on import or registering, proceed as follows:

The `register_xcp.xml` contains the following `InnerProfile` OOTB for creating thumbnails and storyboards for the MSG assets:

```
<InnerProfile path="/System/Media Server/System Profiles/msg_
registration_xcp" waitOnCompletion="true" useTargetFormat="true">
  <InnerTokenMapping LocalProfileToken="true" InnerProfileToken=
"overwrite_rendition" Literal="true"/>
 <InnerTokenMapping InnerProfileToken="doc_token_targetFormat"
Literal="true" LocalProfileToken="pdf"/>
</InnerProfile>
```

Add the `InnerProfile` content to the `register_legacy.xml`, if the value of the *`ForClients`* tag in the `CTSProfileService.xml` file is `LEGACY`.

3.2.2 Configuring Doc4 plug-in for Dynamic XFA PDF forms

PDF source files with dynamic XFA forms fail to generate valid renditions if *`<HAS_XFA_PDF>`* option is not set to `true`. To enable processing of dynamic XFA PDF forms, configure the following option in the *`<Documentum Content Transformation Services_HOME>`*`\config\doc4\doc4.xml` file:

```
<HAS_XFA_PDF>true</HAS_XFA_PDF>
```

3.2.3 Configuring Doc8 plug-in

1. Create the `poppler` folder in the `<Documentum_Home>\CTS\lib\` file.

2. Download the `poppler-0.51_x86.7z` version of Poppler Windows binary to a file path on your computer.

3. Copy all the contents of the `bin` folder from the Poppler archive and paste it in `<Documentum_Home>\CTS\lib\poppler\`.

4. Modify the `doc8.xml` file in `<Documentum_Home>\CTS\config\doc8\` as follows:

   ```
   <PROCESS name="poppler\pdftops.exe"/>
   ```

5. Restart the Transformation Services.

3.2.4 Configuring Doc9 plug-in

1. Create the `poppler` folder in `<Documentum_Home>\CTS\lib\`.

2. Download the `poppler-0.51_x86.7z` version of Poppler Windows binary to a file path on your computer.

3. Copy all the contents of the `bin` folder from the Poppler archive and paste it in `<Documentum_Home>\CTS\lib\poppler\`.

4. Modify the `doc9.xml` file in `<Documentum_Home>\CTS\config\doc9` as follows:

   ```
   <PROCESS name="poppler\pdftohtml.exe"/>
   ```

5. Restart the Transformation Services.

3.2.5 Configuring Doc11 plug-in

Blazon server integration enables high fidelity Microsoft Office formats and support for AutoCAD formats. Blazon server utilizes Microsoft Office installation to generate PDF output and is suitable for customers who need fidelity over performance. Since Microsoft Office based transformations are single threaded, Doc11 cannot match multi-threaded performance of Aspose-based Doc6 plug-in.

To enable the Doc11 plug-in:

1. Shut down Transformation Services - Documents.

2. Install the supported version of Blazon Server.

   The product *Release Notes* on My Support (support.opentext.com) contains information about the supported versions.

   Note: Before beginning the installation, see *OpenText Blazon Enterprise Administration Guide* to install the prerequisite software. Ensure that all Microsoft Office products are registered with valid product keys. Open each Microsoft Office application and ensure that no pop-up message appears.

3. During the Blazon installation, choose only **Job Processor** and its subcomponents. Ensure that you disable the **Queue Server** option.

4. After you install Job Processor, stop the Blazon Job Processor windows service.

5. Copy a valid Blazon server license file `IGCKey.lic` to the `C:\Program Files (x86)\OpenText\Blazon Enterprise\JobProcessor\` folder. Otherwise, the server runs in an evaluation mode and might add a watermark to the output PDF files.

6. Open the configuration file located at `C:\Program Files (x86)\OpenText\Blazon Enterprise\JobProcessor\JobProcessor.config\`.

7. Comment out all the lines between `#jobgetter.classname.0=igc.jobprocessor.PopProcessorJobGetter` and `#JobGetter.DirectoryWatcherExample`.

8. Add the following lines in `# JobGetter.DirectoryWatcherExample`:

   ```
   jobgetter.classname.0=igc.jobprocessor.DirectoryWatcherJobGetter
   job.file.dir.0=C:\Documentum\CTS\plugins\doc11
   dir.watcher.sleep.time.0=1000
   ```

   Note: Ensure that the Transformation Services install path in the preceding lines is accurate.

9. Restart the `jobprocessor` service.

10. Verify if the JobProcessor status URL is accessible: `http://localhost:7070/status/`

11. Restart the Transformation Services service. Otherwise, the Doc11 plug-in is disabled.

12. To collect logs, enable DEBUG logging of the Doc11 plug-in in the `CTS/config/log4j2.properties/` file.

13. After the Transformation Services startup, check the log files, `CTS_log.txt` and `Doc11_log.txt` to confirm if the Doc11 plug-in is successfully loaded.

You can customize the output PDF based on various configuration options provided by Blazon Server. *OpenText Blazon Enterprise Administration Guide* provides the complete list of available job options. Some of the relevant job options are added to the command line profile `document_to_pdf_doc11.xml`. You can add more options if required under the `CUSTOM` scope. You can provide the `doc_token` values as realtime profile parameters or add to document registration profiles such as `document_registration_xcp.xml` or `document_registration_d2.xml` depending on the client type.

Note: When the Doc11 plug-in fails to render documents, the transformation is automatically forwarded to the Doc6 plug-in due to the plug-in failover feature of Transformation Services - Documents. This gets unnoticed unless you check the log files. If you want the Doc11 plug-in failures to be reported, then disable the Doc6 plug-in in `document_to_pdf` and `document_to_custom_pdf` system profiles by commenting out the Doc6 plug-in option.

Table 3-3: Formats supported by Doc11 plug-in

Documentum format	DOS extension
DGN	DGN
DWF	DWF
DWFX	DWFX
DWG	DWG
DXF	DXF
EXCEL8BOOK	XLS
EXCEL12BOOK, EXCEL14BOOK, EXCEL15BOOK	XLSX
MSPROJECT	MPP
MSW8	DOC
MSW12, MSW14, MSW15	DOCX
PPT8_TEMPLATE	POT
PPT12, PPT12_SLIDE, PPT14, PPT14_SLIDE, PPT15, PPT15_SLIDE	PPTX
PPT8SLIDESHOW	PPS
PPT12SLIDESHOW, PPT14SLIDESHOW, PPT15SLIDESHOW	PPSX
VSD	VSD
VSDX	VSDX
HEIC	HEIC
msg	msg
email	EML

3.2.5.1 Enabling OCR support

With the Doc11 plug-in, you can convert documents with text-oriented images into searchable documents using Image Text Recognition or Optical Character Recognition (OCR) technology.

The OCR feature of Blazon requires an add-on purchase and is available when licensed.

You can submit one or more source documents of the following formats: TIFF, BMP, PNG, PDF, or JPG, with job parameter *`DoOCR`* set to `true`. The source document is automatically processed and any image text in the source document becomes searchable in the target PDF file.

You must enable the *`DoOCR`* job parameter in the command line file (CLF) `document_to_pdf_doc11` or in `legacy_to_pdf` by setting *`doc_token_blz_OCREnable`* value to `true`. You can configure additional parameters for OCR accuracy in `document_to_pdf_doc11`. *OpenText Blazon Enterprise Administration Guide* provides a detailed description of these parameters.

Note: For better text recognition results, the raster resolution of the source image must be at least 300 x 300 DPI before performing OCR.

3.2.5.2 Enabling additional source format support

If the source format is not listed in “Formats supported by Doc11 plug-in” on page 74 but is supported by Blazon, enable the source formats by completing the following procedure:

Note: In this section, we are adding a sample extension `xyz`.

1. Verify if the source format is available in dm_format:

   ```
   select * from dm_format where name='xyz'
   ```

2. If the format is not available, create it with the following DQL statements:

   ```
   create "dm_format" object set "name"='xyz',
   set "description" = 'Auto xyz format',
   set "can_index" = 1,
   set "topic_transform" = 0,
   set "mac_creator" = '????',
   set "dos_extension" = 'xyz',
   set "is_hidden" = 0,
   set "mime_type" = 'application/xyz',
   set "icon_index" = 0,
   set "richmedia_enabled" = 0
   ```

   To enable rendition generation of the format after importing the document into the repository, set the value of *`richmedia_enabled`* attribute to `1`. When richmedia flag is enabled, a transformation request is generated each time a new document is added to the repository.

3. To enable automatic processing of the document, update the `register.xml` system profile and the related `register*.xml` profile to process the new format. For more information, see "Register Profile" on page 144.

4. Update the `document_registration_*`, `legacy_to_pdf`, and `document_to_pdf` profiles to include the new format as a source format.

5. Add the new source format to `C:\Documentum\CTS\config\doc11\doc11.xml` in the *<SUPPORTED_FORMATS>* tag. For example:

```
<SOURCE format="xyz">
<TARGET format="pdf"/>
</SOURCE>
```

6. Add the new source format in the appropriate Blazon queues. The Blazon `Doc` queue renders document format extensions, `PDF` queue renders PDF documents, and `DRW` queue renders image and drawing formats. For more information about the input file types and format-queue mapping, see *OpenText Blazon Enterprise Administration Guide*.

   Add the new source format to `C:\Documentum\CTS\config\doc11\doc11.xml` in the *<DOC_QUEUE>* tag. For example:

```
<DOC_QUEUE THREADS="2">
<format>xyz</format>
</DOC_QUEUE>
```

Note: Even though new formats are added, only the certified or supported formats will function as expected. For a list of supported formats, see the following appendixes:

- Appendix A, Formats supported by Transformation Services - Documents on page 217
- Appendix B, Formats supported by Transformation Services - Media on page 219
- Appendix C, Formats supported by Transformation Services - Audio/ Video on page 223
- Appendix D, Formats supported by XML Transformation Services on page 227

3.2.5.3 Enabling Blazon Redaction support

Blazon Enterprise provides Redaction module to intelligently remove (redact) sensitive content and privacy information from documents. You can configure and enable the Blazon Redaction in Transformation Services.

To enable the Blazon redaction feature:

1. Add the `RedactionScriptFilename` property in `General properties` in the `document_to_pdf_doc11` command line file, and provide the path of the redaction script file including the name of the file as follows:

```
<PROPERTY name="RedactionScriptFilename" value="C:\Blazon_test\redaction.script"/
><PROPERTY name="RedactionScriptFilename" value="C:\Blazon_test\redaction.script"/>
```

2. Save the redaction script file with the `.script` extension, in the specified path of the Transformation Services machine.

```
Sample redaction script--
Sample (txt):
Unable to render embedded object: File (redact) not found.
email
[:email:]
Note:- it should be able to redact all the occurrences of the string "email" (case
sensitivity and matching criteria can be further enhanced using scripts, refer to
blazon guide)
[:email:] macro should redact all the email ids (if present in a proper format in
the document).
```

You can provide other redaction options using the same redaction script.

3.2.6 Configuring Doc12 plug-in

The Doc12 plug-in in Transformation Services uses the Transformation Publication-Service of OpenText Intelligent Viewing as a rendition engine for end users to view diverse content formats stored in the repository with a single-click viewing application.

1. Stop Transformation Services service.
2. Install OpenText Intelligent Viewing and OpenText Directory Services in a separate environment. For information about the installation instructions, see *OpenText Intelligent Viewing Installation Guide*.

   Note: After installing OpenText Intelligent Viewing, you must add `TRUSTED_SOURCE_ORIGINS` with `http://`*`<cts_ip>`*`:9095`, where *<cts_ip>* is the CTS URL and 9095 is the default port number, in the respective PublisherService and PublicationService's `env.conf` files.

3. Create an OAuth client in OTDS and ensure that you create a confidential client with the following configurations:
   - To create a confidential client, ensure that you select the **Confidential** check box.
   - In the **Advanced** page:
     - Set **Access token lifetime(seconds)** to a positive number.
     - Add the following **Permissible scopes** and **Default scopes** values – *`view_publications`*, *`create_publications`*, *`delete_publications`*, *`view_any_publication`*, and *`delete_any_publication`*.
     - Make a note of the *`CLIENT_ID`* and *`CLIENT_SECRET`* values. These values are required during configuration of the `doc12.xml` file.
4. Set up the CTSFileServer.

a. Extract the `CTSFileServer.war` file and save at the *<Documentum_Home>*`\CTS\AdminAgent\webapps` file path.

b. Open the `application.properties` file available at *<Documentum_Home>*`\CTS\AdminAgent\webapps\CTSFileServer\WEB-INF\classes` and modify the OTDS server and the Doc12 plug-in details.

5. Configure the `doc12.xml` file.

   a. Modify the Intelligent Viewing parameters and the OTDS Authorization server parameters in the `doc12.xml` file. The *CLIENT_ID* and *CLIENT_SECRET* values should be the same from step 3 during the OTDS OAuth client creation.

   Note: Ensure that you do not change the *GRANT_TYPE* and *SCOPE* parameter values.

   b. Modify the value of the *CTS_FILE_SERVER* tag with the IP address or machine name. For example, `http://`*<Documentum Content Transformation Services_host_name>*`:9095/CTSFileServer`, where `9095` is the default port number where CTSFileServer is deployed and should not be changed.

6. Start the Documentum Content Transformation Services AdminAgent first and then the Transformation Services service.

Table 3-4: Formats supported by Doc12 plug-in

Documentum format	DOS extension
DGN	DGN
DWF	DWF
DWFX	DWFX
DWG	DWG
EXCEL8BOOK	XLS
EXCEL12BOOK, EXCEL14BOOK, EXCEL15BOOK	XLSX
MSW8	DOC
MSW12, MSW14, MSW15	DOCX
PPT12, PPT12_SLIDE, PPT14, PPT14_SLIDE, PPT15, PPT15_SLIDE	PPTX
PPT8SLIDESHOW	PPT
CRTEXT	TXT
JPG	JPG
JPEG	JPEG
PNG	PNG

Documentum format	DOS extension
GIF	GIF
PDF	PDF
HEIC	HEIC

3.2.6.1 Enabling Intelligent Viewing features for Doc12 plug-in

Enabling the ExportPdf 1.1.0 feature of OpenText Intelligent Viewing provides additional PDF settings such as enabling fast web view, generating specific isoConformance PDF, and generating specific version of PDF.

To enable ExportPdf 1.1.0 features:

1. Check out the `document_to_pdf_doc12.xml` file.
2. Uncomment the following line:

```
<FEATURE NAME="ExportPdf" NAMESPACE="opentext.publishing.renditions"
VERSION="1.1.0" JSON=""/>
```

3. Check in the `document_to_pdf_doc12.xml` file.
4. Navigate to *<Documentum_Home>*`/CTS/config/doc12/json/` and open the `ExportPdf_1.1.0.json` file.
5. Modify the key-value pairs in the value tags *`isoConformance`*, *`requestedVersion`*, and *`fastWeb`* as required.

Element name	Acceptable values	Description
`isoConformance`	Default: `none` Valid values: `a1a`, `a1b`, `a2b`, `a2u`, `a3a`, `a3b`, `a3u`, `e`	This property is used to inject the corresponding conformance tag into the generated PDF artifact and in some cases alters the PDF internals to avoid conformance violations.
`requestedVersion`	Default: `1.4` Valid values: `1.4`, `1.5`, `1.6`, `1.7`	This property changes the PDF version of the generated document from the default of 1.4 to the specified version.
`fastWeb`	Default: `false` Valid values: `true`, `false`	This property generates a linearized PDF that can be more efficiently viewed as a stream of pages.

3.2.6.2 Adding OTDS certificate to dfc.keystore on Transformation Services host

To add OTDS certificate to `dfc.keystore` when Documentum CM Server, OpenText Intelligent Viewing, and OTDS are SSL enabled, perform the following steps:

1. Go to the OTDS website *`https://<otds_url>/otds-admin`* and click **View Site Information** (lock button) in the browser address bar.

2. Navigate to the **Details** tab on the **Certificate** screen.

3. Click **Copy to File**.

4. In **Certificate Export Wizard**, click **Next**.

5. In **Export File Format**, click **Next**.

6. In **File to Export**, browse the file path, type the file name, and then click **Save**.

7. Navigate to `JAVA_HOME/bin/` and run the following command:

```
keytool -import -v -trustcacerts -alias <downloaded certificate name> -file
<downloaded certificate path> -keystore <keystore path name mentioned in the
dfc.properties file> -keypass Password@123
```

8. Type the default password of `dfc.keystore`, <*downloaded certificate name*> `Password@123` and then install the certificate.

9. Restart the <*Documentum CTS Admin. Agent*> first and then <*Documentum Content Transformation Services*>. Verify the `Doc12_log.txt` file.

3.2.7 Microsoft Information Protection for Transformation Services

Transformation Services supports Microsoft Information Protection (MIP) to transform MIP-protected documents.

Note: The MIP feature is available only in cloud deployment.

The MIP-protected documents are processed only if the MIP processing is enabled. The MIP configuration is performed in the Documentum Client Smart View Business Administration application. For more information about configuring MIP, see *OpenText Documentum Content Management* documentation.

Table 3-5: Supported formats

Documentum format	DOS extension
Microsoft Word (DOC, DOCX)	doc, docx
Microsoft Excel (XLS, XLSX)	xls, xlsx
Microsoft PowerPoint (PPT, PPTX)	ppt, pptx

Documentum format	DOS extension
PDF (PDF)	pdf
email	EML

The following table lists the limitations in Transformation Services features, while supporting the MIP services:

Table 3-6: Limitations

Feature	Limitation
PDF merge	PDF merge is not supported if documents have different labels or if any one of the documents is not attached with a label.
Email with attachment	Emails with PDF attachments that have MIP labels are not supported.
File In File Out (FIFO)	FIFO transformation is not supported for documents marked with MIP labels.
File In Repo Out (FIRO)	FIRO transformation is not supported for documents marked with MIP labels.

3.2.8 Controlling the size of PDF renditions

Conversion of Microsoft Office documents to PDF results in large-sized documents.

To create small-sized PDF renditions from Microsoft Word documents, set the following options in the `document_to_pdf.xml` command line file:

```
<PDFSETTINGS TEXTCOMPRESSION="Flate" PRESERVEFORMFIELDS="false">
<FONTEMBEDDING ENABLED="Yes" PARTIALFONTS="Yes" STANDARDFONTS="No"/>
</PDFSETTINGS>
```

Note: Transformation Services must be restarted for the command line file changes to take effect.

3.2.9 Embedding fonts for PDF/A renditions

Conversion of Microsoft Office documents to PDF/A requires all fonts to be embedded in the PDF file.

To enable embedding of fonts in the PDF file, add the following tag as the child of the `PDFSETTINGS` element in the `document_to_custom_pdf.xml` command line file:

```
<PDFSETTINGS TEXTCOMPRESSION="Flate" PRESERVEFORMFIELDS="false">
<FONTEMBEDDING ENABLED="Yes" STANDARDFONTS="No" PARTIALFONTS="Yes"/>
</PDFSETTINGS>
```

3.2.10 PDF customization

This section provides information about the Doc6 plug-in customizations, which are exposed to the users through command line files and system profiles. These customizations are specific only to the transformation of Microsoft Office documents to the PDF format. The customization options are included in the document_to_pdf command line profile and can be configured by specifying profile parameters (*InnerTokenMapping*) through the following system profiles:

- document_registration
- legacy_to_pdf.xml
- document_registration_xcp

The following sections provide information about XML elements that are found in the document_to_pdf command line profile:

TRANSFORMATION: This is the root element of the job ticket and contains the settings provided as inputs to the Doc6 plug-in to process the documents.

Element name	Acceptable values	Description
TRANSFORMATION	This element has no attributes and does not accept values.	Root element

SETTINGS: This is the child of the TRANSFORMATION element and contains all the customization settings available for the Doc6 plug-in.

Element name	Acceptable values	Description
SETTINGS	This element has no attributes and does not accept values.	Denotes the customization settings available for the Doc6 plug-in.

The following are the child elements of the SETTINGS element:

- “MSWORDSETTINGS” on page 83
- “MSEXCELSETTINGS” on page 104
- “MSPPTSETTINGS” on page 105
- “METADATAELEMENTS” on page 106

3.2.10.1 MSWORDSETTINGS

This is the child of the `SETTINGS` element and contains customizations applicable to source documents that are in the Microsoft Word format.

Element name	Acceptable values	Description	Contains profile parameter
UPDATEFIELDS	Yes, No (Default value)	Specifies if the fields in the Microsoft Word document should be updated.	Yes doc_token_updateFields

3.2.10.1.1 TOC

This is the child of the MSWORDSETTINGS element and allows users to:

- Generate the Table of Contents, Table of Tables, and Table of Figures
- Control the look and feel of Table of Contents

Element name	Acceptable values	Description	Contains profile parameter
ENABLED	Yes, No (Default value)	Enables or disables the creation of Table of Contents, Table of Figures, and Table of Tables.	No
CONTENTBOOKMARKS	Yes (Default value), No	Enables or disables the creation of a bookmark on top of Table of Contents, which reads `Table of Contents`.	Yes doc_token_contentBookmarks
DOCUMENTBOOKMARKS	Yes (Default value), No	Enables or disables the creation of a bookmark at the start of the document content.	Yes doc_token_documentBookmarks
MARGINBOTTOM	Number (Default value is 0.5)	Specifies the bottom margin of the Table of Contents page.	No
MARGINLEFT	Number (Default value is 0.5)	Specifies the left margin of the Table of Contents page.	No
MARGINRIGHT	Number (Default value is 0.5)	Specifies the right margin of the Table of Contents page.	No

Element name	Acceptable values	Description	Contains profile parameter
MARGINTOP	Number (Default value is 0.5)	Specifies the top margin of the Table of Contents page.	No
MAXLEVEL	1, 2, 3 (Default value), 4, 5, 6, 7, 8, 9	Specifies the maximum level of headings that Table of Contents must include.	Yes doc_token_toc_maxLevel
ORIENTATION	Portrait (Default value), Landscape	Specifies the page orientation of the Table of Contents page.	Yes doc_token_toc_orientation
SHOWPAGENUMBERS	Yes (Default value), No	Specifies if Table of Contents should display the page numbers for each item.	No
TABLEADERSTYLE	DottedLine (Default value), DashedLine, Underline, None	Specifies the tab leader style between the Table of Contents items and their page numbers.	Yes doc_token_tabLeaderStyle
TOTANDTOFTABLEADERSTYLE	DottedLine (Default value), DashedLine, Underline, None	Specifies the tab leader style between the items of Table of Tables and Table of Figures and their page numbers.	Yes doc_token_tottofLeaderStyle
SHOWTOC	Yes, No (Default value)	Enables or disables the creation of Table of Contents.	Yes doc_token_toc_enabled
SHOWTOT	Yes, No (Default value)	Enables or disables the creation of Table of Tables.	Yes doc_token_tot_enabled
SHOWTOF	Yes, No (Default value)	Enables or disables the creation of Table of Figures.	Yes doc_token_tof_enabled
TOCNAME	Contents (Default value)	Caption for Table of Contents.	Yes doc_token_tocName
TOTNAME	Tables (Default value)	Caption for Table of Tables.	Yes doc_token_totName

Element name	Acceptable values	Description	Contains profile parameter
TOFNAME	Figures (Default value)	Caption for Table of Figures.	Yes doc_token_tofName
EXISTINGTOCPAGE NUMBERCOLOR	BLUE, BLACK, CYAN, DARK_GRAY, GRAY, GREEN, LIGHT_GRAY, MAGENTA, ORANGE, PINK, RED, WHITE, YELLOW	Specifies the TOC page number color.	Font color of the input document.

3.2.10.1.2 HEADER

This is the child of the MSWORDSETTINGS element and allows users to add headers to the generated PDF.

Element name	Acceptable values	Description	Contains profile parameter
ENABLED	Yes, No (Default value)	Enables or disables the creation of headers.	Yes doc_token_headers
FONTNAME	Helvetica (Default value)	Name of the font for the header.	Yes doc_token_h_fontName
FONTSIZE	Numeric (Default value is - 10)	Specifies the font size.	Yes doc_token_h_fontSize
FONTCOLOR	Numeric	Specifies the font color.	Yes doc_token_h_color
LAYER	Foreground (Default value), Background	Specifies the layer where the header is placed.	Yes doc_token_h_layer
TEXTLEFT	Text	Specifies text that is left-justified.	Yes doc_token_h_textLeft
TEXTCENTER	Text	Specifies the center text.	Yes doc_token_h_textCenter

Element name	Acceptable values	Description	Contains profile parameter
TEXTRIGHT	Text	Specifies text that is right-justified.	Yes doc_token_h_textRight
PAGES	All (Default value), First, Last, Even, Odd	Specifies the pages to which headers are applied.	Yes doc_token_h_pages

3.2.10.1.3 FOOTER

This is the child of the MSWORDSETTINGS element and allows users to add footers to the generated PDF.

Element name	Acceptable values	Description	Contains profile parameter
ENABLED	Yes, No (Default value)	Enables or disables the creation of footers.	Yes doc_token_footers
FONTNAME	Helvetica (Default value)	Name of the font for the footer.	Yes doc_token_f_fontName
FONTSIZE	Numeric (Default value is - 10)	Specifies the font size.	Yes doc_token_f_fontSize
FONTCOLOR	Numeric	Specifies the font color.	Yes doc_token_f_color
LAYER	Foreground (Default value), Background	Specifies the layer where the footer is placed.	Yes doc_token_f_layer
TEXTLEFT	Text	Specifies text that is left-justified.	Yes doc_token_f_textLeft
TEXTCENTER	Text	Specifies the center text.	Yes doc_token_f_textCenter
TEXTRIGHT	Text	Specifies text that is right-justified	Yes doc_token_f_textRight
PAGES	All (Default value), First, Last, Even, Odd	Specifies the pages to which footers are applied.	Yes doc_token_f_pages

3.2.10.1.4 MARKUP

This is the child of the MSWORDSETTINGS element and controls the generation of markup and comments in the PDF.

Element name	Acceptable values	Description	Contains profile parameter
PRINTTYPE	DocumentWithMarkup, DocumentContent (Default value)	Enables or disables the generation of word markup.	Yes doc_token_PrintType
PDFNOTES	Yes, No (Default value)	Enables or disables the generation of word comments.	Yes doc_token_NoteIcons
INSERTIONCOLOR	BLACK, BLUE (Default value), BRIGHT_GREEN, DARK_BLUE, DARK_RED, DARK_YELLOW, GRAY_25, GRAY_50, GREEN, PINK, RED, VIOLET, WHITE, YELLOW, NO_HIGHLIGHT	Applies the specified color to the inserted text.	Yes doc_token_insertionColor
INSERTIONSTYLE	None (Default value), Bold, Italic	Applies the specified style for the inserted text.	Yes doc_token_insertionStyle
DELETIONCOLOR	BLACK, BLUE, BRIGHT_GREEN, DARK_BLUE, DARK_RED, DARK_YELLOW, GRAY_25, GRAY_50, GREEN, PINK, RED (Default value), VIOLET, WHITE, YELLOW, NO_HIGHLIGHT	Applies the specified color to the deleted text.	Yes doc_token_deletionColor
DELETIONSTYLE	None (Default value), Bold, Italic	Applies the specified style for the deleted text.	Yes doc_token_deletionStyle
SHOWREVISIONBAR	Yes (Default value), No	Enables or disables the revision bar.	No

Element name	Acceptable values	Description	Contains profile parameter
REVISIONBARCOLOR	BLACK, BLUE, BRIGHT_GREEN, DARK_BLUE, DARK_RED, DARK_YELLOW, GRAY_25, GRAY_50, GREEN, PINK, RED, VIOLET, WHITE, YELLOW, NO_HIGHLIGHT (Default value)	Applies the specified color to the revision bar.	Yes doc_token_revisionBarColor
FORMATCHANGECOLOR	BLACK, BLUE, BRIGHT_GREEN, DARK_BLUE, DARK_RED, DARK_YELLOW, GRAY_25, GRAY_50, GREEN, PINK, RED, VIOLET, WHITE, YELLOW, NO_HIGHLIGHT (Default value)	Applies the specified color to the text for which the format was changed.	Yes doc_token_formatChangeColor
MOVEFROMSTYLE	None (Default value), Bold, Italic	Applies the specified style for moveFrom markup text.	Yes doc_token_moveFromStyle
MOVETOSTYLE	None (Default value), Bold, Italic	Applies the specified style for moveTo markup text.	Yes doc_token_moveToStyle
MOVEFROMCOLOR	BLACK, BLUE (Default value), BRIGHT_GREEN, DARK_BLUE, DARK_RED, DARK_YELLOW, GRAY_25, GRAY_50, GREEN, PINK, RED, VIOLET, WHITE, YELLOW, NO_HIGHLIGHT	Applies the specified color to the moveFrom markup text.	Yes doc_token_moveFromColor

Element name	Acceptable values	Description	Contains profile parameter
MOVETOCOLOR	BLACK, BLUE (Default value), BRIGHT_GREEN, DARK_BLUE, DARK_RED, DARK_YELLOW, GRAY_25, GRAY_50, GREEN, PINK, RED, VIOLET, WHITE, YELLOW, NO_HIGHLIGHT	Applies the specified color to the moveTo markup text.	Yes doc_token_moveToColor
REVISIONBARPOSITION	LEFT (Default value), RIGHT	Applies the specified control bar position.	doc_token_revisionBarPosition
ACCEPTINSERTIONS	NONE (Default value), ACCEPT, REJECT	Accept or reject the insertion revisions.	doc_token_acceptInsertions
ACCEPTDELETIONS	NONE (Default value), ACCEPT, REJECT	Accept or reject the deletion revisions.	doc_token_acceptDeletions
SHOWINSERTIONS	TRUE (Default value), FALSE	Show or hide the of insertion revisions.	doc_token_showInsertions
SHOWDELETIONS	TRUE (Default value), FALSE	Show or hide the deletion revisions.	doc_token_showDeletions
SHOWMOVES	TRUE (Default value), FALSE	Show or hide the move revisions.	doc_token_showMoves
ACCEPTMOVES	NONE (Default value), ACCEPT, REJECT	Accept or reject the text move revisions.	doc_token_acceptMoves

3.2.10.1.5 ENCODING

This is the child of the MSWORDSETTINGS element and controls to convert plain text object with specific encoding to a PDF rendition.

Element name	Acceptable values	Description	Contains profile parameter
ENCODINGTYPE	Yes, No (Default value)	Enables to convert plain text object with specific encoding to a PDF rendition.	Yes doc_token_textEncodingType

3.2.10.1.6 TEXTSETTINGS

This is the child of the MSWORDSETTINGS element and controls the rendition of the text in the Hindi font in Microsoft Word documents.

Element name	Acceptable values	Description	Contains profile parameter
TEXTSHAPER	Yes, No (Default value)	Enables to rendition the text in the Hindi font in Microsoft Word documents.	Yes doc_token_enableTextShaper

3.2.10.1.7 HIDDENTEXTSETTINGS

This is the child of the MSWORDSETTINGS element and controls the visibility of the hidden text in Microsoft Word documents.

Element name	Acceptable values	Description	Contains profile parameter
SHOWHIDDENTEXT	Yes, No (Default value)	Enables to display or hide the hidden text in Microsoft Word documents.	Yes doc_token_showHiddenText

3.2.10.1.8 HYPERLINKS

This is the child of the MSWORDSETTINGS element and controls the generation of hyperlinks in the PDF.

Element name	Acceptable values	Description	Contains profile parameter
ENABLED	Yes (Default value), No	Enables or disables the generation of PDF hyperlinks.	Yes doc_token_enableHyperlinks
INHERITAPPEARANCE	Yes (Default value), No	Specifies whether to inherit the hyperlink appearance from the source.	Yes doc_token_inheritAppearance
DISPLAYCOLOR	BLUE (Default value), CYAN, MAGENTA, GREEN, YELLOW, ORANGE, PINK, RED, BLACK, DARK_GRAY, LIGHT_GRAY, GRAY, WHITE	Specifies the color of the hyperlink.	Yes doc_token_hyperlinksColor

Element name	Acceptable values	Description	Contains profile parameter
DISPLAYSTYLE	None (Default value), DashedLine, DottedLine, Underline, Wavy	Specifies the underline style for the hyperlink.	Yes doc_token_hyperlinksStyle
INTERNALREFERENCES	REF, NOTEREF, PAGEREF **Note:** It is a comma-separated value.	Specifies the type of references to which you can apply color or style.	Yes doc_token_internalReferences

Note: The Doc4 plug-in does not support hyperlink with named destination.

3.2.10.1.9 NOTES

This is the child of MSWORDSETTINGS element and controls appearance of footnotes and endnotes.

Element name	Acceptable values	Description	Contains profile parameter
ENABLED	Yes, No (Default value)	Enables or disables styling or coloring of footnotes and endnotes.	Yes doc_token_enableNotes
CREATENOTESHYPERLINKS	Yes, No (Default value)	Specifies whether to convert footnote or endnote references into active hyperlinks. When you click the hyperlink, it leads to the corresponding footnote or endnote.	Yes doc_token_createNoteHyperlinks
FOOTNOTECOLOR	BLUE (Default value), CYAN, MAGENTA, GREEN, YELLOW, ORANGE, PINK, RED, BLACK, DARK_GRAY, LIGHT_GRAY, GRAY, WHITE	Specifies the color of the footnote and its description.	Yes doc_token_footnoteColor
FOOTNOTEDISPLAYSTYLE	None (Default value), DashedLine, DottedLine, Underline, Wavy	Specifies the underline style for the footnote and its description.	Yes doc_token_footnoteStyle

Element name	Acceptable values	Description	Contains profile parameter
ENDNOTECOLOR	BLUE (Default value), CYAN, MAGENTA, GREEN, YELLOW, ORANGE, PINK, RED, BLACK, DARK_GRAY, LIGHT_GRAY, GRAY, WHITE	Specifies the color of the endnote and its description.	Yes doc_token_endnoteColor
ENDNOTEDISPLAYSTYLE	None (Default value), DashedLine, DottedLine, Underline, Wavy	Specifies the underline style for the endnote and its description.	Yes doc_token_endnoteStyle

3.2.10.1.10 PDFSETTINGS

This is the child of the MSWORDSETTINGS element and provides customization options that must be specified for the output PDF. It denotes the customization settings applicable to the PDF output documents.

Element name	Acceptable values	Description	Contains profile parameter
TEXTCOMPRESSION	Flate, None (Default value)	Enables the compression of PDF content.	No
PRESERVEFORMFIELDS	False, True (Default value)	Specifies if you need to preserve the Microsoft Word form fields as is in the PDF or convert them to text.	Yes doc_token_preserveFormFields
DOWNSAMPLEIMAGES	Yes (Default value), No	Enables or disables down sampling of images.	Yes doc_token_downSampleImages

Element name	Acceptable values	Description	Contains profile parameter
RESOLUTION	0-600 (Default value is 220)	Specifies the resolution in pixels per inch (ppi) to which the images should be down sampled. Down sampling of images occurs if the reduction in image size is 75 KB or lower. **Note:** When 0 is specified, all images that can be reduced in size are down sampled to 220 ppi.	Yes doc_token_resolution
COLORMODEL	AUTO (Default value), CMYK	Specifies the color space for the images in the PDF document. AUTO mode automatically chooses the most appropriate color space for each image.	Yes doc_token_colorModel

3.2.10.1.10.1 PAGE

This is the child of the PDFSETTINGS element and controls the look of the document pages.

Element name	Acceptable values	Description	Contains profile parameter
ENABLED	No (Default value), Yes	Enables or disables the page setup.	Yes doc_token_enablePageSetup
ORIENTATION	Portrait (Default value), Landscape Landscape	Specifies the page orientation.	Yes doc_token_page_orientation
MARGINLEFT	Numeric (Default value is 0.5)	Specifies the left margin for the document.	No

Element name	Acceptable values	Description	Contains profile parameter
MARGINRIGHT	Numeric (Default value is 0.5)	Specifies the right margin for the document.	No
MARGINTOP	Numeric (Default value is 0.5)	Specifies the top margin for the document.	No
MARGINBOTTOM	Numeric (Default value is 0.5)	Specifies the bottom margin for the document.	No
PAGESIZE	A4 (Default value), Letter	Specifies the page size.	Yes doc_token_page_size
PAGEINDEX	0 (Default value)	Specifies the start page number in the Microsoft Word document from which the PDF must be saved.	No
PAGECOUNT	-1 (Default value)	Specifies the number of pages to save. -1 denotes that all pages must be saved.	No

3.2.10.1.10.2 BOOKMARKS

This is the child of the PDFSETTINGS element and controls the generation of bookmarks in the PDF.

Element name	Acceptable values	Description	Contains profile parameter
ENABLED	Yes (Default value), No	Enables or disables the generation of bookmarks based on the heading styles.	Yes doc_token_enableBookMarks
CREATEDOCUMENTBOOKMARKS	Yes, No (Default value)	Enables or disables the generation of bookmarks based on the heading styles available in the document.	Yes doc_token_enableBookMarks
CREATEBOOKMARKMODE	AutomaticStyles, AndLevels (Default value), DefinedStyles, AndLevels, TCFields	Specifies the mode in which bookmarks are created.	Yes doc_token_bookmarkMode

Element name	Acceptable values	Description	Contains profile parameter
EXPANDEDBOOKMARKLEVEL	1, 2, 3 (Default value), 4, 5, 6, 7, 8, 9	Specifies the level to which the created bookmarks must be expanded.	Yes doc_token_expandedBookmarkLevel
MAXLEVEL	1, 2, 3 (Default value), 4, 5, 6, 7, 8, 9	Specifies the maximum level of heading styles to be included in the bookmarks generated.	Yes doc_token_bookmark_MaxLevel

3.2.10.1.10.3 FONTEMBEDDING

This is the child of the PDFSETTINGS element and controls the embedding of fonts into documents.

Element name	Acceptable values	Description	Contains profile parameter
ENABLED	Yes (Default value), No	Enables or disables the embedding of fonts in the PDF.	Yes doc_token_enableFontEmbedding
PARTIALFONTS	Yes (Default value), No	Controls how fonts are embedded in the PDF documents. If set to `Yes`, subsets of fonts are embedded into documents.	Yes doc_token_partialFonts
STANDARDFONTS	Yes (Default value), No	Specifies whether Windows standard fonts must be embedded.	Yes doc_token_standardFonts

3.2.10.1.10.4 OPENSETTINGS

This is the child of the PDFSETTINGS element and controls how PDF documents should be displayed when opened in a PDF reader.

Element name	Acceptable values	Description	Contains profile parameter
ENABLED	Yes, No (Default value)	Enables or disables the PDF Open settings.	Yes doc_token_enableOpenSettings

Element name	Acceptable values	Description	Contains profile parameter
MAGNIFICATION	None, FitWidth, FitHeight, ZOOM percentages. For example: Zoom200	Specifies the size in which the page is displayed in the document pane.	Yes doc_token_setMagnification
PAGEMODE	PageOnly (Default value), BookmarksAndPage, ThumbnailsAndPage, FullScreen, LayersAndPage	Displays the document in any of the following modes: • Document pane only. • Document pane with bookmarks. • Document pane with thumbnails. • Full-screen mode with no menu bar, window controls, or any other window. • Optional content group panel is visible.	Yes doc_token_pageMode
PAGELAYOUT	SinglePage (Default values), Continuous, ContinuousFacing	Displays the document in any of the following layouts: • One page in the document pane at a time. • Arranges the pages in a continuous vertical column. • Arranges the pages in two columns, side by side.	Yes doc_token_pageLayout
HIDEMENUBAR	Yes, No (Default value)	Enables or disables the menu bar.	Yes doc_token_hideMenubar
HIDETOOLBAR	Yes, No (Default value)	Enables or disables the tool bar.	Yes doc_token_hideToolbar

3.2.10.1.10.5 FIELDCODE

This is the child of the PDFSETTINGS element for transformation of PDF with an automatic field (for example, the date field) to be signed.

Element name	Acceptable values	Description	Container profile parameter
ENABLED	Yes, No (Default value)	Enables or disables the time field.	No doc_token_enableFieldCode

Note: For transformation of PDF/A with an automatic field, enable the field code token in the `document_to_custom_pdf.xml` command line file.

3.2.10.1.10.6 PAGECONTENTSCALING

This is the child of the PDFSETTINGS element and defines how to increase or decrease the page content size as documents are processed. Page Content Scaling is useful to allocate space to add the following:

- Header or footer
- Watermark
- Overlay

Element name	Acceptable values	Description	Contains profile parameter
ENABLED	Yes, No (Default value)	Enables or disables the PDF content scaling.	Yes doc_token_pageContentScalingEnabled
SCALEMODE	ScaleByMargin, ScaleByPercent (Default value)	Specifies the scaling value based on the margin or as a percentage value.	Yes doc_token_pageScaleMode
MARGINBOTTOM	Margin values must be defined in inches.	Specifies the margin at the bottom of the page. **Note:** Applicable only if *`MODE`* is set to `ScaleByMargin`.	Yes doc_token_pageScaleMarginBottom

Element name	Acceptable values	Description	Contains profile parameter
MARGINTOP	Margin values must be defined in inches.	Specifies the margin at the top of the page. **Note:** Applicable only if *MODE* is set to `ScaleByMargin`.	Yes doc_token_pageScaleMarginTop
MARGINLEFT	Margin values must be defined in inches.	Specifies the margin at the left side of the page. **Note:** Applicable only if *MODE* is set to `ScaleByMargin`.	Yes doc_token_pageScaleMarginLeft
MARGINRIGHT	Margin values must be defined in inches.	Specifies the margin at the right side of the page. **Note:** Applicable only if *MODE* is set to `ScaleByMargin`.	Yes doc_token_pageScaleMarginRight
SCALEHORIZONTAL	50 (Default value)	Specifies the scaling ratio for the Width only. The value is expressed as a percentage of the original page size. **Note:** Applicable only if *MODE* is set to `ScaleByPercent`.	Yes doc_token_pageScaleHorizontal

Element name	Acceptable values	Description	Contains profile parameter
SCALEVERTICAL	50 (Default value)	Specifies the scaling ratio for the Height only. The value is expressed as a percentage of the original page size. **Note:** Applicable only if *`MODE`* is set to `ScaleByPercent`.	Yes doc_token_pageScaleVertical

3.2.10.1.10.7 OPTIMIZE

This is the child of the PDFSETTINGS element and allows the creation of a web optimized PDF for a quick web view.

Element name	Acceptable values	Description	Contains profile parameter
OPTIMIZE	Yes No (Default value)	Specifies if the PDF has to be web-optimized.	Yes doc_token_optimize
CONVERTSOFTMASKACTION	Yes No (Default value)	Defines how the images with the soft mask are handled. If the image is converted to an image with stencil mask the file size is reduced, but may also reduce the image quality. Else, part of the page converted by image will be converted to JPG image and drawn on the page instead of the original image.	Yes No

3.2.10.1.10.8 OPTIMIZATIONOPTIONS

This is the child of the PDFSETTINGS element and provides the optimization options for PDF.

Element name	Acceptable values	Description	Contains profile parameter
ENABLED	Yes No (Default value)	Specifies if the PDF has to be web-optimized.	Yes doc_token_enableOptimizationOptions
SUBSETFONTS	Yes No (Default value)	Specifies if the PDF has glyphs for font characters. Every font used to display the text on a page contains a set of glyphs for the font characters. PDF specification supports “font subset”, that is, font with only those glyphs which are used. This may cause issues when text should be updated (since probably required glyphs are absent in the font subset), but for the document which is not planned to change this allows to decrease size.	Yes doc_token_enableSubsetFonts

Element name	Acceptable values	Description	Contains profile parameter
REMOVEUNUSEDSTREAMS	Yes No (Default value)	Every document page has its Resources dictionary which contains data like images, fonts and more which are used in the page contents. Resources are referenced by their names in the dictionary. In some cases, the resources may become unused, RemoveUnusedStreams finds and removes these unnecessary resources.	Yes doc_token_removeUnusedStreams
REMOVEUNUSEDOBJECTS	Yes No (Default value)	Finds orphaned objects in the PDF document and removes them, this helps decrease the document size.	Yes doc_token_removeUnusedObjects

3.2.10.1.10.9 PDFVERSION

This is the child of the PDFSETTINGS element and allows you to specify the version of the PDF.

Element name	Acceptable values	Description	Contains profile parameter
VERSION	PDFVERSION14 PDFVERSION15 (Default value) PDFVERSION16 PDFVERSION17	Specifies the PDF document version.	Yes doc_token_pdfVersion

Element name	Acceptable values	Description	Contains profile parameter
PDFCOMPLIANCE	None (Default value) PDFA-1A	Specifies the conformance level of the generated PDF. **Note:** Enable font embedding to apply this setting.	Yes doc_token_pdfCompliance

3.2.10.1.10.10 JPEGIMAGECOMPRESS

This is the child of the PDFSETTINGS element and allows the compression of JPEG images embedded in the document to reduce the final PDF file size.

Element name	Acceptable values	Description	Contains profile parameter
QUALITY	0 to 100 100 (Default value)	Specifies the quality of the JPEG image.	Yes doc_token_jpegquality

3.2.10.1.10.11 SECURITY

This is the child of the PDFSETTINGS element and allows users to add PDF security to the PDF files created.

Element name	Acceptable values	Description	Contains profile parameter
ENABLED	Yes No (Default value)	Enables or disables PDF security.	Yes doc_token_enableSecurity
ENCRYPTIONMODE	40-bit (Default value) 128-bit	Specifies the encryption level.	Yes doc_token_encryptionMode
NOCHANGINGANNOTATION	Disabled (Default value) Enabled	Controls the addition or modification of annotations. When using the 40-bit encryption, this option also allows you to fill form fields.	No

Element name	Acceptable values	Description	Contains profile parameter
NOCHANGINGDOCUMENT	Disabled (Default value) Enabled	Controls the modification of the document contents.	Yes doc_token_changesAllowed
NOCONTENTACCESSIBILITY	Disabled (Default value) Enabled	Controls the text extraction and graphics to support accessibility.	Yes doc_token_enableAccess
NODOCUMENTASSEMBLY	Disabled (Default value) Enabled	Controls the assembly of the document (for example, the insertion or deletion of pages). When using the 40-bit encryption, this option is ignored and document assembly is allowed when NOCHANGING-DOCUMENT is set to “Disabled”.	Yes doc_token_docAssembly
NOFORMFIELDFILLIN	Disabled (Default value) Enabled	Controls the filling in of forms. When using the 40-bit encryption, this option is ignored and filling in the form is allowed when NOCHANGING-DOCUMENT is set to “Disabled”.	Yes doc_token_formFieldFilling
NOPRINTING	Disabled (Default value) Enabled	Controls the printing of the document.	Yes doc_token_printing
NOSELECTIONCOPYINGCONTENT	Disabled (Default value) Enabled	Controls the copying of content.	Yes doc_token_allowCopy
OPENPASSWORD	Text	Specifies the password for opening the document.	Yes doc_token_secOpass
SECURITYPASSWORD	Text	Specifies the password for modifying the security settings.	Yes doc_token_secCpass

3.2.10.2 MSEXCELSETTINGS

This is the child of the SETTINGS element and contains customizations applicable to the source documents that are in the Microsoft Excel format.

EXCELPDFSETTINGS

This is the child of the MSEXCELSETTINGS element and provides customization options that must be specified to the PDF output documents.

Element name	Acceptable values	Description	Contains profile parameter
PRINTWHAT	ActiveSheets EntireWorkbook (Default value)	Enables or disables printing selected sheets of the entire workbook to PDF.	Yes doc_token_printWhat
ENABLERESAMPLE	Yes No (Default value)	Enables or disables resampling of images within the Microsoft Excel sheets.	Yes doc_token_enableResample
RESAMPLEDPI	0-600	Specifies the resolution in ppi. Images that are greater than the specified ppi are resampled.	Yes doc_token_resolution
RESAMPLEJPEGQUALITY	0-100	All images are converted to JPEG with the specified quality setting.	Yes doc_token_jpegquality
AUTOADJUSTROWHEIGHT	YES NO (default)	Enables or disables the hidden lines in Excel to PDF.	YES doc_token_autoAdjustRowHeight
ENABLEEMFRENDERING	YES NO (default)	Enables or disables the meta files embedded in excel workbooks to PDF.	YES doc_token_emfRendering

Element name	Acceptable values	Description	Contains profile parameter
ESTIMATEDMAXPAGECOUNT	0-2,127,483,647	To prevent Transformation Services from running out of memory whenever a large excel file is queued for transformation. If the output of the document tends to have more pages than the `estimatedMaxPageCount`, then the transformation is stopped and the next queue item is processed.	YES doc_token_estimatedMaxPageCount

EXCELBOOKMARKS

This is the child of the MSEXCELSETTINGS element and controls generation of Microsoft Excel bookmarks.

Element name	Acceptable values	Description	Contains profile parameter
ENABLED	Yes No (Default value)	Enables or disables bookmark creation.	Yes doc_token_enableBookMarks

3.2.10.3 MSPPTSETTINGS

This is the child of the SETTINGS element and contains customizations that are applicable to source documents that are in the Microsoft PowerPoint format.

PPTPDFSETTINGS

This is the child of the MSPPTSETTINGS element and provides customization options to be specified to the PDF output documents.

Element name	Acceptable values	Description	Contains profile parameter
ENABLED	Yes No (Default value)	Enables or disables PDF settings.	Yes doc_token_enablePdfSettings

Element name	Acceptable values	Description	Contains profile parameter
RESOLUTION	0-600	Specifies the resolution of images inside the PDF document.	Yes doc_token_resolution
QUALITY	0-100	All images are converted to JPEG with the specified quality setting.	Yes doc_token_jpegquality

3.2.10.4 HTMLSETTINGS

This is the child of the SETTINGS element.

You can make changes to profiles like Legacy_to_Pdf or any other profiles that require changes related to margins and other HTML settings.

Element name	Acceptable values	Description	Contains profile parameter
MARGIN	0,0,0,0 (Default value)	It specifies the margin of the page. The given values are in the order Top, Left, Bottom, and Right for the respective margins.	Yes doc_token_htmlToPDFMargins

3.2.10.5 METADATAELEMENTS

This element is the child element of the SETTINGS element and contains customizations that will be applicable to source documents of the Microsoft Word format.

METADATAELEMENT

This is the child of METADATAELEMENTS element and provides the ability to set metadata by using parameters on the PDF output documents.

Element name	Acceptable values	Description	Contains profile parameter
FIELD	Title	Specifies the Title value.	No
	Subject	Specifies the Subject value.	
	Author	Specifies the Author value.	

Element name	Acceptable values	Description	Contains profile parameter
	Keywords (Comma Separated)	Specifies the Keywords value(s).	
	User Defined	Specifies user defined metadata. User defined metadata field names cannot contain spaces.	
VALUE		Specifies the value to be stored in the metadata.	Yes doc_token_title doc_token_subject doc_token_author doc_token_keywords

3.2.11 Generating a specific number of storyboard renditions

If you want Transformation Services - Documents to create only first n storyboards of any document, then do the following:

1. Check out `storyboard_pdfstoryboard.xml` from the repository in the `/System/Media Server/command line Files/` folder.
2. Change the following line in the file:

   ```
   <PROP name="Max Pages" type="unsigned long" token="doc_token_maxPages">-1</PROP>
   ```

   to

   ```
   <PROP name="Max Pages" type="unsigned long" token="doc_token_maxPages">n</PROP>
   ```

   where n is the number of storyboards.
3. Check in `storyboard_pdfstoryboard.xml`.
4. Restart the Transformation Services.

If you want Transformation Services - Documents to create only first n storyboards of any document, but with each storyboard equally spaced, then do the following:

1. Check out `storyboard_pdfstoryboard.xml` from the repository in the `/System/Media Server/command line Files/` folder.
2. Change the following line:

   ```
   <PROP name="Max Pages" type="unsigned long" token="doc_token_frames_requested">

   -1</PROP>
   ```

   to

   ```
   <PROP name="Max Pages" type="unsigned long" token="doc_token_frames_requested">
   ```

```
n</PROP>
```

where n is the number of storyboards.

3. Check in `storyboard_pdfstoryboard.xml`.
4. Restart the Transformation Services.

Notes

- The doc_token_frames_requested parameter gets preference over the doc_token_maxPages parameter, if both parameters are set.
- These settings are applicable only for PDFStoryboard plug-in.
- These settings are not applicable when Transformation Services - Documents is deployed in an xCP environment and xCP DARs are installed on the repository.

3.2.12 Adding a signature page

addSignature_adts is a user profile, which allows users to add a signature page to Microsoft Office or PDF documents. This profile converts the provided document to PDF and adds a pre-defined signature page to the output. This is specially useful if a company wants to append its logo to specific documents or other legal regulations, usage limitations, company info, disclaimers, and so on.

To specify a suitable signature document, perform the following steps after installation:

1. Login as an administrator.
2. Using Documentum Administrator, navigate to the following file path : `/System/Applications/Signature/`.
3. Check out the blank `signature_template.pdf`.
4. Replace the blank `signature_template.pdf` with your custom PDF signature file of the same name.
5. Set the world permission to “READ”.
6. Check in the custom PDF.

You can append this custom PDF to documents that need to be generated.

3.2.13 Creating PDF renditions and enabling thumbnail and storyboard renditions on import

To automatically create PDF renditions when importing content, first enable the particular format for rich media, and then edit the `register_*.xml` file. “Register Profile” on page 144 provides more information.

Transformation Services can also create thumbnail and storyboard renditions when importing content.

3.2.13.1 Rich media enabling formats

This procedure is essential for proper processing of imported content. If formats are not enabled for rich media, the Documentum CM Server will not pass them to the Transformation Services queue.

To enable formats for rich media:

- Run the following DQL statement:

```
update dm_format object set richmedia_enabled = 1 where name
in ('<format name>')
```

  For example, to create thumbnails and storyboards for Microsoft Excel and Word formats, run the following DQL statement:

```
update dm_format object set richmedia_enabled = 1 where
 name in ('excel8book','msw8')
```

This table lists some examples of the format mapping used in Documentum systems. Use these format names in DQL.

Table 3-7: Documentum format mappings for Transformation Services

Description	Documentum name (Office 2003)	Documentum name (Office 2007)	Documentum name (Office 2010)	Documentum name (Office 2013)
Microsoft Excel (XLS, XLSX)	excel8book	excel12book	excel14book	excel15book
Microsoft Excel (XLT)	excel8template	excel12template	excel14template	excel15template
Microsoft PowerPoint (POT)	ppt8_template	ppt12template	ppt14template	ppt15template
Microsoft PowerPoint (PPT)	ppt8	ppt12	ppt14	ppt15
Microsoft Project (MPP)	msproject	msproject	msproject	msproject

Description	Documentum name (Office 2003)	Documentum name (Office 2007)	Documentum name (Office 2010)	Documentum name (Office 2013)
Microsoft Word (DOC, DOCX)	msw8	msw12	msw14	msw15
Microsoft Word (DOT)	msw8template	msw12template	msw14template	msw15template

3.2.13.2 Creating a PDF rendition on import

After enabling the formats for rich media, modify the Register Profile to activate the PDF rendition on import.

To activate the PDF rendition on import:

1. In the repository, browse to `System/Media Server/System Profiles/` and check out the Register Profile.

2. Open the Register Profile in an XML or text editor.

3. Change the following tag:

   ```
   <Format source="<sample>" target="<sample>"/>
   ```

   to

   ```
   <Format source="<Add source format here>" target="<Add target format here>"/>
   ```

   For example, to enable PDF renditions for Microsoft Word documents, the tag looks as follows:

   ```
   <Format source="msw8" target="msw8"/>
   ```

   Note: Source and target formats are the same in the register profile. This is because processing does not start from this profile; the source is passed to the next profile.

4. Add the following tags to the `ProfileSequence` section of the relevant `register_xxxx.xml`, if it is not already included:

   ```
   <InnerProfile path="/System/Media Server/System Profiles/
   document_to_pdf" waitOnCompletion="true" useTargetFormat="true">
   <InnerTokenMapping LocalProfileToken="pdf" InnerProfileToken=
   "doc_token_targetFormat" Literal="true"/>
   <InnerTokenMapping LocalProfileToken="PDFVersion15"
   InnerProfileToken="doc_token_pdfVersion" Literal="true"/>
   <InnerTokenMapping LocalProfileToken="Automatic" InnerProfileToken
   ="doc_token_usePrinterMetrics" Literal="true"/>
   <InnerTokenMapping LocalProfileToken="600" InnerProfileToken=
   "doc_token_resolution" Literal="true"/>
   <InnerTokenMapping LocalProfileToken="Yes" InnerProfileToken=
   "doc_token_optimize" Literal="true"/>
   <InnerTokenMapping LocalProfileToken="Yes" InnerProfileToken=
   "doc_token_enableBookMarks" Literal="true"/>
   <InnerTokenMapping LocalProfileToken="DocumentContent"
   InnerProfileToken="doc_token_PrintType" Literal="true"/>
   <InnerTokenMapping LocalProfileToken="false" InnerProfileToken
   ="overwrite_rendition" Literal="true"/>
   ```

```
<InnerTokenMapping LocalProfileToken="legacy" InnerProfileToken=
"transformation_type" Literal="true"/>
<!security settings>
<InnerTokenMapping LocalProfileToken="No" InnerProfileToken=
"doc_token_enableSecurity" Literal="true"/>
<InnerTokenMapping LocalProfileToken="40bit" InnerProfileToken=
"doc_token_encryptionMode" Literal="true"/>
<InnerTokenMapping LocalProfileToken="Disabled" InnerProfileToken=
"doc_token_changesAllowed" Literal="true"/>
<InnerTokenMapping LocalProfileToken="Disabled" InnerProfileToken=
"doc_token_enableAccess" Literal="true"/>
<InnerTokenMapping LocalProfileToken="Disabled" InnerProfileToken=
"doc_token_docAssembly" Literal="true"/>
<InnerTokenMapping LocalProfileToken="Disabled" InnerProfileToken=
"doc_token_formFieldFilling" Literal="true"/>
<InnerTokenMapping LocalProfileToken="Disabled" InnerProfileToken=
"doc_token_printing" Literal="true"/>
<InnerTokenMapping LocalProfileToken="Disabled" InnerProfileToken=
"doc_token_allowCopy" Literal="true"/>
<InnerTokenMapping LocalProfileToken=" " InnerProfileToken=
"doc_token_secOpass" Literal="true"/>
<InnerTokenMapping LocalProfileToken=" " InnerProfileToken=
"doc_token_secCpass" Literal="true"/>
</InnerProfile>
```

5. Save the file.

6. Check the file into the repository.

7. Restart the Transformation Services.

3.2.13.3 Registering formats to enable thumbnails and storyboards

After rich-media enabling the formats, modify the system profiles, `register.xml` and `register_legacy.xml` to activate the thumbnail and storyboard rendition creation on import of Microsoft Office and image formats.

Note: This procedure is applicable only if `register_legacy.xml` is targeted in `register.xml` profile.

To register the formats:

1. In the repository, navigate to `System/Media Server/System Profiles/` and check out the following files: `register.xml` and `register_legacy.xml`.

2. Open `register.xml` and `register_legacy.xml` in an XML or text editor.

3. Change or add the following tag in both the `register.xml` and `register_legacy.xml` files:

   ```
   <Format source="<sample>" target="<sample>"/>
   ```

   to

   ```
   <Format source="<Add source format here>" target="<Add target format here>"/>
   ```

   For example, to enable thumbnails and storyboards for Microsoft Word documents, your tag looks as follows:

   ```
   <Format source="msw8" target="msw8"/>
   ```

   To enable thumbnails and storyboards for image formats, your tag looks as follows:

```
<Format source="gif" target="gif"/>
```

Note: Source and target formats are the same in the register profile. This is because processing does not start from this profile; the source is passed to the next profile.

4. Navigate to `System/Media Server/System Profiles` and check out the `document_registration.xml` file.

5. Add the following image formats, if you require thumbnails for images: jpeg, gif, png, tiff, photoshop6, photoshop7, photoshop8, cr2, crw, wmf, and bmp.

   For example:

   ```
   <Format source="gif" target="pdf"/>
   ```

   For Microsoft Office formats, ensure that the `document_registration.xml` file contains tags similar to the following:

   ```
   Format source="msw8" target="pdf"/>
   ```

6. Add the following tags to the `ProfileSequence` section of the `register_legacy.xml` file:

   ```
   <!--This extracts properties for office formats. Add if required-->
    <InnerProfile path="/System/Media Server/System Profiles/
   document_extract_props" waitOnCompletion="true" useTargetFormat=
   "true">
   </InnerProfile>
   <!--This will generate thumbnail, proxy and storyboards for
   all document & image formats-->
   <InnerProfile path="/System/Media Server/System Profiles/
   document_registration" waitOnCompletion="true" useTargetFormat="true">
   <InnerTokenMapping LocalProfileToken="pdf" InnerProfileToken=
   "doc_token_targetFormat" Literal="true"/>
   </InnerProfile>
   ```

7. Save the files.

8. Check the files back into the repository.

9. Restart the Transformation Services.

3.2.14 Customizing printing and rendition output

Various printing and rendition processes can be configured in the legacy profiles. By default, these processes are disabled. You can modify the profiles to:

- Save the PDF text (PDT) file
- Store PDF metadata
- Configure legacy transformations to replace previous renditions, instead of storing multiple renditions

These configuration procedures are described in this section.

3.2.14.1 Saving PDF Text files

The PDF Text rendition of a document is the text file used specifically for term-hit highlighting. By default, PDF Text files are not saved to the repository. To store the PDF Text rendition of a document in the repository, modify the legacy_to_pdf profile.

To save PDF Text renditions in the repository:

1. Log in to the Transformation Services configured repository as an administrator user.

2. Navigate to System\Media Server\System Profiles\.

3. Check out the `legacy_to_pdf` profile.

   Note: If this file is not immediately visible, choose **Show All Objects and Versions** from the Show list box.

4. Open the `legacy_to_pdf` file from the checkout folder on the local machine.

5. Uncomment the following section:

```
<!-- uncomment the below section to enable the storing of pdf text files -->
<!--
<InnerProfile path="/System/Media Server/System Profiles/
transformToText" waitOnCompletion="false" useTargetFormat="true">
         <InnerTokenMapping LocalProfileToken="pdftext" InnerProfileToken=
"target_format" Literal="true"/>
         <InnerTokenMapping LocalProfileToken="pdf" InnerProfileToken=
"source_format" Literal="true"/>
         <InnerTokenMapping LocalProfileToken="True" InnerProfileToken=
"doc_token_storePDTRendition" Literal="true"/>
         <InnerTokenMapping LocalProfileToken="1" InnerProfileToken=
"target_page_modifier" Literal="true"/>
<InnerTokenMapping LocalProfileToken="false" InnerProfileToken=
"overwrite_rendition" Literal="true"/>
<InnerTokenMapping LocalProfileToken="legacy" InnerProfileToken=
"transformation_type" Literal="true"/>
</InnerProfile>
-->
```

6. Save and close the file.

7. Check in the modified profile to the repository.

PDF text renditions of documents are now stored in the repository.

3.2.14.2 Storing PDF metadata

The storing of PDF metadata is also configured in the `legacy_to_pdf` profile. By default, PDF metadata is not stored.

To store PDF metadata in the repository:

1. Log in to the Transformation Services configured repository as an administrator user.
2. Navigate to System\Media Server\System Profiles\.
3. Check out the `legacy_to_pdf` profile.

   Note: If this file is not immediately visible, choose **Show All Objects and Versions** from the Show list box.

4. Open the `legacy_to_pdf` file from the checkout folder on the local machine.
5. Uncomment the following section:

```
<!-- uncomment the below section to enable the storing of pdf metadata-->
<!--
      <InnerProfile path="/System/Media Server/System Profiles/
embedMetadataToContent_adts" waitOnCompletion="false" useTargetFormat="true">
         <InnerTokenMapping LocalProfileToken="pdf" InnerProfileToken=
"target_format" Literal="true"/>
         <InnerTokenMapping LocalProfileToken="pdf" InnerProfileToken=
"source_format" Literal="true"/>
         <InnerTokenMapping LocalProfileToken="True" InnerProfileToken=
"doc_token_embedMetadata" Literal="true"/>
         <InnerTokenMapping LocalProfileToken="embed_metadata"
InnerProfileToken="target_page_modifier" Literal="true"/>
         <InnerTokenMapping LocalProfileToken="false" InnerProfileToken=
"overwrite_rendition" Literal="true"/
<InnerTokenMapping LocalProfileToken="legacy" InnerProfileToken=
"transformation_type" Literal="true"/>
</InnerProfile>
-->
```

6. Save and close the file.
7. Check in the modified profile to the repository.

PDF metadata of documents is now stored in the repository.

3.2.14.3 Configuring for rendition replacement

By default, Transformation Services allows for multiple PDF and HTML renditions from the same source file. In other words, each time a user transforms a file, an additional rendition is created and stored. If you would prefer that subsequent renditions *replace* previous renditions of a source document, this can be configured using the following procedure.

To overwrite existing renditions when transforming files:

1. Log in to an applicable application (such as Digital Asset Manager) as an administrator user.

2. Navigate to System\Media Server\System Profiles.
3. Check out the following profiles:
   - `legacy_to_html`
   - `legacy_to_pdf`

   Note: If these files are not immediately visible, choose **Show All Objects and Versions** from the Show list box.
4. Open the `legacy_to_html.xml` file from the checkout folder on the local machine.
5. Locate the following tag within the `InnerProfile` section:

   ```
   <InnerTokenMapping LocalProfileToken="false"
   InnerProfileToken="overwrite_rendition" Literal="true"/>
   ```

   and change it to:

   ```
   <InnerTokenMapping LocalProfileToken="true"
   InnerProfileToken="overwrite_rendition" Literal="true"/>
   ```
6. Save and close the file.
7. Open the `legacy_to_html.xml` file from the checkout folder on the local machine.
8. Repeat step 5.
9. Save and close the file.
10. Check in the two files.

Transformation Services will now store only one PDF or HTML rendition for each source file.

To replicate this behavior when the **Reset Renditions** option is applied, modify the `register.xml` file in the same way.

3.2.15 Configuring change tracking

By default, Transformation Services does not pass change tracking to output PDFs. However, change tracking can be enabled by configuring the `legacy_to_pdf` profile.

To enable change tracking:

1. In the repository, browse to `System/Media Server/System Profiles` and check out `legacy_to_pdf.xml`.
2. Open the file in a text editor.
3. Change this line:

   ```
   <InnerTokenMapping LocalProfileToken="DocumentContent"
   InnerProfileToken="doc_token_PrintType" Literal="true"/>
   ```

to:

```
<InnerTokenMapping LocalProfileToken="DocumentWithMarkup"
InnerProfileToken="doc_token_PrintType" Literal="true"/>
```

4. Save the file.

5. Check in the file to the repository.

3.2.16 Transferring document attributes

A document’s attributes are its descriptive characteristics, such as name, title, author, creation date, object ID, or subject. You can set the values for some attributes. Other attributes, such as the document’s creation date, are set by the system.

Transformation Services lets you transfer these attributes to a document’s PDF rendition (but not to a document’s HTML rendition). Use the attributes for identification or tracking purposes after the PDF document is exported from the repository where the source document resides.

Note: Some attributes, such as repository ID, may not be visible when viewed through Document Info in Adobe Exchange.

You can also transfer attribute information to custom plug-ins, where the plug-in needs to know certain information about a document (such as type, author, or source type).

For example, write a plug-in that looks for certain attributes identifying a document as a Standard Operating Procedure (SOP). If those attributes are found, the plug-in prints a special cover page, or adds or subtracts context-sensitive information before the document is rendered into PostScript format.

3.2.16.1 Transferring document attributes to PDF and embedding metadata

To transfer attributes, edit the appropriate command line files of Transformation Services.

The following command line files are responsible for transferring the attributes from the source document to the PDF rendition:

- `embedMetadataToContent_adts.xml` (used by the transformation profiles for embedding the metadata)

Define the attribute mapping for PDF renditions under the `PDF_DOC_INFO` tag of these command line files. The syntax is:

```
value = object_alias.attribute[repeating_spec]
```

An example would appear as follows:

```
<PDF_DOC_INFO>
...
...
```

```
<MetadataMapper name="DCTMobjectTitle" value="title"
    token="doc_metadata_dctmTitle"/>
...
...
</PDF_DOC_INFO>
```

The attribute token is used internally by the server code, and has to be named with the prefix doc_metadata_ . For example:

```
token=" doc_metadata_dctmTitle”
```

Table 3-8: Attribute mapping arguments

Argument	Description
key	*Key* can be an existing attribute or a user-defined attribute. Examples: author, subject, title, producer.
object_alias	When omitted or specified as o, the *object_alias* indicates that the following attribute is that of a sysobject or one of its subtypes (the document being rendered). Other valid object_aliases are: • d (repository config object) • s (server config object) • q (attributes from special runtime-computed items) *OpenText Documentum Content Management - Server Administration and Configuration Guide (EDCCS250400-AGD)* provides more information about configuration objects.
attribute	In the case of o, s, and d object aliases, *attribute* dynamically maps the attribute. In the case of the q object, the only currently available attributes are sentby, login_ticket, and username.
value	For repeating attributes, *value* defines the repeating attribute separator (for example, a comma or a slash). If no separator is defined, and the attribute is repeating, Documentum CM Server maps only the first attribute in the series [element 0].

Here are some examples of attribute mapping syntax.

The first example shows an attribute where the object alias (o) is implied:

```
<MetadataMapper name="Subject" value="subject"
    token="doc_metadata_subject"/>
```

The following example shows an attribute mapped with specified object alias:

```
<MetadataMapper name="DCTMSourceContentType" value="o.a_content_type"
    token="doc_token_sourceContentType"/>
```

The following example shows a comma-separated list of keywords from sysobject:

```
<MetadataMapper name="Keywords" value="keywords[,]"
    token="doc_metadata_keywords"/>
```

The following example shows a list of authors, separated by // (two forward slashes):

```
<MetadataMapper name="Author" value="authors[//]"
    token="doc_metadata_author"/>
```

Note: Restart the Transformation Services for the changes in the command line files to take effect.

3.2.16.2 Viewing the attributes in the PDF rendition

To see the transferred attributes for a PDF rendition, choose Document Properties in Adobe Acrobat.

3.2.17 Configuring HTML renditions in fluid layout

The HTML renditions of a Microsoft Word file by default are in fixed layout. To transform Microsoft Word file to HTML file with fluid layout, perform the following:

1. Log in to an applicable application (such as Digital Asset Manager) as an administrator user.
2. Open the `System\Media Server\System Profiles` folder.
3. Check out the `legacy_to_html` profile.

   Note: If this file is not immediately visible, choose **Show All Objects and Versions** from the Show list box.

4. Open the `legacy_to_html.xml` file from the checkout folder on the local machine.
5. Set *`fluid_word_to_html_layout`* to true.
6. Save and close the file.
7. Check in the file.

3.3 Configuration tasks specific to Transformation Services - Media

This section contains procedures for the image-based configuration tasks that are performed with Transformation Services - Media.

3.3.1 Configuring the PowerPoint2 plug-in

The Microsoft Office-based PowerPoint2 plug-in has been reintroduced in this release of Transformation Services because of better stability and reduced data loss, which was seen in PowerPoint transformations in Transformation Services 6.7.x products.

The PowerPoint1 and PowerPoint2 plug-ins are bundled along with Transformation Services - Media. Only the PowerPoint1 plug-in is turned on by default. To use the PowerPoint2 plug-in, add the following in the `CTSPluginService.xml` file:

```
<CTSPlugin DELEGATE_CLASS="com.documentum.cts.plugin.powerpoint2.PowerPointPlugin"
CONFIGFILE="%Documentum Content Transformation Services_HOME%
\PowerPoint2\PowerPoint2.xml"/>
```

The configuration files for the PowerPoint1 and PowerPoint2 plug-ins are located in the *`<Documentum Content Transformation Services_HOME>`*`\config\powerpoint1\` and *`<Documentum Content Transformation Services_HOME>`*`\config\powerpoint2` folders on the Transformation Services - Media server respectively.

All the OOTB Transformation Services - Media profiles and system profiles will refer to the PowerPoint1 plug-in only. Hence, PowerPoint registration, assembly, conversions, slide generation, property extraction and storyboarding is handled by the PowerPoint1 plug-in OOTB. To use the PowerPoint2 plug-in instead, do the following:

1. Check out the corresponding profile(s) or system profile(s).
2. Change the mptype attribute of CommandFilePath element (for example, extraction).
   Change

   ```
   <CommandFilePath mptype="POWERPOINT1">
                       /System/Media Server/command line Files/ppt_extract_props.xml
   </CommandFilePath>
   ```

   To

   ```
   <CommandFilePath mptype="POWERPOINT2">
                       /System/Media Server/command line Files/ppt_extract_props.xml
   </CommandFilePath>
   ```

3. Check in the profile in to the repository.

Note: The name and content of the command line files are the same for the PowerPoint1 and PowerPoint2 plug-ins. Hence, changing the plug-in name in profiles or system profiles is adequate.

The following OOTB profiles or system profiles can be switched to use either of the two plug-ins:

- ppt_conversion
- powerpointDefaultStoryboard
- powerpointRegistration
- powerpointRegistrationStoryboard
- powerpointStoryboard
- ppt_assembler
- ppt_extract_props
- ppt_slide_generator
- thumbnail
- thumbnail_ppt
- autoGenPreviewProxy
- autoGenProxy

Note: For the PowerPoint2 plug-in to work, ensure that you have installed Microsoft PowerPoint 2007 SP3 (32-bit) on the Transformation Services - Media server that is installed on Windows Server 2008 R2 SP1 (64-bit) operating system.

Make the following changes in the `log4j2.properties` file (located at *`<Documentum Content Transformation Services_HOME>`*`\config\log4j2.properties`) to enable logging for the PowerPoint2 plug-in:

```
log4j.category.com.documentum.cts.plugin.powerpoint2=DEBUG, Powerpoint2Appender
log4j.appender.Powerpoint2Appender=org.apache.log4j.DailyRollingFileAppender
log4j.appender.Powerpoint2Appender.File=<Documentum Content Transformation Services_HOME>
\\logs\\Powerpoint2_log.txt
log4j.appender.Powerpoint2Appender.Append=true
log4j.appender.Powerpoint2Appender.layout=org.apache.log4j.PatternLayout
log4j.appender.Powerpoint2Appender.layout.ConversionPattern=%d{HH\:mm\:ss,SSS}
ndi %10r %5p [%10t] %-20c{1} - %5x %m%n
log4j.appender.Powerpoint2Appender.DatePattern='.'yyyy-ww-dd
```

After you make these changes, restart the Transformation Services.

3.3.2 Managing PowerPoint registration

Due to complex processing during PowerPoint registration, the total time taken to complete registration can be substantial depending on the number of slides in the presentation. This prevents other PowerPoint registrations from being completed. To mitigate this, the PowerPoint registration process is split into two steps: Image Rendition generation and Slide generation. Among these two processes, the Slide generation process takes the longest time. To split this registration process, you must have at least two Documentum Content Transformation Services - Media machines configured to the same repository.

PowerPoint Registration —> Primary Documentum Content Transformation Services - Media (Image Rendition generation) —> Delegate to —> Secondary Documentum Content Transformation Services - Media (Slide generation)

The Primary Documentum Content Transformation Services - Media machine generates image renditions and delegates slide generation to the secondary Documentum Content Transformation Services - Media machine. The secondary Documentum Content Transformation Services - Media machine generates all the slides and uploads them to the repository. Following are the changes that must be made to the `powerpoint1.xml` to enable the process:

```
<PPT_IMPORT_PROCESSING MODE="STORYBOARDS_AND_SLIDES | STORYBOARDS _ONLY |
SLIDES_ONLY" IMAGE_RENDITION_PROPERTIES="FALSE">
 <!-- XML below is read only by STORYBOARDS_ONLY mode MTS -->
  <CTS_DELEGATE_QUEUE QUEUE_NAME="dm_mediaserver_ppt_a1">
   <MIN_SLIDES_COUNT>10</MIN_SLIDES_COUNT>
  </CTS_DELEGATE_QUEUE>
 <CTS_DELEGATE_QUEUE QUEUE_NAME="dm_mediaserver_ppt_a2">
  <MIN_SLIDES_COUNT>10</MIN_SLIDES_COUNT>
 </CTS_DELEGATE_QUEUE>
 <CTS_DELEGATE_QUEUE QUEUE_NAME="dm_mediaserver_ppt_b">
  <MIN_SLIDES_COUNT>100</MIN_SLIDES_COUNT>
 </CTS_DELEGATE_QUEUE>
</PPT_IMPORT_PROCESSING>

....
.....
</PREPROCESS_SLIDE_RENDITIONS>
```

Note: This is only an example and can be customized as required.

The MODE parameter must contain one of the following values:

- STORYBOARDS_AND_SLIDES – By default the attribute value 'STORYBOARDS_AND_SLIDES' means that primary Transformation Services - Media processes the PowerPoint registration completely, without delegating slide generation to a secondary Transformation Services - Media instance. This means that the Transformation Services - Media behaves as a regular Transformation Services - Media.
- STORYBOARDS_ONLY – The Transformation Services - Media that is configured to split the PowerPoint registration must have this value. This mode generates image renditions and delegates slide generation to a secondary Transformation Services - Media instance. If the minimum slide size of the

secondary Transformation Services - Media is more than the current PowerPoint file being processed, then the PowerPoint asset will be processed fully by Transformation Services - Media. In the example, any slide deck between sizes 1 to 9 will be processed fully by the primary Transformation Services - Media itself without being delegated to the secondary Transformation Services - Media.

- SLIDES_ONLY – All the secondary Transformation Services - Media that handle slide generation must have this value. This mode generates all the slides and uploads them to the repository.

IMAGE_RENDITION_PROPERTIES turns off updating all the slide image renditions with height, width, and format values. This saves considerable processing time. It is recommended to always retain the value as “False”.

The <*CTS_DELEGATE_QUEUE*> elements list the secondary Transformation Services - Media that can be configured with the specific queue names and slide sizes that it processes. In the sample, three secondary Transformation Services - Media instances are configured to handle PowerPoint files of different sizes. The QUEUE_NAME should be unique and each queue should have a specific secondary Transformation Services - Media configured against it.

In this example, only slides with deck size 10 or higher are delegated to the secondary Transformation Services - Media and smaller sized PowerPoint files (1-9 slides) are processed by the Transformation Services - Media that is marked as STORYBOARDS_ONLY (that is, the primary Transformation Services - Media). If you need all the slide generation to be delegated to the secondary Transformation Services - Media then at least one Transformation Services - Media instance should be marked with MIN_SLIDES_COUNT = 1.

Following are some sample load balancing scenarios based on the input slide sizes and sample configuration:

Slide Size	10	20	21	9	125	155	300
Delegated to Transformation Services - Media	a1	a2	a1	Primary Transformation Services - Media instance	b	b	b

Slide Size	125	5	75	146	65	210	400
Delegated to Transformation Services - Media	b	Primary Transformation Services - Media instance	a1	b	a2	b	b

Different secondary Transformation Services - Media that have the same MIN_SLIDES_COUNT are delegated based in a round robin manner to distribute the load evenly. The queue names must be unique to ensure that registration tasks are picked up by specific Transformation Services - Media configured to those PowerPoint sizes.

Note: To disable the PowerPoint file split processing, specify MODE="STORYBOARDS_AND_SLIDES".

3.3.2.1 Configuring a primary Transformation Services - Media instance

1. Specify the attribute as MODE="STORYBOARDS_ONLY" to enable PowerPoint storyboard processing and splitting the process to the secondary Transformation Services - Media.
2. List the secondary Transformation Services - Media under the CTS_DELEGATE_QUEUE nodes.
3. Create all users specified in the attribute QUEUE_NAME (for example, dm_mediaserver_ppt_a1) with the same values as the default user "dm_mediaserver" using Documentum Administrator. If this is not done, queue generation will fail with an exception.
4. Retain IMAGE_RENDITION_PROPERTIES="FALSE" to improve the processing time as rendition image attributes of width and height are not updated, thus saving considerable time.

3.3.2.2 Configuring a secondary Transformation Services - Media instance

Note: This process requires one Transformation Services - Media for each CTS_DELEGATE_QUEUE node. The QUEUE_NAME should be unique to each secondary Transformation Services - Media instance.

1. Specify the attribute as MODE="SLIDES_ONLY".
2. Retain IMAGE_RENDITION_PROPERTIES="FALSE" to improve the processing time as rendition image attributes of width and height are not updated, thus saving considerable time.
3. Ensure that you modify the `CTSServerService.xml` file with the queue name configured in the CTS_DELEGATE_QUEUE node.

   For example: Replace dm_mediaserver with dm_mediaserver_ppt_a1

   ```
   <QueueProcessorContext>
   <CTSServer AttributeName="queueItemName" AttributeValue="dm_mediaserver_ppt_a1"/>
   ```

   This indicates that this Transformation Services - Media would not process any other Transformation Services jobs but would remain dedicated to processing PowerPoint slides.

   Remove <*QueueProcessorContext*> for QueueItemName 'dm_autorender_win31'.

Note: Restart the Transformation Services to implement any changes in the XML files.

3.3.3 Selecting a colorspace

To choose a specific colorspace, edit the `register_legacy.xml` system profile and add the following inner token to all profiles that generate thumbnails such as thumbnail, autoGenProxy, autoGenPreviewProxy, and so on.

```
<!-- if RGB output images are needed -->
<InnerTokenMapping Literal="true" InnerProfileToken="doc_token_colorSpace"
LocalProfileToken="RGB"/>
```

or

```
<!-- if CMYK output images are needed -->
<InnerTokenMapping Literal="true" InnerProfileToken="doc_token_colorSpace"
LocalProfileToken="CMYK"/>
```

This will ensure that the output image corresponds to the specified colorspace.

3.3.4 Generating the background color for thumbnails from transparent PDF files

When you open some PDFs in Acrobat Reader, they may appear as a white page. However, on enabling the “Page Display / Show transparency grid” option in the Preferences menu, you can view the content. Out-of-the-box, the Transformation Services - Media server replaces the transparent layers with white. Hence, you must configure this to another color (for example, light gray) so that the images can be seen.

To generate the preferred background color for PDF files, do the following:

This solution may not work for all types of PDF files and is a third party limitation.

1. Checkout the `thumbnail_pdfstoryboard.xml` system profile and ensure that Image3 is the first plug-in in the list.

2. Checkout the `thumbnail_pdf_imw.xml` command line (heavyweight) profile and modify the following:

   Add:

   ```
   <PROP name="bgColor" type="string" token="doc_token_bgColor">#FFFFFF</PROP>
   ```

   Change:

   ```
   <Options>-limit memory
   ```

   To:

   ```
   <Options>-colorspace doc_token_colorSpace doc_token_sourceFile -limit memory
   ```

   Change:

   ```
   -quality 80
   ```

   To:

   ```
   -flatten -quality 80
   ```

Here, #FFFFFF is white, which can be changed to any color of choice. Navigate to the Imagemagick website to check the color options available.

If PDF transparency is required in the Preview proxy images, then make similar changes to the following files:

- `System Profiles/autoGenPreviewProxy_pdfstoryboard/` and Command Line `Files/autoGenProxy_pdf_imw.xml`
- `System Profiles/autoGenProxy_pdfstoryboard/` and Command Line `Files/autoGenProxy_pdf_imw.xml`

If PDF transparency is required in PDF storyboards for the background color of choice, do the following:

1. Checkout the `storyboard_pdfstoryboard.xml` system profile and ensure that Image3 is the first plug-in in the list.

2. Checkout the `storyboard_pdf_imw.xml` command line (heavyweight) profile and modify the following:

   Change:

   ```
   <Options>-limit memory
   ```

   To:

   <Options>-colorspace doc_token_colorSpace doc_token_sourceFile -limit memory

   Change:

   ```
   -quality 80
   ```

   To:

   ```
   -bordercolor #FFFFFF -border 0 -alpha off -quality 80
   ```

   Here, #FFFFFF is white, which can be changed to any color of choice. Navigate to the Imagemagick website to check the color options available.

3.3.5 Improving the quality of JPEG image files converted from SVG format

The quality of the SVG file converted to JPEG using the resize_preserveRatio profile is 80% by default. To increase the quality to 100%, update the value of “doc_token_quality” in the TransformTo and Resize profiles as follows:

```
InnerTokenMapping LocalProfileToken="100" InnerProfileToken=
"doc_token_quality" literal="true"
```

Note: The jpeg_preview can be changed to filestore for storage.

3.3.6 Configuring the Image3 plug-in

The Image3 plug-in has some extra options other than the standard configuration options for plug-ins that can be configured through the plug-in’s configuration file. This section provides details about these options and how they can be configured.

3.3.6.1 ICC profiles

The ICC profiles section of the configuration file holds a list of individual ICC profiles. Specify one ICC profile per colorspace. An ICC profile is specified in an ICC_PROFILE_INFO XML tag:

```
<ICC_PROFILE_INFO profile_token="doc_token_cmyk_profile" colorspace="CMYK">
C:\Documentum\CTS\Image3\required\color\USWebCoatedSWOP.icc
</ICC_PROFILE_INFO>
```

The value of the XML element is the fully qualified path to the ICC profile. The attributes are as follows:

- profile_token – the token in the profile that will be substituted with the path of the ICC profile.
- colorspace – the colorspace of this profile.

Two ICC profiles will be used in conjunction to perform color conversions when doing transformations.

The Image3 plug-in is configured out-of-the-box with a CMYK and RGB profile, which will be used to convert CMYK images to RGB when creating thumbnails and low-resolution JPEGs.

Note: ICC version 4 profiles are not supported.

3.3.6.2 COM Server parameters

There are two configurable parameters relating to the COM Server:

- Process name

  This is the name of the COM Server that performs the work based on the ImageMagick library. It should always be IMW_COMServer.exe.
- Pool size

  The value of this configuration parameter is the maximum number of ImageMagick COM servers that can be running at one time.

3.3.6.3 Pixel cache parameters

A number of parameters can be configured for the pixel cache in the Image3 plug-in.

Table 3-9: Pixel cache parameters for Image3 plug-in

Parameter name	Function	XML attribute
Memory size	Sets the maximum amount of memory in megabytes to allocate for the pixel cache from the heap. When this limit is exceeded, the image pixels are cached to memory-mapped disk.	`doc_token_limit_memory`
Memory map size	Sets the maximum amount of memory map in megabytes to allocate for the pixel cache. When this limit is exceeded, the image pixels are cached to disk.	`doc_token_limit_map`
Area size	Sets the maximum width x height of an image that can reside in the pixel cache memory. Images that exceed the area limit are cached to disk.	`doc_token_limit_area`

3.3.6.4 Timeout configuration

If your installation of Transformation Services will be processing large images, configure the timeout value in the `image3.xml` configuration file. This value will determine how long Transformation Services will maintain an image in its processes before calling a timeout and causing the processing to fail.

Transformation Services requires at least 2 GB of free memory to process large images. If you expect to process large dimensioned images, it may be necessary to raise the default timeout value, taking your computer's capabilities into consideration.

This value can be configured in the `image3.xml` file:

- application_monitor_timeout

  This is the number of seconds to wait for an image file to process before terminating it and logging the timeout. With the default of 30 minutes, this value appears as follows:

  <APPLICATION_MONITOR_TIMEOUT>1800</
  APPLICATION_MONITOR_TIMEOUT>

The timeout value should be based on the hardware on the machine in question, and the number of and size of the images. If you are experiencing timeouts with this default value, increase it in the config file.

3.4 Configuration tasks specific to Transformation Services - Audio/Video

This section contains procedures for the audio and video-based configuration tasks that are performed with Transformation Services - Audio/Video.

3.4.1 Configuring Documentum CM Server required for large video files

When importing a video file larger than 2.5 GB, the renditions, thumbnail, and storyboard are not created. The Documentum CM Server must be configured to allow processing of files larger than 2.5 GB.

To configure Documentum CM Server for transforming video files larger than 2.5 GB:

1. Locate the file `server.ini` at the following file path:
   - On Windows, `<DOCUMENTUM>\dba\config\<repository>\`
   - On non-Windows, `<DOCUMENTUM>/dba/config/<repository>/`
2. Open the file in a text editor.
3. Locate the section [SERVER_STARTUP].
4. Add the client_session_time key and a value for timeout_in_minutes. The value for timeout_in_minutes should be at least twice your expected maximum video duration, depending on the machine’s performance.

   For example, for a largest expected video size of 2 GB, set it to:

   ```
   client_session_timeout = 120
   ```
5. Restart the Documentum CM Server and Transformation Services product.

3.4.2 Creating different default preview formats on import of video files

Out of the box, Transformation Services creates Flash 9 renditions for the default preview rendition of video files. This can be changed to other formats such as WMV, MPEG4, QuickTime, and WebM. Some profile changes are required to set the default preview rendition to another format.

For example, to configure Transformation Services to generate the WMV format as the default preview, instead of Flash 9, the following profile changes must be made:

1. Check out the video_registration profile from the following folder in the repository: \System\MediaServer\System Profiles.
2. Uncomment the to_wmv “InnerProfile” snippet. This will enable the WMV rendition to be generated out of the box.
3. Move the entire “default_proxy_format” tag from the to_flash9 section to the to_wmv section and update the following format:

   ```
   <InnerTokenMapping LocalProfileToken="wmv" InnerProfileToken=
   "default_proxy_format" Literal="true"/>
   ```

   Ensure that the “default_proxy_format” tag is not present for more than one rendition.
4. Check in the video_registration profile.
5. Restart the Transformation Services.

Note: If Transformation Services is configured for the xCP client, the same change must be made in the video_registration_xcp profile, instead of the video_registration profile. The video_registration_xcp profile does not have the default_proxy_format tag present for any rendition by default, but it can be added afresh for any of the two existing OOTB renditions (MPEG4 or WebM). Similarly if it is configured for the client, same changes must be made in video_registration_d2 profile, and so on. Currently, client registration profile does not generate any other video-format renditions OOTB. Therefore, it will be applicable only if other renditions are enabled earlier.

3.5 Configuration tasks specific to XML Transformation Services

This section provides details about configuring the tasks that are specific to XML Transformation Services.

3.5.1 Configuring filters

Filters allow you to configure the output to include or exclude specific content that has been tagged within the source document. For example, a user guide may have content that applies to specific operating systems on which the application is installed. You can tag content specific to those operating systems to be included in the output.

XML Transformation Services is configured out-of-the-box with the following four filters:

- profile_os

  The following operating systems are available: Windows, Linux, HP-UX, and Mac.
- profile_audience

  The following audience levels are available: Expert, Intermediate, and Beginner.
- profile_revision

  The following revision names are available: New, Changed, Deleted, and Final.
- profile_product

  Default product names are not listed in this filter. You must add the product names that apply to your organization.

If any of these default filters are appropriate for your organization, you must ensure that the source documents are profiled accordingly. To use the operating system profile, for instance, you must tag sections within your source documents with operating system names. Consult the documentation for your authoring system to determine how to profile your source documents.

For many organizations, these filters will require modification, or new filters will be needed.

One transformation profile controls all DITA filters, while three different profiles are required for DocBook transformations. Separate procedures are provided for both DITA and DocBook filter configurations.

3.5.1.1 Adding product names to the product filter

Since each organization will have a unique set of product names for which to filter content, you must add the appropriate product names to the transformation profiles.

Note: By default, the profiles have no product names specified. If you do not add product names as outlined later in this section, users can simply enter the product name in a text box when they perform transformations. However, this approach could lead to errors if incorrect product names are entered. Configuring the product names in the profiles is highly recommended.

To add product names to appear in the product filter list box:

1. Log in to the repository using an appropriate application.
2. Navigate to `/System/Media Server/Profiles/`.
3. Check out or export the `ProfileSchema.dtd` file.

   If you export the file, you only need to do so the first time you modify a profile. Leave the DTD file in your checkout file path so that the profiles you check out in the future can reference the DTD.
4. Check out the appropriate profile:

   - For all DITA transformations, choose the `dita_transform.xml`.
   - For DocBook transformations, choose the `docbook_transform_chm.xml`, `docbook_transform_html.xml`, or `docbook_transform_pdf.xml`, depending on the transformations your organization will be using.

   Note: If these files are not immediately visible, choose **Show All Objects and Versions** from the Show list box.
5. Open the profile in an XML or text editor.
6. Locate the profile_product parameter. Change the controltype from “text” to “listbox”.
7. Add <*ValueList*> tokens to the profile_product parameter.

   For example, if you want to filter content based on model number, you can add values as follows:

```
<Parameter name="profile_product"
                label="Product Filter"
                controltype="listbox"
                datatype="string"
                isRequired="false"
                description="Select any Products You wish...">
<ValueList>
<Value label="Model A">Model A</Value>
<Value label="Model B">Model B</Value>
<Value label="Model C">Model C</Value>
<Value label="Model D">Model D</Value>
</ValueList></Parameter>
```

8. Save your changes.
9. Check in the profile.
10. Stop and restart the Transformation Services.

The correct product names will appear in the product list box when users are requesting transformations.

3.5.1.2 Configuring other DITA filters

The `dita_transform.xml` profile file controls the filters for all DITA transformations.

To modify an existing DITA filter:

1. Log in to the repository using an appropriate application.
2. Navigate to `/System/Media Server/Profiles/`.
3. Check out or export the `ProfileSchema.dtd` file.

   If you export the file, you only need to do so the first time you modify a profile. Leave the DTD file in your checkout file path so that the profiles you check out in the future can reference the DTD.
4. Check out the `dita_transform.xml` file.

   Note: If this file is not immediately visible, choose **Show All Objects and Versions** from the Show list box.
5. Open the profile in an XML or text editor.
6. Locate the filter parameter to modify within the `Transcodings` section.

   For example, the revision filter appears as follows:

```
<Parameter name="profile_revision"
                label="Revision Filter"
                controltype="listbox"
                datatype="string"
                isRequired="false"
                description="Select Any Revision Flags You Wish to
Exclude">
<ValueList>
<Value label="New">New</Value>
<Value label="Changed">Changed</Value>
<Value label="Deleted">Deleted</Value>
<Value label="Final">Final</Value>
</ValueList>
</Parameter>
```

7. Add, modify, or delete a value within the list, being careful to follow the proper syntax.

   For example, you could change the values for profile_revision, as follows:

```
<Parameter name="profile_revision"
                label="Revision Filter"
                controltype="listbox"
                datatype="string"
                isRequired="false"
                description="Select Any Revision Flags You Wish to
Exclude">
<ValueList>
<Value label="First draft">First draft</Value>
<Value label="Technical review">Technical review</Value>
<Value label="Approved">Approved</Value>
</ValueList>
</Parameter>
```

8. Save your changes.
9. Check in the profile.
10. Stop and restart the Transformation Services.

The modified values will appear when users request transformations for DITA files.

To add a new filter for DITA source files:

1. Log in to the repository using an appropriate application.
2. Navigate to `/System/Media Server/Profiles/`.
3. Check out or export the `ProfileSchema.dtd` file.

   If you export the file, you only need to do so the first time you modify a profile. Leave the DTD file in your checkout file path so that the profiles you check out in the future can reference the DTD.
4. Check out the `dita_transform.xml` file.

   Note: If this file is not immediately visible, choose **Show All Objects and Versions** from the Show list box.
5. Open the profile in an XML or text editor.
6. Add a parameter for the filter you wish to add.

   For example, to add a filter based on application servers, the new parameter might appear as follows:

```
<Parameter name="profile_appserver"
                label="Application Server Filter"
                controltype="listbox"
                datatype="string"
                isRequired="false"
                description="Select an application server to exclude">
<ValueList>
<Value label="BEA">BEA</Value>
<Value label="Oracle">Oracle</Value>
<Value label="Sun One">Sun One</Value>
<Value label="Tomcat">Tomcat</Value>
</ValueList>
</Parameter>
```

7. New or existing filters can also have a blank value, to be used in 'inclusion' mode. Under *<ValueList>* node add a line such as the following:

```
<Parameter name="profile_appserver"
label="Application Server Filter"
controltype="listbox"
datatype="string"
```

```
isRequired="false"
description="Select an application server to exclude">
<ValueList>
<Value label=" "> </Value>
<Value label="BEA">BEA</Value>
<Value label="Oracle">Oracle</Value>
<Value label="Sun One">Sun One</Value>
<Value label="Tomcat">Tomcat</Value>
</ValueList>
</Parameter>
```

Note: The value itself cannot be empty. Use at least one blank space.

8. Save your changes.

9. Check in the profile.

10. Navigate to `/System/Media Server/command line Files`.

11. Check out the `dita_transform_command.xml` file.

    Note: If this file is not immediately visible, choose **Show All Objects and Versions** from the Show list box.

12. Open the profile in an XML or text editor.

13. In the `input_properties` section, add the new filter. This illustrates the entry for the application server example:

```
<!-- property maps to the 'profile_appserver' parameter
- used as a conditional processing filter
                (default value '' indicates that there are no
filter tokens) -->
<property name="profile_appserver" value=""/>
```

14. In this same file, add an entry to the `conditional_params` section. Continuing the application server example, this addition would appear as follows:

```
<param name="appserver" use-property="profile_appserver"/>
```

15. Save your changes.

    Note: If this file is not immediately visible, choose **Show All Objects** and **Versions** from the Show list box.

16. Check in the profile.

17. Stop and restart the Transformation Services.

The new filter will appear when users request transformations for DITA files.

3.5.1.3 Using inclusion mode for DITA filters

By default, XML Transformation Services excludes the conditional content based on selected filters in the Transformation Wizard interface. There may be cases where a customer wants to specify what content should be included. Configuration is required to support such a scenario. After the change is in place all transformation will use filtering in “inclusion” mode.

This feature works with DITA OT 1.4.2 or later. While in ‘inclusion’ mode, if you need to tell XML Transformation Services not to include any content that is tagged with the specified filter, you will need to add a blank value to the profile parameter associated with that filter.

To configure inclusion mode for DITA conditional processing:

1. Log in to the repository using an appropriate application.
2. Navigate to `/System/Media Server/command line Files`.
3. Check out the `dita_transform_command.xml` file.

   Note: If this file is not immediately visible, choose **Show All Objects** and **Versions** from the Show list box.
4. Open the profile in an XML editor.
5. Uncomment the following line:

```
<property name="cond_proc_action" value="include"/>
```

6. Save your changes.
7. Check in the profile.
8. Stop and restart the Transformation Services.

3.5.1.4 Configuring other DocBook filters

Three profile files handle DocBook transformations, based on the output format:

- docbook_transform_chm
- docbook_transform_html
- docbook_transform_pdf

Each of these profiles contains a unique set of parameters. They also include the standard four filters as for DITA files.

To modify an existing DocBook filter:

1. Log in to the repository using an appropriate application.
2. Navigate to `/System/Media Server/Profiles`.

3. Check out or export the `ProfileSchema.dtd` file.

   If you export the file, you only need to do so the first time you modify a profile. Leave the DTD file in your checkout file path so that the profiles you check out in the future can reference the DTD.

4. Check out the `docbook_transform` file you wish to modify.

   Note: If these files are not immediately visible, choose **Show All Objects and Versions** from the Show list box.

5. Open the profile in an XML or text editor.

6. Locate the filter parameter you wish to modify within the `Transcodings` section.

   For example, the audience filter appears as follows:

```
<Parameter name="profile_audience"
                label="Audience Filter"
                controltype="listbox"
                datatype="string"
                isRequired="false"
                default="">
<ValueList>
<Value label="Expert">Expert</Value>
<Value label="Intermediate">Intermediate</Value>
<Value label="Beginner">Beginner</Value>
</ValueList>
</Parameter>
```

7. Add, modify, or delete a value within the list, being careful to follow the proper syntax.

   For example, you could change the values for profile_audience, as follows:

```
<Parameter name="_audience"
                label="Audience Filter"
                controltype="listbox"
                datatype="string"
                isRequired="false"
                default="">
<ValueList>
<Value label="User">User</Value>
<Value label="Administrator">Administrator</Value>
<Value label="Developer">Developer</Value>
</ValueList>
</Parameter>
```

8. Save your changes.

9. Check in the profile.

10. Stop and restart the Transformation Services.

The modified values will appear when users request transformations for DocBook files.

To add a new filter for DocBook source files:

1. Log in to the repository using an appropriate application.

2. Navigate to `/System/Media Server/Profiles/`.

3. Check out or export the `ProfileSchema.dtd` file.

   If you export the file, you only need to do so the first time you modify a profile. Leave the DTD file in your checkout file path so that the profiles you check out in the future can reference the DTD.

4. Check out the docbook_transform file for which to add a filter.

   Note: If these files are not immediately visible, choose **Show All Objects and Versions** from the Show list box.

5. Open the profile in an XML or text editor.

6. Add a parameter for the filter you wish to add.

   For example, if your documents contain different regulatory statements pertaining to different countries, you might require a country filter. Such a parameter can appear as follows:

```
<Parameter name="profile_country"
                label="Country Filter"
                controltype="listbox"
                datatype="string"
                isRequired="false"
                description="Include content specific to this country">
<ValueList>
<Value label="USA">USA</Value>
<Value label="Canada">Canada</Value>
<Value label="Mexico">Mexico</Value>
</ValueList>
</Parameter>
```

7. Save your changes.

8. Check in the profile.

9. Navigate to `/System/Media Server/command line Files`.

10. Check out the `docbook_transform_command.xml` file.

    Note: If this file is not immediately visible, choose **Show All Objects and Versions** from the Show list box.

11. Open the profile in an XML or text editor.

12. In the `input_properties` section, add the new filter. This illustrates the entry for the country example:

```
<!-- property maps to the 'profile_country' parameter
- used as a conditional processing filter
               (default value '' indicates that there are no
filter tokens) -->
<property name="profile_country" value=""/>
```

13. In this same file, add an entry to the `conditional_params` section. Continuing the country example, this addition would appear as follows:

```
<param name="country" use-property="profile_country"/>
```

14. Save your changes.

15. Check in the profile.

16. Stop and restart the Transformation Services.

The new filter will appear when users request transformations for DocBook files.

3.5.2 Configuring XML Transformation Services with custom style sheets

This procedure allows you to configure XML Transformation Services with custom style sheets. This is required to transform custom XML schema to PDF or HTML (“Transform XML to HTML” or “Transform XML to PDF” profiles). Further, additional profile parameters may be also required. The following procedures include these configurations and parameters.

> **Caution**
> Ensure your custom XSL is compliant with the configured XSLT processor.

To configure XML Transformation Services with custom style sheet:

1. Open the *`<Documentum Content Transformation Services_HOME>`*`\config\xpub\xpub.xml` file.

2. Locate the `STYLESHEET` section.

3. Update the *<STYLESHEET>* node corresponding to “XML” handler to point to the new file on your local system.

```
<STYLESHEET handler="XML" name="DEFAULT" fileref="C:\Documentum\
CTS\config\xpub\default_fo.xsl">
```

4. You can also add additional styles sheets for a handler and control which one to use using the “use_stylesheet” profile parameter.

   For example, you can have the following :

```
<STYLESHEET handler="XML" name="DEFAULT" fileref="C:\Documentum\
CTS\config\xpub\default_fo.xsl">
<STYLESHEET handler="XML" name="MY_TRANSFORM1" fileref="C:\myxsl\transform1.xsl">
<STYLESHEET handler="XML" name="MY_TRANSFORM2" fileref="C:\myxsl\transform2.xsl">
```

5. Check out the corresponding profile and add a new parameter named “use_stylesheet”.

   The following example matches the style sheets from the previous step:

```
<Parameter name="use_stylesheet" label="Style Sheet" controltype="listbox"
 datatype="string" isRequired="false">
        <ValueList>
                <Value label="Transform using my_transform1.xsl">MY_TRANSFORM1</
Value>
                <Value label="Transform using my_transform1.xsl">MY_TRANSFORM2</
Value>
        </ValueList>
</Parameter>
```

Note: The values in the list are the same as the values in the attribute ‘name’ of the <*STYLESHEET*> nodes mentioned in step 4.

6. Check in the profile and save `xpub.xml`.

7. Restart the Transformation Services.

To configure additional parameters for transforming custom XML:

1. Check out the profile (for example, “`xml_transform_pdf.xml`”) from the following folder in the repository: `/System/Media Server/Profiles/`.

2. Open the profile in an XML or text editor.

3. Locate the `Transcoding` tag and add the new parameter(s) as required.

4. Save and check in the file.

5. Check out the command line file (for example, “`xml_transform_pdf_command.xml`”) from the following folder in the repository: `/System/Media Server/command line Files`.

6. Locate the `xslt_params` section (for example, <*xslt_params format=“pdf”*>). For each of the parameters that you add in step 3, add a new tag, using the new parameter name:

```
<param name="new_xslt_param_name1" use-property=" new_xslt_param_
name1"/>
<param name="new_xslt_param_name2" use-property=" new_xslt_param_
name2"/>
</xslt_params>
```

7. Save and check in the file.

8. Restart the Transformation Services.

3.5.3 Configuring XML Transformation Services to transform XML content

There are scenarios where XML Transformation Services should be configured to handle transformations successfully and give the desired outcome. Some examples of such cases are switching graphic files to different resolution or removing unsupported characters from file names and references.

Follow this procedure if your XML content (for example, DITA) requires special processing:

To configure XML Transformation Services to transform XML content:

1. Locate and open the following file *`<Documentum Content Transformation Services_HOME>`*`\config\xpub\xpub.xml`.

2. Locate the `PATCHING_CONFIG` section.

3. From the PARSE sub node, choose the required attributes:

   - *xpath*

     Represents an Xpath expression to an XML element that should be parsed.

   - *handler*

     Represents the type of XML document being processed.

   - *link_type*

     Represents the type of reference. For DRL patching, the value is conref; for graphics patching the value is image; for special characters patching, the value is specialchar.

   - *patcher*

     Specifies a java class that processes the XML element.

   - *source*

     Specifies the content origins for which processing should occur. Possible values are “repo”, “file”, and “any”.

4. From the <*FILE_FILTER*> node, choose the required attribute:

   - *extension*

     Lists file extensions of XML content that should be processed.

5. Choose the <*IMG_PATCHING*> node if graphics references should be processed. This is used to map graphics in the original content to a repository rendition that will replace them. Choose the required attributes:

   - <*CONFIG*> sub-node:

     – *target* – target format (file extension, not Documentum dm_format) of transformation

     – *handler* – type of XML content (see 3-second bullet)

     <*IMAGE_REF*> sub-node:

     – *source_ext*

       represents the extension of graphic asset file to be replaced

     – *rend_dctm_format*

       represents the Documentum format of the rendition that should replace the original file

     – *rend_flag*

       specifies the flag value to identify a rendition

     – *rend_flag_type*

       Represents the flag type to be checked when used with rend_flag.

Possible rendition attributes are: page_modifier, rendition_name, or rendition_description_name. If you leave the space empty, the first rendition that matched the requested format, is selected.

Note: This features is relevant only if XML Transformation Services is used in combination with Media Transformation Server. Media Transformation Server is needed to generate renditions for graphic assets.

6. For DRL Patching, choose the `DRL_PATCHING` section. Choose the required attributes from `PARAM` nodes.

   - – *brokenlink*

     Specifies a repository file path for a DITA document that serves as a broken link page.

     – *depth*

     Controls how many levels should be processed when XML files are being exported as part of processing an existing level.

     – *special_chars*

     Lists special characters (separated by '/') that should be replaced with '_' in XML file names and references.

7. Restart the Transformation Services.

3.5.4 Configuring a custom XSLT processor

The profile `xml_transform_html` (label "Transform XML to HTML") allows you to perform XML (custom schema) to HTML transformation using XSL. The transformation needs an XSLT file. A default sample file is provided by default in *<Documentum Content Transformation Services_HOME>*`\config\xpub\`. Configurations related to this transformation are in *<Documentum Content Transformation Services_HOME>*`\config\xpub\xpub.xml` under doc_type "XML2HTML". Customers must provide their own XSL and configure it under *<STYLESHEET handler="XML2HTML" name="DEFAULT" fileref="....."/>*. Multiple XSL files can be configured for XML to HTML.

The XSLT processor can be configured at the plug-in level or at the Profile level. Profile level will overwrite the plug-in level.

To configure the XSLT processor at the plug-in level:

1. Take a backup of the `xpub.xml` file located at *<Documentum Content Transformation Services_HOME>*`\config\xpub\`.

2. Open `xpub.xml` in an XML or text editor.

3. Locate the node //XML_TRANSFORM_CONFIG/XML_PROPERTIES and add (or edit) XML_PROPERTY child. The following example is the default value for Saxon 6.x:

```
<XML_PROPERTY name="transformer_factory"
value="com.icl.saxon.TransformerFactoryImpl"/>
```

Notes

- Attribute 'name' should always be “transformer_factory”.
- Attribute 'value' should be the desired implementation of javax.xml.transform.TransformerFactory.

4. Save and close `xpub.xml`.

5. Restart the XML Transformation Services server.

To configure the XSLT processor at the transformation level:

1. In the repository, navigate to the command line file used for the transformation (for example, `/System/Media Server/command line Files/xml_transform_pdf_command.xml`).

2. Checkout the command line file.

3. Open the command line file in an xml or text editor.

4. Locate the node `//xml_transform_command/input_properties/transformer_factory` (add it if missing).

5. Set the attribute 'classname' with the desired implementation of javax.xml.transform.TransformerFactory similar to the following example (using Saxon 9.x):

   ```
   <transformer_factory classname="net.sf.saxon.TransformerFactoryImpl" />
   ```

6. Save and close the command line file and check it back in to the repository.

7. Restart the XML Transformation Services server.

To alter the classpath for the new processor:

1. Deploy the libraries on *`<Documentum Content Transformation Services_HOME>`*`\lib\`.

2. Edit the MANIFEST.MF inside *`<Documentum Content Transformation Services_HOME>`*`\lib\cts.jar\META-INF\` by appending the new libraries to the end. Ensure you follow the required protocol when listing jars: the list of jars is within cols [1-70]; jars are separated by two spaces.

Note: Transformation Services - Media and Transformation Services - Documents are not compatible with Saxon XSLT processor. As such, any new XLST that is deployed to XML Transformation Services should be configured as described in this section. Then, the \META- INF\services \javax.xml.transform.TransformerFactory entry should be removed from any jar containing it, in order to prevent such a library to “broadcast” itself as an XSLT processor.

Chapter 4

Profiles and Profile Modification

4.1 Understanding Profiles

Transformation Services is shipped with sets of predefined profiles that provide a simple method for an application to access plug-ins used to identify, process, and transform source files.

Before attempting to customize or reuse profiles, it is important to understand how transformation profiles and command line files work together.

4.1.1 What are profiles

Profiles are loaded to the memory of Transformation Services from the repository after it starts up. Applications pass profile parameters (in the XML format) to Transformation Services.

There are two types of transformation profiles:

- *System profiles* are invoked silently, either through registration (import or check in of files) or by other profiles, to control the processing of content. Some system profiles relate to transformations, while others relate to internal processing such as property extraction or metadata embedment. These profiles are not exposed to users.
- *User profiles* are exposed to the user some client applications. Users can also use WebServices to execute the user profiles. Some user profiles include user-defined parameters, such as image height or width.

Both types of transformation profiles are related to specific *command line files*. The profiles, plug-ins, and command line files work together to perform transformations and other processing functions offered through Transformation Services.

4.1.1.1 System profiles

System profiles are only used internally by Transformation Services, and are not exposed through the client application. Generally speaking, the system profiles specify the source and target formats to be processed, and invoke the necessary command line files.

Note: System profiles “document_to_pdf” and “document_to_html” are an exception, and are exposed through Digital Asset Manager.

When files are imported or checked in to the repository, they are first processed by the *register* profile. This process is called registration. The register profile acts as a gatekeeper, directing the way in which imported content is handled. Only those profiles explicitly referenced in the register profile are invoked on import.

You can also customize system profiles to perform the following tasks:

- Enable thumbnail and storyboard creation when an item is imported to the repository.
- Specify the order in which plug-ins are executed for a particular transformation.
- Change the dimensions of storyboards and low-resolution renditions.

System profiles are dm_media_profile objects stored in the repository at `/System/Media Server/System Profiles`.

Register profiles are a special type of system profile, and are discussed in the following section.

4.1.1.1.1 Register Profile

The register profile controls the processing of rich media-enabled content that is imported or checked in to the repository. The register profile can automatically invoke a thumbnail, storyboard, or low-resolution (proxy) transformation of imported files.

The register profile is the most commonly used system profile. It is unlike any other system profile, because it does not actually perform a transformation. It directs the transformation server to the appropriate system *transformation* profile, which invokes the related command line file.

Whenever rich media-enabled content is imported or checked in, the transformation server accesses the register profile. Some of the configuration procedures are commonly required when the application is initially installed.

The basic functionality of the register profile is to register all supported file formats and enable property extraction and thumbnail, proxy, and storyboard creation on import.

Configure Transformation Services to create renditions on import. Following are the available configurations:

- Add format pairs to `Formats` section to enable PDF renditions on import.
- Add <*InnerProfile*> tags to enable thumbnail and storyboard creation on import.

The register profile contains four inner profiles, `register_mobile.xml`, `register_xcp.xml`, `register_d2.xml`, and `register_legacy.xml`, out of the box. By default, `register_xcp.xml` is executed. You can control the inner profile (register_xcp, register_legacy, register_d2, a custom register profile, or all together) that needs to be executed from the `register.xml` by updating `CTSProfileService.xml`.

The register_<*xxxx*>.xml provides the following information to the server:

- The `Formats` section contains all the formats supported by Transformation Services to generate default renditions. Source and target formats are the same in the register_<*xxx*> profile. This is because processing does not start from this profile; the source is passed to the next profile.
- The <*InnerProfile*> path specifies the transformation profile that the server must load. To enable additional default renditions, add an `InnerProfile` tag to the register_<*xxxx*>.xml for these additional capabilities.

Note: The register_<*xxxx*>.xml can be `register.xml`, `register_mobile.xml`, `register_xcp.xml`, `register_d2.xml`, and `register_legacy.xml`.

4.1.1.1.1.1 Register profile for xCP clients (register_xcp.xml)

The `ProfileSequence` section of the `register_xcp.xml` contains Inner Profiles entries that create specific renditions required by xCP clients, upon importing a rich media-enabled format.

For example, it creates low resolution storyboards from each page of PDF, multi-page TIFF, or Microsoft Office documents.

4.1.1.1.1.2 Register profile for client (register_d2.xml)

The `ProfileSequence` section of the `register_d2.xml` contains Inner Profiles entries that create specific renditions required by client, upon importing a rich media-enabled format.

4.1.1.1.1.3 Register profile for Legacy clients (register_legacy.xml)

The `ProfileSequence` section of `register_legacy.xml` contains Inner Profiles entries to support legacy clients such as Webtop, Digital Asset Manager, Media Workspace, and so on. It performs a profile sequence that creates thumbnails and low-resolution renditions, extracts file properties, processes PowerPoint and PDF files, and extracts XMP metadata.

4.1.1.1.1.4 Register profile for specific clients

The `register.xml` allows you to configure rendition requirements of specific clients such as the xCP Viewer, client, Mobile and the existing legacy clients. It aims at externalizing the rendition requirements of these specific clients (xCP, client, Mobile, and so on) into their respective profile configuration files, so that their specific requirements can be configured as needed, without compromising on performance and storage capacity. This provides a mechanism to selectively turn on or off the rendition generation for these clients on a per-repository basis.

The following is the `ProfileSequence` section of `register.xml`:

```
<ProfileSequence>
            <InnerProfile
path="/System/Media Server/System Profiles/register_legacy"
waitOnCompletion="false" forClient="LEGACY" useTargetFormat="true"
> </InnerProfile>
<InnerProfile
path="/System/Media Server/System Profiles/register_mobile"
```

```
waitOnCompletion="false" forClient="MOBILE" useTargetFormat="true"
> </InnerProfile>
<InnerProfile path="/System/Media Server/System Profiles/register_xcp"
waitOnCompletion="false" forClient="XCP" useTargetFormat="true"> </InnerProfile
>
<InnerProfile
path="/System/Media Server/System Profiles/register_xcp_adv"
waitOnCompletion="false" forClient="XCP_ADV" useTargetFormat="true"
> </InnerProfile>
<InnerProfile path="/System/Media Server/System Profiles/register_d2"
waitOnCompletion="false" forClient="D2" useTargetFormat="true"> </InnerProfile
>
</ProfileSequence>
```

Notes

- The presence of a client in the *forClient* attribute in the inner profile does not automatically mean that the inner profile is executed. This behavior is controlled by the setting in `<Documentum Content Transformation Services_HOME>/config/CTSProfileService.xml/`.
- By default, the *forClient* attribute is set to client.

4.1.1.2 User profiles

User profiles process transformations requested by users through a rich media enabled application. These types of transformations give users the flexibility to choose specific files in the repository, and specify parameters (such as header and footer information, watermarks, job options, and security settings) during the transformation.

When the user invokes a parameterized profile, the user is prompted to select or enter the parameters required by the profile, as some parameters are mandatory (for example, overlay_adts profile). The profile and its parameters are passed as a request to the transformation server, which maps the request to the appropriate plug-in and performs the transformation.

The parameters required by a profile can be assigned validation rules, whereby the application validates the values entered by the user. The validation rules can check the unit of measure, format, or similar metrics to avoid processing incorrect values.

"Profiles Installed with Transformation Services products" on page 183 provides a complete list of predefined profiles.

User profiles are dm_media_profile objects stored in the repository at `/System/Media Server/Profiles`.

4.1.1.2.1 Enabling a user profile in a rich media enabled client

When modifying user profiles, ensure that the following tags are present in the profile file:

- <Filter name="CTSProduct" value="<*Documentum Content Transformation Services product initialism*>"/>

  This tag ensures that the profile is visible with the profiles packaged with Transformation Services.

- <Filter name="Visibility" value="Public"/>

  The profile must be set to *Public* to be visible to users.

> **Caution**
>
> The value of <*Documentum Content Transformation Services product initialism*> must only be that product that is configured to the repository, which means that, it can be any one of the following values: MTS, ADTS, XTS, and AVTS. If Transformation Services - Media is configured and the value of <*Documentum Content Transformation Services product initialism*> is set to "AVTS", then this profile will not be executed by the Transformation Services - Media server.

4.1.1.2.2 Chaining and sequencing profiles

You can create new transformations using the transformation profiles that are installed to the repository by Transformation Services. You can build new transformation profiles by using an existing profile as a template for a new profile, by chaining multiple profiles together, and sequencing multiple profiles.

- Chained profiles invoke each profile within them in stages. Transformations occur one at a time, and the result of one transformation is required for the next transformation. The result of a chained profile would be one output file.
- Sequenced profiles specify a list of profiles that are executed one after another. For each profile in the profile sequence, you can specify whether the next profile should wait on the successful completion of the previous task. If there is no specification, profiles can be executed as soon as the task threads are available for processing. This means that each profile does not have to be complete before the next profile starts. The result of sequenced profiles is one output file for each profile in the sequence.

  Note: In the Documentum Content Transformation Services Profile Editor, sequenced profiles are referred to as *parallel* profiles. However, you can consider these terms to be synonymous.

Sequencing is the most common method of reusing profiles.

After chained or sequenced transformations are created, they will appear as transformation options in the user interface (assuming they have been properly tagged for visibility). For example, if you chained a resize profile and a transform profile, the name and description you specified for this new transformation option would appear in the WDK application along with the other transformation profiles.

"Chaining and sequencing profiles" on page 157 provides more information on reusing transformation profiles.

4.1.2 Command-line files

Command-line files are stored in the repository as XML files that can be used by the plug-ins as needed. Each profile is linked to an actual XML command line file for a given transformation.

Note: Only customers with a developer license should modify command line files. Consult the *OpenText Documentum Content Transformation Services Development Guide* for more information about working with command line files.

The command line file contains detailed instructions for the plug-in. It may also contain additional parameters that are not user-selectable. For example, a profile could be written to create a thumbnail that always produces JPEG output. The resulting file format and many of its attributes (such as JPEG quality and color mode) could be controlled at the command level. Command-line files are specific to each plug-in, since the capabilities of plug-ins vary.

Command-line files do not need to be created for profiles that reuse other profiles. A user profile that serves only to chain or sequence other profiles has no need for CommandFilePath entries. Since the DTD does not require their presence, no change is required to exclude them. If present, these entries are ignored, as only the atomic profiles require them.

4.1.3 Transformation Services DTD

All Transformation Services profiles are written in XML. An external document type definition (DTD) reference (`ProfileSchema.dtd`) defines the legal structure of profile files. All profile files must conform to the Transformation Services DTD.

As defined by the DTD, all XML files consist of:

- elements
- attributes
- entities

The Transformation Services DTD file:

```
<!-- CTS Profile DTD v6.0.0.001
     Types and MediaEnabled nodes are specified just for backward compatibility.
     These nodes should not be converted to Filters
-->
<!ELEMENT Profile (Formats, Types?, MediaEnabled*, Filters, Transcodings,
(CommandFilePath+ |
 ProfileChain | ProfileSequence))>
<!-- Profile attributes -->
<!ATTLIST Profile
   xmlns:dctm CDATA #IMPLIED
   name CDATA #REQUIRED
   label CDATA #REQUIRED
   related_objects_only (true | false) "false"
   notify_result (true | false) "false"
   operation (transform | extractProperties) "transform"
   taskImpl CDATA "com.documentum.cts.impl.services.task.CTSTask"
   description CDATA #IMPLIED
   dctm:obj_status CDATA #IMPLIED
   dctm:obj_id CDATA #IMPLIED
   dctm:version_label CDATA #IMPLIED
>
<!ELEMENT Formats (Format+)>
<!ELEMENT Format EMPTY>
<!-- Format attributes -->
<!ATTLIST Format
   source CDATA #REQUIRED
   target CDATA #REQUIRED
>

<!-- Types and MediaEnabled nodes are specified just for backward
compatibility. These nodes should be converted to Filters -->
<!ELEMENT Types (Type*)>
<!ELEMENT Type (#PCDATA)>
<!-- Format attributes -->
<!ATTLIST Type
   name CDATA #REQUIRED
>

<!ELEMENT MediaEnabled (#PCDATA)>
<!ATTLIST MediaEnabled
            category CDATA #REQUIRED
>
<!-- Types and MediaEnabled nodes are specified just for backward
compatibility. These nodes should be converted to Filters -->

<!ELEMENT Filters (Filter+)>
<!ELEMENT Filter EMPTY>
<!-- Filter attributes
        Predefined filter names: CTSProduct, AppProduct, Visibility, XMLApp,
ObjectType, VirtualDocOnly
 -->

<!ATTLIST Filter
   name CDATA #REQUIRED
   value CDATA #REQUIRED
>

<!ELEMENT Transcodings (Transcode*)>
<!ELEMENT Transcode (Parameter*)>
<!-- Transcode attributes -->
<!ATTLIST Transcode
   name CDATA #REQUIRED
   label CDATA #REQUIRED
>
<!ELEMENT Parameter ((ValueList | ValueRange | Value* | TupleElement+ |
ContentObject+), (depends-on-param | depends-on-format)?)>
<!-- Parameter attributes -->
<!ATTLIST Parameter
   name CDATA #REQUIRED
   label CDATA #REQUIRED
   description CDATA #IMPLIED
   controltype (list | range | text | text-block | color-picker |
object | listbox) #REQUIRED
   datatype (string | integer | float | hex | tupleSequence | content)
#REQUIRED
   dql CDATA #IMPLIED
   default CDATA #IMPLIED
   isRequired (true | false) "true"
>

<!ELEMENT ValueList (Value+)>
<!ELEMENT Value (#PCDATA)>
<!-- Value attributes -->
<!ATTLIST Value
   label CDATA #REQUIRED
>

<!ELEMENT ValueRange (MinValue, MaxValue)>
<!ELEMENT MinValue (#PCDATA)>
<!ELEMENT MaxValue (#PCDATA)>

<!ELEMENT depends-on-param (Value+)>
<!ATTLIST depends-on-param
   param-name CDATA #REQUIRED
   action (hide | maintain_proportion ) #REQUIRED
>

<!ELEMENT depends-on-format (ValueList)>
<!ATTLIST depends-on-format
   format-name (source | target) #REQUIRED
   action (hide | show ) #REQUIRED
>

<!ELEMENT TupleElement (#PCDATA)>
<!ATTLIST TupleElement
   label CDATA #REQUIRED
   name CDATA #REQUIRED
   datatype (string | integer | hex | date) #REQUIRED
>

<!ELEMENT ContentObject EMPTY>

<!-- The remaining elements are for server-side use only.
They should NOT be
     parsed by application other than Media Services. -->
<!ELEMENT CommandFilePath (#PCDATA)>
<!-- CommandFilePath attributes -->
<!ATTLIST CommandFilePath
   mptype CDATA #REQUIRED
>

<!ELEMENT ProfileChain (InnerProfile+)>
<!ELEMENT ProfileSequence (InnerProfile+)>

<!ELEMENT InnerProfile (InnerTokenMapping*)>
<!-- waitOnCompletion will be ignored of the parent element of
InnerProfile is
     ProfileChain.  It is intended for use only with
ProfileSequence -->
<!-- InnerProfile attributes -->
<!ATTLIST InnerProfile
   path CDATA #REQUIRED
   waitOnCompletion (true | false) "true"
   useTargetFormat (true | false) "true"
   innerProfileId CDATA #IMPLIED
   forClient CDATA #IMPLIED
>
<!-- forClient is required for registering assets based on clients-->

<!ELEMENT InnerTokenMapping EMPTY>
<!-- InnerTokenMapping attributes -->
<!ATTLIST InnerTokenMapping
   LocalProfileToken CDATA #REQUIRED
   InnerProfileToken CDATA #REQUIRED
   Literal (true | false) "false"
>
```

4.2 Modifying Transformation Profiles

The Transformation Services application includes a set of predefined transformation profiles that perform the functionality prescribed by the application. Some profiles can be modified to optimize their functionality for your organization.

This section describes how to make minor modifications to existing profiles. It presents the most common modifications that are required by our customers.

Note: Profile Editor is available through Documentum Administrator, which simplifies the transformation modification process for Transformation Services. Some modifications described in this chapter, such as chaining and sequencing, can be performed using the Profile Editor.

More advanced editing, such as reordering the plug-ins invoked by certain profiles, can only be achieved by modifying the XML files directly. This section gives procedures for working with XML files.

Transformation Services profiles are written in XML and must conform to the document type definition (DTD). Profiles are mapped to XML command line files which are plug-in specific, meaning that there is one command line file for every profile.

4.2.1 Getting started

Review this section to gain an understanding of where to find profile files, and how to make modifications to these files.

This section contains the following topics:

- “Locating the profiles” on page 151 tells you where the XML files are located in the repository. Profiles should be checked out, modified, and then checked into these file paths.
- “Understanding the modification procedure” on page 152 outlines the basic steps to follow when modifying profiles.

4.2.1.1 Locating the profiles

All transformation profiles can be found in the System cabinet of your repository. Use an application such as Documentum Administrator to access the folders that contain Transformation Services profiles. When you need to modify profiles, check out the files from these folders.

The transformation profile components are stored in these file paths in the repository:

- `/System/Media Server/command line Files/`

  This folder stores command line files as dm_sysobject objects. If you want to modify command line files, you must hold a developer's license.

- `/System/Media Server/Profiles/`

  This folder stores profiles as dm_media_profile objects. These are the user profiles that are exposed by rich media enabled applications.

- `/System/Media Server/System Profiles/`

  This folder stores profiles as dm_media_profile objects. These are the system profiles that are only used internally by Transformation Services, and are not exposed through the client application.

Caution
The profile folder file paths listed indicate where the profiles are placed when you install and configure Transformation Services. The paths to these profiles are coded in all files that invoke other profiles. For example, the register profile may invoke other system profiles, and many system profiles invoke command line files. *If you change the file path of these folders*, you must modify the paths within the profile files accordingly. Otherwise, transformations will fail.

4.2.1.2 Understanding the modification procedure

Transformation Services allows for a great deal of flexibility through the modification of profiles. For example, you can change image output size, or combine multiple operations into one profile to create a one-step transformation. There are many options available for customizing profiles for your organization.

This section provides the basic procedure for modifying a transformation profile. Specific modification procedures are provided in subsequent sections.

After you modify a dm_media_profile object and check it in, Transformation Services automatically refreshes the profile without the need for a restart.

To modify an existing transformation profile:

1. Log in to the repository using an appropriate application.
2. Check out the profile you want to modify. If you are unsure where the profiles reside in the repository, "Locating the profiles" on page 151 provides more information.

   Note: If these files are not immediately visible, choose **Show All Objects and Versions** from the Show list box.

3. Open the profile in an XML or text editor. Make the necessary changes and save the file.
4. Check in the profile.

The profile has now been modified.

4.2.2 Disabling processing on import

In Transformation Services, certain processing – such as thumbnail or storyboard generation – occurs automatically when a rich media-enabled format is imported or checked in to a repository. The Transformation Services configuration utility updates dm_format objects for common image, audio, and video formats to make them rich media-enabled by default.

You may want to disable this processing for certain formats, or disable all processing on import. The procedures in this section address various ways to disable processing on import.

Following options are available when making this modification:

- "Disabling a process for all imported files" on page 153
- "Disabling all processing for specific file formats" on page 154
- "Disabling a process for a specific file format" on page 155

Note: The following examples are based on `register_legacy.xml` and can be applied for `register_xcp.xml`.

4.2.2.1 Disabling a process for all imported files

If required, you can disable a process for all source files that are imported to the repository. For example, you can modify the register profile so that no thumbnails are generated on import.

To disable a particular process for all source files:

1. Log in to the repository using an appropriate application.

2. Check out the `register_legacy.xml` file.

3. Locate the `InnerProfile` tag that relates to the process you want to disable.

   To identify the correct `InnerProfile` tag, look for the name of the profile it invokes. For example, in the Transformation Services `register_legacy.xml` file, the `InnerProfile` tags appear as follows:

   - For thumbnail generation:

     ```
     <InnerProfile path="/System/Media Server/System Profiles/thumbnail"
     waitOnCompletion="false">
     <InnerTokenMapping LocalProfileToken="jpeg_th" InnerProfileToken=
     "doc_token_targetFormat" Literal="true"/>
     <InnerTokenMapping LocalProfileToken="false" InnerProfileToken=
     "overwrite_rendition" Literal="true"/>
     ...
     </InnerProfile>
     ```

   - For low-resolution renditions:

     ```
     <InnerProfile path="/System/Media Server/System Profiles/
     autoGenProxy" waitOnCompletion="false">
     <InnerTokenMapping LocalProfileToken="jpeg_lres" InnerProfileToken
     ```

```
="doc_token_targetFormat" Literal="true"/>
<InnerTokenMapping LocalProfileToken="jpeg_lres" InnerProfileToken
="default_proxy_format" Literal="true"/>
<InnerTokenMapping LocalProfileToken="false" InnerProfileToken=
"overwrite_rendition" Literal="true"/>
...
</InnerProfile>
```

- For extracting file properties:

```
<InnerProfile path="/System/Media Server/System Profiles/
importExtractProperties" waitOnCompletion="false">
<InnerTokenMapping LocalProfileToken="false" InnerProfileToken=
"overwrite_rendition" Literal="true"/>
</InnerProfile>
```

4. Choose to either temporarily or permanently disable this process.

   a. To temporarily disable the process, comment out the entire <*InnerProfile*> tag. This will allow you to easily enable the process at another time. “How do I “uncomment” text in XML files?” on page 180 provides more information to add comments to an XML file.

   b. To permanently disable the process, remove the entire <*InnerProfile*> tag.

5. Save the file.

6. Check in the register file to the repository.

The selected process is now disabled for imported files. If you have commented out the profile, you can enable the process at a later time by removing the comment syntax from the file.

4.2.2.2 Disabling all processing for specific file formats

There are two ways to disable import processing for specific file formats: by modifying the register profile, or by disabling the richmedia_enabled flag using DQL.

To disable all import processing for a specific format by modifying the register profile:

1. Log in to the repository using an appropriate application.

2. Check out the `register_legacy.xml` file.

3. Locate the <*Format*> entry for the file format for which you want to disable processing.

   For example, the entry for GIF files appears as follows:

```
<Format source="gif" target="gif"/>
```

4. Choose to either temporarily or permanently disable processing for this file format.

   a. To temporarily disable this format, comment out the <*Format*> tag. This will allow you to easily enable the process at another time. “How do I “uncomment” text in XML files?” on page 180 provides nformation to add comments to an XML file.

b. To permanently disable this format, remove the <*Format*> tag.

5. Save the file.

6. Check in the register file to the repository.

The selected process is now disabled for imported files having this format. If you have commented out the format, you can enable this format to be processed at a later time by removing the comment syntax from the file.

To disable all import processing for a specific format by using DQL:

1. Disable the richmedia_enabled flag by running the following DQL statement:

```
update dm_format object set richmedia_enabled = 0 where
name in ('<format name>')
```

For example, to disable processing on import for Microsoft Excel and Word formats, run the following DQL statement:

```
update dm_format object set richmedia_enabled = 0 where
name in ('excel8book','msw8')
```

2. Restart the Transformation Services repository.

4.2.2.3 Disabling a process for a specific file format

It is possible to prevent a particular process – such as thumbnail generation – when a specific file format is imported to the repository. To make this change, you need to remove or comment out the `Format` tag from the system profile for the process. Check the register profile to ensure you are modifying the correct system profile.

The following procedure uses the example of suppressing thumbnails for imported GIF files.

To disable an import process for a specific file format:

1. Log in to the repository using an appropriate application.

2. Check out the system profile that relates to the import process you want to disable.

   For example, thumbnails in Transformation Services - Media are processed by the **thumbnail.xml** system profile.

   Note: If you are unsure of the system profile name, check the register profile to see which profile is invoked for this process.

3. Locate the <*Format*> pair relating to the file format for which you want to disable the process.

   For example, the <*Format*> pair relating to GIF files in the thumbnail system profile is:

```
<Format source="gif" target="jpeg_th"/>
```

4. Choose to either temporarily or permanently disable processing for this file format.

   a. To temporarily disable this format, comment out the <*Format*> tag. This will allow you to easily enable the process at another time. “How do I “uncomment” text in XML files?” on page 180 provides information to add comments to an XML file.

   b. To permanently disable this format, remove the <*Format*> tag.

5. Save the file.

6. Check in the register file to the repository.

7. Restart the Transformation Services so that the change takes effect.

When files of this format are imported to the repository, this process will not take place.

4.2.3 Changing dimensions of image renditions

If the default image dimensions for thumbnails or low-resolution renditions are not ideal for your organization, you can customize these values by following the procedures in this section.

Transformation Services can create thumbnails and low-resolution renditions when files are imported or checked in to the repository. Transformation Services is configured out-of-the-box for this functionality. Image dimensions are specified in the register profile.

To modify the dimensions of images created by Transformation Services on import:

1. Check out the `register_xxx.xml` file from the repository.

2. Open the file in an XML or text editor.

3. Modify the image size of any of these renditions:

   - small thumbnails
   - medium thumbnails
   - large thumbnails
   - low-resolution renditions
   - PowerPoint thumbnails, low-resolution renditions, storyboards, or low-resolution storyboards

   For example, low-resolution renditions are configured by default to be 300 pixels in width and 300 pixels in height. You can modify these default values, which appear as follows:

```
<InnerProfile path="/System/Media Server/System Profiles/autoGenProxy"
waitOnCompletion="false">
<InnerTokenMapping LocalProfileToken="jpeg_lres" InnerProfileToken=
"doc_token_targetFormat" Literal="true"/>
<InnerTokenMapping LocalProfileToken="jpeg_lres" InnerProfileToken=
"default_proxy_format" Literal="true"/>
<InnerTokenMapping LocalProfileToken="false" InnerProfileToken=
"overwrite_rendition" Literal="true"/>
<InnerTokenMapping LocalProfileToken="300" InnerProfileToken=
"doc_token_width_proxy" Literal="true"/>
<InnerTokenMapping LocalProfileToken="300" InnerProfileToken=
"doc_token_height_proxy" Literal="true"/>
<InnerTokenMapping LocalProfileToken="Low Resolution Proxy"
 InnerProfileToken="rendition_description" Literal="true"/>
</InnerProfile>
```

4. Save the register file back to the repository.

5. Check the Transformation Services error log file to ensure the profile was loaded without any errors.

6. Test the modified profile to ensure the renditions are being created in the correct size.

4.2.4 Chaining and sequencing profiles

With Transformation Services, it is possible to link existing system or user profiles into a multi-step transformation. The benefit of this type of transformation is that it can save time for end users. If there is a particular series of transformations they perform regularly, you can streamline the process by linking multiple transformations together. Users can then simply request one multi-step transformation, which is more time efficient.

There are two methods of linking user profiles:

- "Chaining" on page 158

  When profiles are chained together, the result of each profile is used as the source for the next profile. The result of the chain is one output file.

- "Sequencing" on page 162

  Sequencing simply invokes multiple transformations on the same source file, and saves the result of each transformation. Each profile transforms the original source file; the second profile does not transform the output of the first profile. If two profiles are sequenced, then two output files will be produced.

Procedures for each method are provided in the following sections.

There is no known limit to the number of profiles that can be chained or sequenced. Ensure that each profile to be chained or sequenced offers support for the required file formats.

Parameters, such as height, width, or angle of rotation, can be handled in two ways:

- Hard-coded in the profile, using the `InnerTokenMapping` tag

- Presented as user-defined parameters, using the `Parameter` tag

See the `Transcodings` and `ProfileSequence` tags in “Understanding the structure of a sequenced transformation” on page 162 for an illustration of these two techniques.

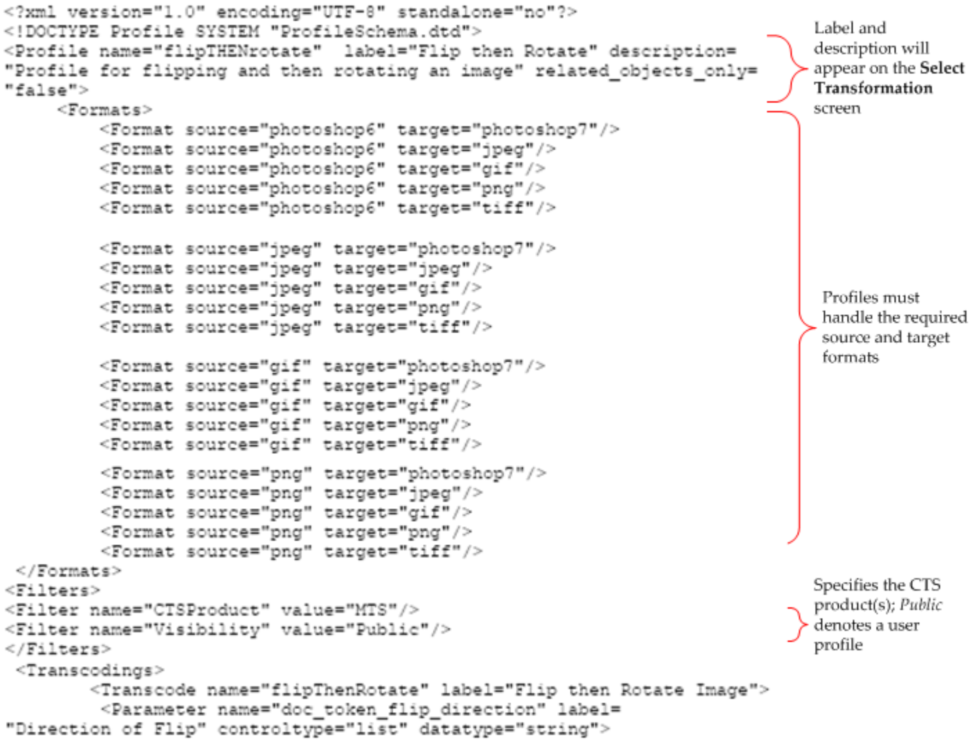
4.2.4.1 Chaining

User profiles are chained together by using the `ProfileChain` tag. After the chained profiles have been executed, the result is one new output file that is derived from the original input file. The result of the chain is then stored in the repository.

“Understanding the structure of a chained transformation” on page 158 shows the components of a chain.

4.2.4.1.1 Understanding the structure of a chained transformation

The following graphic illustrates the coding required for a chained transformation profile.

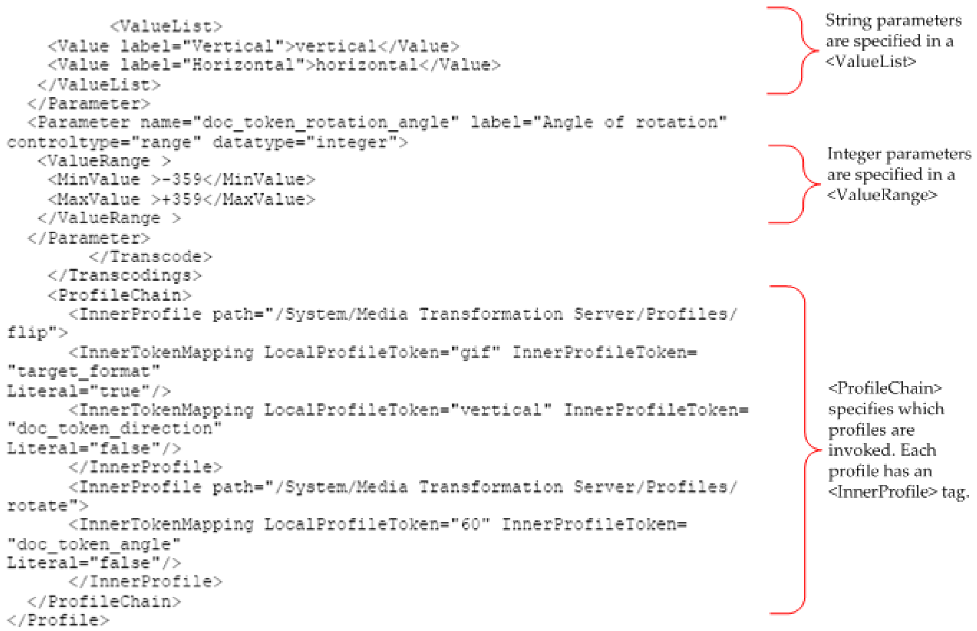


```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE Profile SYSTEM "ProfileSchema.dtd">
<Profile name="flipTHENrotate"  label="Flip then Rotate" description=
"Profile for flipping and then rotating an image" related_objects_only=
"false">
     <Formats>
         <Format source="photoshop6" target="photoshop7"/>
         <Format source="photoshop6" target="jpeg"/>
         <Format source="photoshop6" target="gif"/>
         <Format source="photoshop6" target="png"/>
         <Format source="photoshop6" target="tiff"/>

         <Format source="jpeg" target="photoshop7"/>
         <Format source="jpeg" target="jpeg"/>
         <Format source="jpeg" target="gif"/>
         <Format source="jpeg" target="png"/>
         <Format source="jpeg" target="tiff"/>

         <Format source="gif" target="photoshop7"/>
         <Format source="gif" target="jpeg"/>
         <Format source="gif" target="gif"/>
         <Format source="gif" target="png"/>
         <Format source="gif" target="tiff"/>

         <Format source="png" target="photoshop7"/>
         <Format source="png" target="jpeg"/>
         <Format source="png" target="gif"/>
         <Format source="png" target="png"/>
         <Format source="png" target="tiff"/>
 </Formats>
<Filters>
<Filter name="CTSProduct" value="MTS"/>
<Filter name="Visibility" value="Public"/>
</Filters>
 <Transcodings>
        <Transcode name="flipThenRotate" label="Flip then Rotate Image">
         <Parameter name="doc_token_flip_direction" label=
"Direction of Flip" controltype="list" datatype="string">
```

Label and description will appear on the **Select Transformation** screen

Profiles must handle the required source and target formats

Specifies the CTS product(s); *Public* denotes a user profile



```
          <ValueList>
    <Value label="Vertical">vertical</Value>
    <Value label="Horizontal">horizontal</Value>
   </ValueList>
  </Parameter>
  <Parameter name="doc_token_rotation_angle" label="Angle of rotation"
controltype="range" datatype="integer">
   <ValueRange >
    <MinValue >-359</MinValue>
    <MaxValue >+359</MaxValue>
   </ValueRange >
  </Parameter>
        </Transcode>
    </Transcodings>
    <ProfileChain>
      <InnerProfile path="/System/Media Transformation Server/Profiles/
flip">
      <InnerTokenMapping LocalProfileToken="gif" InnerProfileToken=
"target_format"
Literal="true"/>
      <InnerTokenMapping LocalProfileToken="vertical" InnerProfileToken=
"doc_token_direction"
Literal="false"/>
      </InnerProfile>
      <InnerProfile path="/System/Media Transformation Server/Profiles/
rotate">
      <InnerTokenMapping LocalProfileToken="60" InnerProfileToken=
"doc_token_angle"
Literal="false"/>
      </InnerProfile>
  </ProfileChain>
</Profile>
```

String parameters are specified in a <ValueList>

Integer parameters are specified in a <ValueRange>

<ProfileChain> specifies which profiles are invoked. Each profile has an <InnerProfile> tag.

"Creating a chained transformation" on page 159 provides a step-by-step procedure for building a chained profile.

4.2.4.1.2 Creating a chained transformation

The following procedure illustrates how to chain two profiles. However, you can chain more than two profiles together by following this approach.

To chain user profiles together:

1. Export both profiles from the repository. User profiles are located in the Profiles folder. "Locating the profiles" on page 151 provides more information.

2. Open one of the profiles in an XML or text editor.

   This file will be used to build the new chained profile.

3. Save this profile with a name that identifies the chained profile.

   For example, if you are chaining `flip.xml` and `rotate.xml`, an appropriate name for the chained profile might be `flipthenrotate.xml`.

4. Edit the profile name, label, and description within the `Profile` tag.

   Continuing the example of a flip and rotate chain, the edited `Profile` tag might appear as follows:

```
<Profile name="flipTHENrotate"  label="Flip Then Rotate" description=
"Profile for flipping and then rotating an image" related_objects_only=
"false">
```

Note: Both the label and description will appear in the user interface.

5. Ensure that each profile can handle the formats listed in the <*Format*> listing.

6. Edit the name and label within the `Transcode` tag.

   For example:

```
<Transcodings>
        <Transcode name="flipTHENRotate" label="Flip then Rotate Image">
```

7. Open the other profile.

8. Choose the `Parameter` tag from the second profile; copy and paste this tag into the `Transcodings` section of the chained profile (in the example, this is the `flipthenrotate.xml` file).

   The full `Transcodings` section for a flip and rotate chain would appear as follows:

```
<Transcodings>
        <Transcode name="flipThenRotate" label="Flip then Rotate Image">
            <Parameter name="doc_token_flip_direction" label=
"Direction of Flip" controltype="list" datatype="string">
                <ValueList>
                <Value label="Vertical">vertical</Value>
                <Value label="Horizontal">horizontal</Value>
            </ValueList>
        </Parameter>
        <Parameter name="doc_token_rotation_angle" label="Angle of rotation"
controltype="range" datatype="integer">
            <ValueRange >
                <MinValue >-359</MinValue>
                <MaxValue >+359</MaxValue>
            </ValueRange >
        </Parameter>
        </Transcode>
    </Transcodings>
```

9. After the `Transcodings` section, add the `ProfileChain` section as follows:

```
<ProfileChain>
      <InnerProfile path="/System/Media Server/Profiles/flip">
       <InnerTokenMapping LocalProfileToken="vertical" InnerProfileToken=
"doc_token_direction"
Literal="false"/>
      </InnerProfile>
      <InnerProfile path="/System/Media Server/Profiles/rotate">
      <InnerTokenMapping LocalProfileToken="60" InnerProfileToken=
"doc_token_angle"
Literal="false"/>
      </InnerProfile>
        </ProfileChain>
```

10. Save the new profile.

    For this example, the chained profile (in its entirety) would appear as follows:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE Profile SYSTEM "ProfileSchema.dtd">
<Profile name="flipTHENrotate"  label="Flip then Rotate" description=
"Profile for flipping and then rotating an image" related_objects_only=
"false">
     <Formats>
            <Format source="photoshop6" target="photoshop7"/>
            <Format source="photoshop6" target="jpeg"/>
            <Format source="photoshop6" target="gif"/>
            <Format source="photoshop6" target="png"/>
            <Format source="photoshop6" target="tiff"/>
            <Format source="jpeg" target="photoshop7"/>
            <Format source="jpeg" target="jpeg"/>
            <Format source="jpeg" target="gif"/>
            <Format source="jpeg" target="png"/>
            <Format source="jpeg" target="tiff"/>

            <Format source="gif" target="photoshop7"/>
            <Format source="gif" target="jpeg"/>
            <Format source="gif" target="gif"/>
            <Format source="gif" target="png"/>
            <Format source="gif" target="tiff"/>

            <Format source="png" target="photoshop7"/>
            <Format source="png" target="jpeg"/>
            <Format source="png" target="gif"/>
            <Format source="png" target="png"/>
            <Format source="png" target="tiff"/>
    </Formats>
<Filters>
<Filter name="CTSProduct" value="MTS"/>
<Filter name="Visibility" value="Public"/>
</Filters>
    <Transcodings>
        <Transcode name="flipThenRotate" label="Flip then Rotate Image">
            <Parameter name="doc_token_flip_direction" label=
"Direction of Flip" controltype="list" datatype="string">
                <ValueList>
                <Value label="Vertical">vertical</Value>
                <Value label="Horizontal">horizontal</Value>
            </ValueList>
        </Parameter>
        <Parameter name="doc_token_rotation_angle" label="Angle of
    rotation" controltype="range" datatype="integer">
            <ValueRange >
                <MinValue >-359</MinValue>
                <MaxValue >+359</MaxValue>
            </ValueRange >
        </Parameter>
        </Transcode>
    </Transcodings>
    <ProfileChain>
      <InnerProfile path="/System/Media Server/
      Profiles/flip">
      <InnerTokenMapping LocalProfileToken="vertical"
      InnerProfileToken="doc_token_direction" Literal="false"/>
      </InnerProfile>
      <InnerProfile path="/System/Media Server/
      Profiles/rotate">
      <InnerTokenMapping LocalProfileToken="60" InnerProfileToken=
      "doc_token_angle" Literal="false"/>
      </InnerProfile>
        </ProfileChain>
</Profile>
```

11. Import the new profile to the repository, in the Profiles folder.

The chained profile is now available to users in the WDK interface.

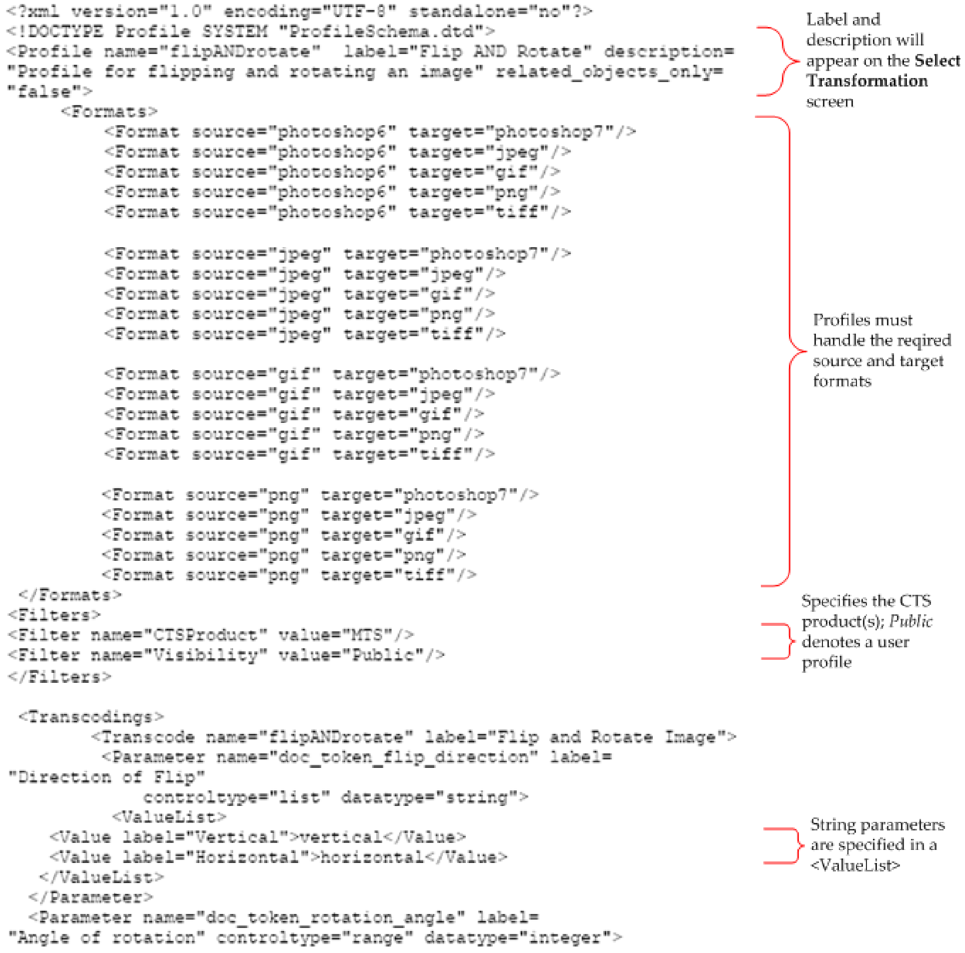
4.2.4.2 Sequencing

The `ProfileSequence` tag allows for a series of transformations to be performed, one after the other. The result of each profile in the sequence is stored in the repository. Therefore, several output files result from one input file.

"Understanding the structure of a sequenced transformation" on page 162 shows the components of a sequence.

4.2.4.2.1 Understanding the structure of a sequenced transformation

The following graphics illustrate the coding required for a sequenced transformation profile:



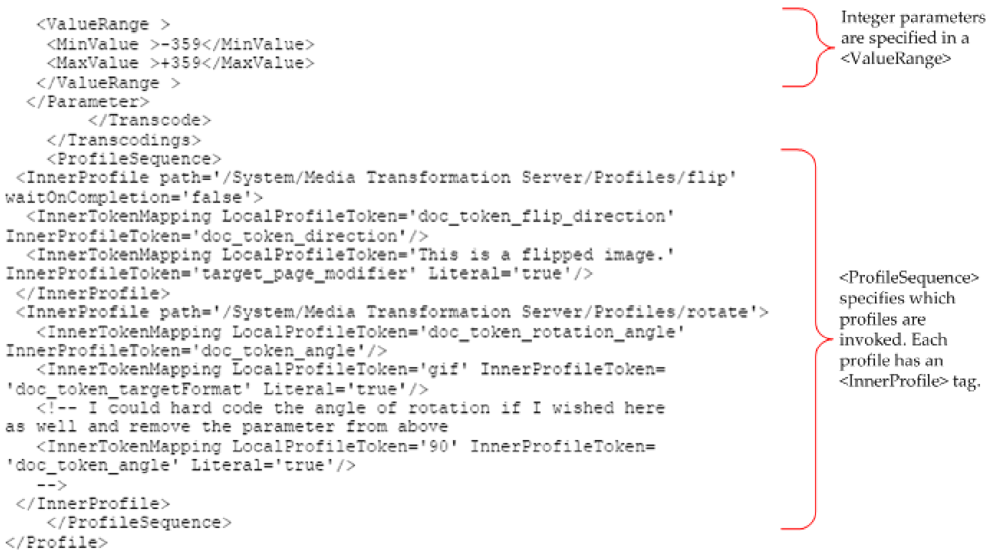
```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE Profile SYSTEM "ProfileSchema.dtd">
<Profile name="flipANDrotate"  label="Flip AND Rotate" description=
"Profile for flipping and rotating an image" related_objects_only=
"false">
     <Formats>
         <Format source="photoshop6" target="photoshop7"/>
         <Format source="photoshop6" target="jpeg"/>
         <Format source="photoshop6" target="gif"/>
         <Format source="photoshop6" target="png"/>
         <Format source="photoshop6" target="tiff"/>

         <Format source="jpeg" target="photoshop7"/>
         <Format source="jpeg" target="jpeg"/>
         <Format source="jpeg" target="gif"/>
         <Format source="jpeg" target="png"/>
         <Format source="jpeg" target="tiff"/>

         <Format source="gif" target="photoshop7"/>
         <Format source="gif" target="jpeg"/>
         <Format source="gif" target="gif"/>
         <Format source="gif" target="png"/>
         <Format source="gif" target="tiff"/>

         <Format source="png" target="photoshop7"/>
         <Format source="png" target="jpeg"/>
         <Format source="png" target="gif"/>
         <Format source="png" target="png"/>
         <Format source="png" target="tiff"/>
 </Formats>
<Filters>
<Filter name="CTSProduct" value="MTS"/>
<Filter name="Visibility" value="Public"/>
</Filters>

 <Transcodings>
        <Transcode name="flipANDrotate" label="Flip and Rotate Image">
         <Parameter name="doc_token_flip_direction" label=
"Direction of Flip"
             controltype="list" datatype="string">
          <ValueList>
    <Value label="Vertical">vertical</Value>
    <Value label="Horizontal">horizontal</Value>
   </ValueList>
  </Parameter>
  <Parameter name="doc_token_rotation_angle" label=
"Angle of rotation" controltype="range" datatype="integer">
```

Label and description will appear on the **Select Transformation** screen

Profiles must handle the reqired source and target formats

Specifies the CTS product(s); *Public* denotes a user profile

String parameters are specified in a <ValueList>



```
    <ValueRange >
     <MinValue >-359</MinValue>
     <MaxValue >+359</MaxValue>
    </ValueRange >
   </Parameter>
         </Transcode>
     </Transcodings>
     <ProfileSequence>
  <InnerProfile path='/System/Media Transformation Server/Profiles/flip'
 waitOnCompletion='false'>
   <InnerTokenMapping LocalProfileToken='doc_token_flip_direction'
 InnerProfileToken='doc_token_direction'/>
   <InnerTokenMapping LocalProfileToken='This is a flipped image.'
 InnerProfileToken='target_page_modifier' Literal='true'/>
  </InnerProfile>
  <InnerProfile path='/System/Media Transformation Server/Profiles/rotate'>
    <InnerTokenMapping LocalProfileToken='doc_token_rotation_angle'
 InnerProfileToken='doc_token_angle'/>
    <InnerTokenMapping LocalProfileToken='gif' InnerProfileToken=
 'doc_token_targetFormat' Literal='true'/>
    <!-- I could hard code the angle of rotation if I wished here
 as well and remove the parameter from above
    <InnerTokenMapping LocalProfileToken='90' InnerProfileToken=
 'doc_token_angle' Literal='true'/>
    -->
  </InnerProfile>
     </ProfileSequence>
 </Profile>
```

Integer parameters are specified in a <ValueRange>

<ProfileSequence> specifies which profiles are invoked. Each profile has an <InnerProfile> tag.

"Creating a sequenced transformation" on page 163 provides a step-by-step procedure for building a sequenced profile.

4.2.4.2.2 Creating a sequenced transformation

The following procedure illustrates how to sequence two profiles. However, you can sequence more than two profiles together by following this approach.

To sequence profiles together:

1. Export both profiles from the repository. User profiles are located in the Profiles folder. "Locating the profiles" on page 151 provides more information.

2. Open one of the profiles in an XML or text editor.

   This file will be used to build the new sequenced profile.

3. Save this profile with a name that identifies the sequenced profile.

   For example, if you are sequencing `flip.xml` and `rotate.xml`, an appropriate name for the sequenced profile might be `flipandrotate.xml`.

4. Edit the profile name, label, and description within the `Profile` tag.

   Continuing the example of a flip and rotate sequence, the edited `Profile` tag might appear as follows:

   ```
   <Profile name="flipANDrotate"  label="Flip And Rotate" description=
   "Profile for flipping and rotating an image" related_objects_only=
   "false">
   ```

   Note: Both the label and description will appear in the user interface.

5. Ensure that each profile can handle the formats listed in the `Format` listing.

6. Edit the name and label within the `Transcodings` tag.

   For example:

```
<Transcodings>
        <Transcode name="flipANDRotate" label="Flip and Rotate Image">
```

7. Open the other profile.

8. Choose the `Parameter` tag from the second profile; copy and paste this tag into the `Transcodings` section of the sequenced profile (in the example, this is the `flipandrotate.xml` file).

   The full `Transcodings` section for a flip and rotate sequence would appear as follows:

```
<Transcodings>
        <Transcode name="flipANDrotate" label="Flip and Rotate Image">
            <Parameter name="doc_token_flip_direction" label=
"Direction of Flip"
             controltype="list" datatype="string">
                <ValueList>
                <Value label="Vertical">vertical</Value>
                <Value label="Horizontal">horizontal</Value>
            </ValueList>
        </Parameter>
        <Parameter name="doc_token_rotation_angle" label=
   "Angle of rotation" controltype="range" datatype="integer">
            <ValueRange >
                <MinValue >-359</MinValue>
                <MaxValue >+359</MaxValue>
            </ValueRange >
        </Parameter>
        </Transcode>
    </Transcodings>
```

9. After the `Transcodings` section, add the `ProfileSequence` section as follows:

```
<ProfileSequence>
     <InnerProfile path='/System/Media Server/Profiles/
  flip' waitOnCompletion='false'>
        <InnerTokenMapping LocalProfileToken='doc_token_flip_direction'
   InnerProfileToken='doc_token_direction'/>
        <InnerTokenMapping LocalProfileToken='This is a flipped image.'
   InnerProfileToken='rendition_description' Literal='true'/>
    </InnerProfile>
    <InnerProfile path='/System/Media Server/Profiles/
  rotate'>
         <InnerTokenMapping LocalProfileToken='doc_token_rotation_angle'
    InnerProfileToken='doc_token_angle'/>
         <InnerTokenMapping LocalProfileToken='gif' InnerProfileToken=
    'doc_token_targetFormat' Literal='true'/>
         <!-- I could hard code the angle of rotation if I wished here
    as well and remove the parameter from above
            <InnerTokenMapping LocalProfileToken='90' InnerProfileToken=
     'doc_token_angle' Literal='true'/>
         -->
    </InnerProfile>
    </ProfileSequence>
```

10. Save the new profile.

    For this example, the sequenced profile (in its entirety) would appear as follows:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE Profile SYSTEM "ProfileSchema.dtd">
<Profile name="flipANDrotate"  label="Flip AND Rotate" description=
"Profile for flipping and rotating an image" related_objects_only=
"false">
     <Formats>
            <Format source="photoshop6" target="photoshop7"/>
            <Format source="photoshop6" target="jpeg"/>
            <Format source="photoshop6" target="gif"/>
            <Format source="photoshop6" target="png"/>
            <Format source="photoshop6" target="tiff"/>

            <Format source="jpeg" target="photoshop7"/>
            <Format source="jpeg" target="jpeg"/>
            <Format source="jpeg" target="gif"/>
            <Format source="jpeg" target="png"/>
            <Format source="jpeg" target="tiff"/>

            <Format source="gif" target="photoshop7"/>
            <Format source="gif" target="jpeg"/>
            <Format source="gif" target="gif"/>
            <Format source="gif" target="png"/>
            <Format source="gif" target="tiff"/>

            <Format source="png" target="photoshop7"/>
            <Format source="png" target="jpeg"/>
            <Format source="png" target="gif"/>
            <Format source="png" target="png"/>
            <Format source="png" target="tiff"/>
    </Formats>
<Filters>
<Filter name="CTSProduct" value="MTS"/>
<Filter name="Visibility" value="Public"/>
</Filters>

    <Transcodings>
        <Transcode name="flipANDrotate" label="Flip and Rotate
Image">
            <Parameter name="doc_token_flip_direction" label=
"Direction of Flip"
             controltype="list" datatype="string">
                <ValueList>
                <Value label="Vertical">vertical</Value>
                <Value label="Horizontal">horizontal</Value>
            </ValueList>
        </Parameter>
        <Parameter name="doc_token_rotation_angle" label=
   "Angle of rotation" controltype="range" datatype="integer">
            <ValueRange >
                <MinValue >-359</MinValue>
                <MaxValue >+359</MaxValue>
            </ValueRange >
        </Parameter>
        </Transcode>
    </Transcodings>
    <ProfileSequence>
    <InnerProfile path='/System/Media Server/Profiles/
  flip' waitOnCompletion='false'>
        <InnerTokenMapping LocalProfileToken='doc_token_flip_direction'
    InnerProfileToken='doc_token_direction'/>
        <InnerTokenMapping LocalProfileToken='This is a flipped image.'
    InnerProfileToken='rendition_description' Literal='true'/>
    </InnerProfile>
    <InnerProfile path='/System/Media Server/Profiles/
  rotate'>
         <InnerTokenMapping LocalProfileToken='doc_token_rotation_angle'
     InnerProfileToken='doc_token_angle'/>
         <InnerTokenMapping LocalProfileToken='gif' InnerProfileToken=
    'doc_token_targetFormat' Literal='true'/>
         <!-- I could hard code the angle of rotation if I wished here
     as well and remove the parameter from above
            <InnerTokenMapping LocalProfileToken='90' InnerProfileToken=
     'doc_token_angle' Literal='true'/>
         -->
    </InnerProfile>
    </ProfileSequence>
</Profile>
```

11. Import the new profile to the repository, in the Profiles folder.

The sequenced profile is now available to users in the rich media enabled interface.

Note: A Sequenced Profile is meant for creating multiple transformation results from a single input file. However, there are circumstances where users want one of the inner profiles (that is part of sequence profile) to use the output of its preceding profile as the source file, rather than the one supplied with original transformation request. To achieve this, users can use the "-source_format" option. This option is supported by Repo-Out requests, and will not work for File-Out requests when used with WebServices clients.

4.2.5 Adding a rendition description to imported files

Users can add rendition descriptions when they request a transformation through a rich media enabled application. This functionality can also be enabled for files that are imported or checked in to a repository.

Note: The following procedure is based on `register_legacy.xml` and can be applied for `register_xcp.xml` and `register_d2.xml`.

To add a rendition description to imported files:

1. Check out the `register_legacy.xml` profile from the System Profiles folder.

2. Open the file in an XML or text editor.

3. Locate the `ProfileSequence` section.

4. Add the following `InnerTokenMapping` tag to the `InnerProfile` section of the process for which you want to add a rendition description:

```
<InnerTokenMapping LocalProfileToken="<rendition description here>"
InnerProfileToken="rendition_description" Literal="true"/>
```

For example, if you are adding the rendition description to the thumbnail `InnerProfile`, this section would appear as follows:

```
<InnerProfile path="/System/Media Server/System Profiles/thumbnail"
waitOnCompletion="false">
<InnerTokenMapping LocalProfileToken="jpeg_th" InnerProfileToken=
"doc_token_targetFormat" Literal="true"/>
<InnerTokenMapping LocalProfileToken="false" InnerProfileToken=
"overwrite_rendition" Literal="true"/>
<InnerTokenMapping LocalProfileToken="thumbnail_import"
InnerProfileToken="rendition_description" Literal="true"/>
</InnerProfile>
```

Note: A unique rendition description can be assigned to any or all *<InnerProfile>* processes.

5. Save and close the `register_legacy.xml` file.

6. Check in the profile to the repository.

A rendition description is assigned to all imported files when they are processed by this system profile.

4.2.6 Changing the order that plug-ins are invoked

Each Transformation Services plug-in is capable of handling various source and target formats. For this reason, some profiles must call on a number of plug-ins to transform their full range of source files.

The <*CommandFilePath*> entries in the profiles list the order in which the plug-ins are polled. The transformation server will poll the plug-ins in this order until the transformation request is completed successfully. If the first plug-in successfully completes the transformation, the remaining plug-ins are not executed.

For illustration purposes, the <*CommandFilePath*> entries from the document_to_pdf profile are provided:

```
<CommandFilePath mptype="DOC4">
/System/Media Server/command line Files/pdf_to_pdf.xml
</CommandFilePath>
<CommandFilePath mptype="DOC3">
/System/Media Server/command line Files/ps_to_pdf.xml
</CommandFilePath>
<CommandFilePath mptype="DOC6">
/System/Media Server/command line Files/document_to_pdf.xml
</CommandFilePath>
<CommandFilePath mptype="IMAGE3">
/System/Media Server/command line Files/transformTo_imw.xml
</CommandFilePath>
```

It is possible to change the order that the plug-ins are invoked by certain profiles. There are a few reasons why you may want to make this modification to a profile. For instance, it may be advantageous to move a plug-in to the top of the order if:

- The majority of files imported to the repository are handled by this plug-in. Performance may be improved by invoking this plug-in first.
- Image files achieve better resolution with this plug-in.
- A failover strategy dictates a particular order.

Note: Generally speaking, all plug-ins that are capable of performing a transformation are included in the <*CommandFilePath*> of the profile. Therefore, it is not advisable to *add* any plug-ins to a profile, because other plug-ins may be incapable of handling this transformation successfully.

"Profiles Installed with Transformation Services products" on page 183 provides information to identify the profiles that involve multiple plug-ins. Follow the procedure to change the order of plug-ins in a profile.

To change the order that plug-ins are invoked by a profile:

1. Locate and check out the profile you wish to modify by following step 1 and step 2 in "Understanding the modification procedure" on page 152.

2. Open the profile in an XML or text editor.
3. Cut and paste the *full <CommandFilePath>* entries within the profile to achieve the desired order.
4. Save the profile.
5. Check in the profile to the repository.

This profile will now poll the plug-ins in the order you have specified.

4.2.7 Modifying profiles for different object types

All files within the Documentum system have an associated object type. The default object type is dm_document; however, any object type that is a subtype of dm_document can be used. The profiles must be modified to accept a different object type.

Object types are specified in the `Filters` tag of a profile. It is important to understand the following behavior with respect to object types:

- If *no* object types are specified in a profile, then only dm_document objects are handled by this profile. This is the default behavior.
- If *only* a custom object type is specified in a profile, then the profile will handle this object type only; files of dm_document object type will not be processed.
- If *both* the custom object type and dm_document object type are specified in a profile, both object types will be processed.

4.2.7.1 Using custom object types

You can create custom object types using an application such as Documentum Composer or Documentum Administrator. These object types can then be assigned to files that are imported to a repository.

By default, all Transformation Services profiles process dm_document object types. However, you can add custom object type to a profile, or restrict a profile to handle files that have a particular object type only. If you are modifying a sequenced profile, all `InnerProfile` tags must be modified to accept custom object types, as well.

Caution
Custom object types must be a subtype of dm_document.

To add a custom object type to a profile:

1. Log in to the repository using an appropriate application (such as Desktop or Digital Asset Manager).
2. Navigate to `/System/Media Server/Profiles/` (for user profiles) or `/System/Media Server/System Profiles/` (for system profiles).

3. Check out the profile you want to modify.

   Note: If the profile files are not immediately visible, choose **Show All Objects and Versions** from the Show list box.

4. Open the file in a text editor.

5. Add *both* dm_document and your custom object type to the `Filters` tag.

   For example, the `Filters` tag appears as follows before any modifications are made:

```
<Filters>
<Filter name="CTSProduct" value="MTS"/>
<Filter name="Visibility" value="System"/>
</Filters>
```

   After adding both dm_document and your custom object type, this section would read:

```
<Filters>
<Filter name="CTSProduct" value="MTS"/>
<Filter name="Visibility" value="System"/>
<Filter name="ObjectType" value="dm_document"/>
<Filter name="ObjectType" value="custom type"/>
</Filters>
```

6. Check in the profile as a new version.

This profile will now handle your custom object type, as well as dm_document files.

To restrict a profile to a custom object type:

1. Log in to the repository using an appropriate client application.

2. Navigate to `/System/Media Server/Profiles` (for user profiles) or `/System/Media Server/System Profiles` (for system profiles).

3. Check out the profile you want to modify.

   Note: If the profile files are not immediately visible, choose **Show All Objects and Versions** from the Show list box.

4. Open the file in a text editor.

5. Add the object type to the `Filters` tag.

   For example, the `Filters` tag appears as follows before any modifications are made:

```
<Filters>
<Filter name="CTSProduct" value="MTS"/>
<Filter name="Visibility" value="Public"/>
</Filters>
```

   After adding your custom object type, this section would read:

```
<Filters>
<Filter name="CTSProduct" value="MTS"/>
<Filter name="Visibility" value="Public"/>
```

```
<Filter name="ObjectType" value="custom type"/>
</Filters>
```

6. Check in the profile as a new version.

7. Restart the Transformation Services for the change to take effect.

This profile will now process only those files that have <*custom type*> as their object type.

4.2.7.2 Configuring the PowerPoint slide object on import

With Transformation Services - Media, PowerPoint slides are dm_document type by default. If you want the slide objects to assume the same type as the parent presentation, this can be configured in the register profile.

Note: The Digital Asset Manager PowerPoint assembly feature is required if you are implementing this procedure.

This procedure is required only if you want the slide objects to assume the same type as the parent presentation; it is *not* required simply because you are using a custom object type for PowerPoint slides.

The following procedure is based on `register_legacy.xml` and can be applied for `register_xcp.xml` and `register_d2.xml`.

To change the object type of slides imported through Transformation Services:

1. Log in to the repository using an appropriate application.

2. Check out the `register_legacy.xml` file.

3. Locate the `InnerProfile` tag that relates to the PowerPoint registration profile.

4. Change the slide_document_type value from:

```
<InnerProfile path="/System/Media Server/System Profiles/
powerpointRegistration" waitOnCompletion="false">
<InnerTokenMapping LocalProfileToken="ppt8" InnerProfileToken=
"doc_token_targetFormat" Literal="true"/>
...
<InnerTokenMapping LocalProfileToken="dm_document" InnerProfileToken=
"slide_document_type" Literal="true"/>
...
</InnerProfile>
```

to:

```
<InnerProfile path="/System/Media Server/System Profiles/
powerpointRegistration" waitOnCompletion="false">
<InnerTokenMapping LocalProfileToken="ppt8" InnerProfileToken=
"doc_token_targetFormat" Literal="true"/>
...
<InnerTokenMapping LocalProfileToken="_parent_type_" InnerProfileToken=
"slide_document_type" Literal="true"/>
...
</InnerProfile>
```

Note: The slide type must be a subtype of dm_document.

5. Save the file.
6. Check in the register file to the repository.

Imported PowerPoint slides will now assume the same object type as the parent presentation.

4.2.8 Configuring formats in PowerPoint assembler

In Transformation Services - Media, the PowerPoint assembler profile accepts the following PowerPoint source formats: ppt8, ppt12, and ppt14. By default, the target format is ppt8. You must configure the profile to create assembly objects in ppt12 and ppt14 formats, or to process a custom format.

To configure the PowerPoint assembler for ppt12 and ppt14 target format:

1. Log in to the repository using an appropriate application (such as Digital Asset Manager).
2. Navigate to `/System/Media Server/System Profiles/`.
3. Check out the ppt_assembler.xml profile.

   Note: If the profile files are not immediately visible, choose **Show All Objects and Versions** from the Show list box.

4. Open the file in a text editor.
5. Change this tag:

```
<Parameter name="doc_token_targetFormat" label="Target Format"
default="ppt8" controltype="text" datatype="string" isRequired="true">
<Value label="Format"/>
</Parameter>
```

   to:

```
<Parameter name="doc_token_targetFormat" label="Target Format"
default="ppt12" controltype="text" datatype="string" isRequired=
"true">
<Value label="Format"/>
</Parameter>
```

   Note: Replace *ppt12* with *ppt14* to configure the PowerPoint assembler for ppt14 target format.

6. Save the file and check it in to the repository.

Transformation Services creates the assembly object in ppt12 and ppt14 format.

Note: This format is static in case a PowerPoint template is not selected in DAM > PPT Assembly. If a PowerPoint template of any format is selected, the PowerPoint created will be of the same format as the selected template.

To configure the PowerPoint assembler to handle a custom format:

1. Log in to the repository using an appropriate client application (such as Digital Asset Manager).

2. Navigate to `/System/Media Server/System Profiles/`.

3. Check out the ppt_assembler.xml profile.

   Note: If the profile files are not immediately visible, choose **Show All Objects and Versions** from the Show list box.

4. Open the file in a text editor.

5. Uncomment the following line, and specify your custom format:

```
<--! <Format source="custom_format" target="custom_format"/> -->
```

6. Change this tag:

```
<Parameter name="doc_token_targetFormat" label="Target Format"
default="ppt8" controltype="text" datatype="string" isRequired=
"true">
<Value label="Format"/>
</Parameter>
```

   to:

```
<Parameter name="doc_token_targetFormat" label="Target Format"
default="custom_format" controltype="text" datatype="string"
 isRequired="true">
<Value label="Format"/>
</Parameter>
```

7. Save the file and check it in to the repository.

The ppt_assembler profile will now handle your custom format.

4.2.9 Modifying the XMP plug-in

The following sections detail how to extract and embed XMP metadata to or from custom object attributes using the XMP Plug-in that is included with Transformation Services - Media and Transformation Services - Documents. You are not limited by the original XMP metadata. Additional fields can be added.

4.2.9.1 Extracting XMP metadata to custom attributes

To extract XMP metadata to a custom object attribute, add or modify entries in the command line file `extractMetadataToObject.xml` located in `/System/Media Server/command line Files`. If you choose to create your own custom profile and command line files then you must ensure that the profile name begins with `extractMetadataToObject` and has the taskImpl set to com.documentum.cts.impl.services.task.metadata.MetadataProcessor. In the clf you must set the doctype to the name of the custom object and the name to the name of the attribute. For example, if you wanted to extract the subject to the desc attribute of a custom object named cch_photo then you would modify the line:

```
<map>
<object_attr name="subject" doctype="dm_document" datatype="string" isrepeating="false"
 token="doc_token_subject"/>
<withValueAt namespace="http://purl.org/dc/elements/1.1/" path="*/RDF/Description/
description/Alt/li/text()"/>
</map>
```

to:

```
<map>
<object_attr name="desc" doctype="cch_photo" datatype="string" isrepeating="false"
 token="doc_token_subject"/>
<withValueAt namespace="http://purl.org/dc/elements/1.1/" path="*/RDF/Description/
description/Alt/li/text()"/>
</map>
```

To extract an XMP property that is not already specified in the existing command line file, you must add an entry. In this case, specify the correct XMP namespace in the namespace element and the correct path in the path element. The token must be unique and should start with doc_token. For example, you should not have more than one entry that has the token set to doc_token_subject. In addition, if the custom attribute is a repeating attribute then "isrepeating" should be set to true. The datatype must be set to the type of the custom attribute (possible values are string, integer, boolean, and date).

4.2.9.2 Embedding XMP metadata from custom attributes to the content

To embed XMP metadata from custom object attributes, add or modify entries in the command line file `embedMetadataToContent.xml` located in `/System/Media Server/command line Files`. If you choose to create your own custom profile and command line files, ensure that the profile name begins with `embedMetadataToContent` and has the taskImpl set to com.documentum.cts.impl.services.task.metadata.MetadataProcessor. In the clf between the RETRIEVE_METADATA tags, specify what attributes to extract the metadata from by specifying the value as the custom object name, followed by a period followed by the name of the attribute. If the attribute is a repeating attribute, then you need to add [,] immediately after (cch_photo.desc[,]). Following the example, if you want to embed the desc attribute of the custom object cch_photo, then you would change the following line:

```
<MetadataMapper name="Subject" value="subject" token="doc_metadata_subject"/>
```

to:

```
<MetadataMapper name="Subject" value="cch_photo.desc" token="doc_metadata_subject"/>
```

To add an entirely new entry, add a line under RETRIEVE_METADATA, using a unique token name beginning with doc_token. Then, add a corresponding line in the `EMBED_METADATA` section containing the XMP namespace as the namespace value and the name of the XMP property as the path. The value will be the same as the value that was used previously for the token in the `RETRIEVE_METADATA` section.

4.2.10 Modifying audio and video renditions

For Transformation Services - Audio/Video, there is an additional register file: `video_registration.xml` (or `video_registration_xcp.xml`, depending on the client Transformation Services - Audio/Video is being used for). This file registers formats that are processed exclusively by the Video1 plug-in.

4.2.11 Creating a PDF rendition with Bates stamping on import

The assumption is formats are rich-media enabled and already added to the registration profile. This example provides a sample of how you can automate Bates stamping in Transformation Services. It shows how to link the parameter doc_token_prefix (the Bates stamp Prefix) of the headers_footers_bates_adts profile to a value provided using DQL – such an attribute of a custom type. Some of the parameters from headers_footers_bates_adts profile are linked in the registration profile using inner token mappings. Customers should adjust values based on their business case.

Note: The following procedure is based on `register_legacy.xml` and can be applied for `register_xcp.xml` and `register_d2.xml`.

To create a Bates stamped PDF rendition on import:

1. In the repository, browse to `System/Media Server/System Profiles/` and check out `register_legacy.xml`.

2. Open `register_legacy.xml` in an XML or text editor.

3. Include the Bates stamping profile by adding the following tags to the `ProfileSequence` section:

```
<InnerProfile path="/System/Media Server/Profiles/headers_footers_bates_cts"
 waitOnCompletion="true" useTargetFormat="true">
<InnerTokenMapping LocalProfileToken="pdf"
InnerProfileToken="doc_token_targetFormat"
 Literal="true"/>
<InnerTokenMapping LocalProfileToken="false" InnerProfileToken="overwrite_rendition"
 Literal="true"/>

<InnerTokenMapping LocalProfileToken="true" InnerProfileToken="doc_token_headers"
 Literal="true"/>
<InnerTokenMapping LocalProfileToken="Helvetica" InnerProfileToken="doc_token_h_
fontName" Literal="true"/>
<InnerTokenMapping LocalProfileToken="10" InnerProfileToken="doc_token_h_fontSize"
 Literal="true"/>
<InnerTokenMapping LocalProfileToken="#000000" InnerProfileToken="doc_token_h_color"
 Literal="true"/>
<InnerTokenMapping LocalProfileToken="Foreground"
InnerProfileToken="doc_token_h_layer"
 Literal="true"/>
<InnerTokenMapping LocalProfileToken="Bates Stamp - &amp;[PageID]"
InnerProfileToken="doc_
token_h_textLeft" Literal="true"/>
<InnerTokenMapping LocalProfileToken="All" InnerProfileToken="doc_token_h_pages"
 Literal="true"/>
<InnerTokenMapping LocalProfileToken="MergedDocument” InnerProfileToken="doc_token
_pagenumbering_mode" Literal="true"/>

<InnerTokenMapping LocalProfileToken="false" InnerProfileToken="doc_token_footers"
```

```
 Literal="true"/>
<InnerTokenMapping LocalProfileToken="Helvetica" InnerProfileToken="doc_token_f
_fontName" Literal="true"/>
<InnerTokenMapping LocalProfileToken="10" InnerProfileToken="doc_token_f_fontSize"
 Literal="true"/>
<InnerTokenMapping LocalProfileToken="#000000" InnerProfileToken="doc_token_f_color"
 Literal="true"/>
<InnerTokenMapping LocalProfileToken="Foreground" InnerProfileToken="doc_token_f
_layer" Literal="true"/>
<InnerTokenMapping LocalProfileToken="All" InnerProfileToken="doc_token_f_pages"
 Literal="true"/>

<!--bates settings-->
<InnerTokenMapping LocalProfileToken="doc_token_prefix" InnerProfileToken="doc_token
_prefix" Literal="false"/>
<InnerTokenMapping LocalProfileToken="" InnerProfileToken="doc_token_suffix"
Literal=
"true"/>
<InnerTokenMapping LocalProfileToken="001" InnerProfileToken="doc_token_b_index"
Literal=
"true"/>


<!-- common settings -->
<InnerTokenMapping LocalProfileToken="Yes"
InnerProfileToken="doc_token_changeDefault
Settings" Literal="true"/>

<InnerTokenMapping LocalProfileToken="Yes"
InnerProfileToken="doc_token_enableBookMarks"
 Literal="true"/>
<InnerTokenMapping LocalProfileToken="Yes"
InnerProfileToken="doc_token_enableHyperlinks"
 Literal="true"/>
<InnerTokenMapping LocalProfileToken="PDFVersion15" InnerProfileToken="doc_token
_pdfVersion" Literal="true"/>
<InnerTokenMapping LocalProfileToken="600" InnerProfileToken="doc_token_resolution"
 Literal="true"/>
<InnerTokenMapping LocalProfileToken="Yes" InnerProfileToken="doc_token_optimize"
 Literal="true"/>
<!-- security settings -->
<InnerTokenMapping LocalProfileToken="No"
InnerProfileToken="doc_token_enableSecurity"
 Literal="true"/>
<InnerTokenMapping LocalProfileToken="40bit"
InnerProfileToken="doc_token_encryptionMode"
 Literal="true"/>
<InnerTokenMapping LocalProfileToken="Disabled"
InnerProfileToken="doc_token_changesAllowed"
 Literal="true"/>
<InnerTokenMapping LocalProfileToken="Disabled"
InnerProfileToken="doc_token_enableAccess"
 Literal="true"/>
<InnerTokenMapping LocalProfileToken="Disabled"
InnerProfileToken="doc_token_docAssembly"
 Literal="true"/>
<InnerTokenMapping LocalProfileToken="Disabled"
InnerProfileToken="doc_token_formField
Filling" Literal="true"/>
<InnerTokenMapping LocalProfileToken="Disabled"
InnerProfileToken="doc_token_printing"
 Literal="true"/>
<InnerTokenMapping LocalProfileToken="Disabled"
InnerProfileToken="doc_token_allowCopy"
 Literal="true"/>
<InnerTokenMapping LocalProfileToken=" " InnerProfileToken="doc_token_secOpass"
 Literal="true"/>
<InnerTokenMapping LocalProfileToken=" " InnerProfileToken="doc_token_secCpass"
 Literal="true"/>
<InnerTokenMapping LocalProfileToken="DocumentContent" InnerProfileToken="doc_token
_PrintType" Literal="true"/>
```

```
<InnerTokenMapping LocalProfileToken="No" InnerProfileToken="doc_token_NoteIcons"
 Literal="true"/>

</InnerProfile>
```

4. Add the following parameter to the profile (as the child of the <*Transcode*> element):

```
<Transcodings>
  <Transcode name="register" label="Register Profile">
    <Parameter name="doc_token_prefix" label="Page Identification - Prefix
 Characters" controltype="text" datatype="string" dql="select stamp_prefix from
 dm_custom_type where r_object_id = [sysobject_id_token]" isRequired="true">
      <Value label="text"/>
    </Parameter>
  </Transcode>
</Transcodings>
```

Note: This parameter will help pass to headers_footers_bates_cts the value selected with the DQL. The link is obtained using the inner token mapping from previous point.

5. Check the file into the repository.

6. Restart the Transformation Services.

4.2.12 Testing a modified profile

After modifying a profile, check the Transformation Services error log for errors. When a profile is loaded, Transformation Services verifies that:

- The schema of the profile is valid with respect to the DTD file.
- The paths specified in the `InnerProfile` tags are valid.

For example, if the watermark and register profiles are successfully imported, the following lines appear in the log file:

```
10:14:43,437  INFO [  Thread-5] CTSProfileManagerImpl -
Successfully updated the profile cache for : watermark ...
10:28:12,984  INFO [  Thread-5] CTSProfileManagerImpl -
Updating the profile : register
10:28:12,984  INFO [  Thread-5] CTSProfileManagerImpl -
Inner profile : /System/Media Server/Profiles/watermark ...
```

Successful loading of a profile does not indicate whether the profile will perform the function you intended it to perform. To verify its functionality, use the modified profile to perform a transformation and examine the results. If the results are not what you expected, you will need to make further modifications.

4.3 Using DQL in Profiles

Document Query Language (DQL) can be used with Transformation Services profile files. Through DQL, default profile parameters can be supplied. For example, header text can be pre-populated with the client's name. This text will appear on the Transformation Details screen (of a rich media enabled client applications such as Webtop) when a user performs an **Add header and/or footer** transformation with Transformation Services.

The DQL is presented as an attribute of the Parameter node of profiles (named dql).

4.3.1 The dql attribute value syntax

The syntax for the dql profile parameter attribute value does not differ from a regular DQL statement. It supports the following tokens (placeholders) which are recognized by the system:

- [sysobject_id_token]: placeholder for a Sys Object ID (the r_object_id attribute for dm_sysobject)
- [format_token]: placeholder for the rendition's format (string)
- [page_token]: placeholder for the page attribute of dmr_content (integer)
- [page_modifier_token]: placeholder for the page_modifier attribute of dmr_content (string)

These tokens are replaced at runtime with the actual values taken from the source sys object that is used to execute the profile, before executing the actual query.

4.3.2 DQL guidelines

Consider the following guidelines when supplying DQL for profiles:

- The system does not try to correct a wrong query. It always assumes the query is correctly supplied in profile. The system replaces the contained placeholders with runtime values, and then executes it against the repository.
- Upon executing the query, the system will just consider the first row returned in the result set. If the query is set to return multiple records, just the first one will be considered.
- All the attribute values returned will be formatted accordingly, based on the attribute type (such as date, string, or Boolean).

4.3.3 DQL query examples

Complex queries can be created with DQL. Queries can include all four supported tokens, as many times as they are needed.

The following examples illustrate how to supply the DQL for the dql profile parameter attribute:

```
<Parameter name=”doc_token_text” label=”Object name” controltype=
”text”
datatype=”string” dql=”SELECT object_name from dm_document WHERE
r_object_id = [sysobject_id_token]”>
<Value label=”text”/>
</Parameter>
```

```
<Parameter name=”doc_token_text” label=”Formatted name” controltype=
”text”
datatype=”string” dql=”SELECT 'Name:' as label, object_name from
dm_document WHERE r_object_id = [sysobject_id_token]”>
<Value label=”text”/>
</Parameter>
```

```
<Parameter name=”doc_token_text” label=”Modified date” controltype=
”text”
datatype=”string” dql=”SELECT 'Modified:' as label, r_modify_date from
dm_document WHERE r_object_id = [sysobject_id_token]”>
<Value label=”text”/>
</Parameter>
```

The following query can be used for getting the current lifecycle state for a sys object, as string:

```
<Parameter name=”doc_token_text” label=”Lifecycle state” controltype=
”text”
datatype=”string” dql=”SELECT state_name FROM dm_policy a, dm_sysobject b
WHERE a.r_object_id = b.r_policy_id AND a.i_state_no = b.r_current_state
AND b.r_object_id = [sysobject_id_token] ENABLE(row_based)”>
<Value label=”text”/>
</Parameter>
```

For getting information that is pertinent to the user who executes the profile, the user keyword can be used in query:

```
<Parameter name=”doc_token_text” label=”User name” controltype=”text”
datatype=”string” dql=”SELECT user_name, user_address FROM dm_user
WHERE user_name = USER”>
<Value label=”text”/>
</Parameter>
```

4.3.4 Querying the r_object_id attribute

Having the r_object_id attribute as part of the SELECT clause of the query will not result in having the actual Object ID value supplied as part of the default parameter value. This attribute sometimes has a special (system) role in query (for example, properly returning the repeating attributes in a single row). The system cannot know its real purpose at runtime, and therefore does not consider it when building the profile parameter default value.

To have the Object ID value supplied as part of the profile default parameter value, syntax must be provided that uses the [sysobject_id_token] placeholder on the query's SELECT clause, as follows:

```
<Parameter name=”doc_token_text” label=”New Text” controltype=”text”
datatype=”string” dql=”SELECT [sysobject_id_token] as objectid from
dm_document WHERE r_object_id = [sysobject_id_token]”>
<Value label=”text”/>
</Parameter>
```

4.4 Troubleshooting

This section lists some of the problems or concerns that system administrators or developers may encounter when modifying profile files.

4.4.1 Why are there no XML profiles appearing in my profile folders?

When you initially open the Profiles folder or the System Profiles folder, you may see only the DTD files. To resolve this issue, select **Show All Objects and Versions** from the Show list dialog box. The profiles appear, and can be checked out from the repository as required.

4.4.2 Why is my modified user profile not appearing in the interface?

User profiles must be tagged as *Public* to appear in the client interface. In contrast, system profiles are tagged as *System* and are not exposed to users.

If you have modified a user profile in some way (or chained/sequenced user profiles to create an additional transformation option) and the transformation does not appear in the interface, ensure that the profile is properly tagged. Look for the `Filter name` tag, which typically appears after the `Formats` section. For example, a user profile includes the following tags:

```
<Filter name="CTSProduct" value="ADTS"/>
<Filter name="Visibility" value="Public"/>
```

Both system and user profiles must contain two `Filter name` tags: one to specify whether it is a system (“System”) or user (“Public”) profile, and the other to specify the product.

Caution

The value of <*Documentum Content Transformation Services product*> must ONLY be that product that is configured to the repository, which means that, it can be any one of the following values: MTS, ADTS, XTS, and AVTS. If Transformation Services - Media is configured and the value of <*Documentum Content Transformation Services product initialism*> is set to "AVTS", then this profile will not be executed by the Transformation Services - Media server.

4.4.3 How do I "uncomment" text in XML files?

Typically, programmers add comments to an XML file to explain the purpose of a section of the coding. For example, in the register file for Transformation Services, this comment introduces a section of coding:

```
<!-- This will generate thumbnail, proxy and storyboards for pdf formats -->
```

The syntax for comments is:

```
<!-- comment -->
```

In XML, any coding that is contained within the comment syntax is ignored by the application during processing. Besides adding notations to the file, the comment syntax can also be used to "hide" processing instructions. For example, in the `legacy_to_pdf` system profile, the storing of PostScript renditions is not enabled out-of-the-box. The coding that pertains to PostScript renditions is commented out:

```
<!-- uncomment the below section to enable the storing of ps renditions -->
<!--
<InnerProfile path="/System/Media Server/System Profiles/document_to_ps"
waitOnCompletion="false" useTargetFormat="true">
     <InnerTokenMapping LocalProfileToken="ps" InnerProfileToken=
     "doc_token_targetFormat" Literal="true" />
     <InnerTokenMapping LocalProfileToken="ps rendition" InnerProfileToken=
     "rendition_description" Literal="true" />
     <InnerTokenMapping LocalProfileToken="true" InnerProfileToken=
     "overwrite_rendition" Literal="true" />
     <InnerTokenMapping LocalProfileToken="legacy" InnerProfileToken=
     "transformation_type" Literal="true" />
     <InnerTokenMapping LocalProfileToken="postscript" InnerProfileToken=
     "rendition_description" Literal="true" />
<\InnerProfile>-->
```

However, by uncommenting – or removing the comment syntax – around this section, PostScript renditions *will* be stored. The coding would then appear as follows:

```
<!-- uncomment the below section to enable the storing of ps renditions -->
<InnerProfile path="/System/Media Server/System Profiles/document_to_ps"
waitOnCompletion="false" useTargetFormat="true">
     <InnerTokenMapping LocalProfileToken="ps" InnerProfileToken=
     "doc_token_targetFormat" Literal="true" />
     <InnerTokenMapping LocalProfileToken="ps rendition" InnerProfileToken=
     "rendition_description" Literal="true" />
     <InnerTokenMapping LocalProfileToken="true" InnerProfileToken=
     "overwrite_rendition" Literal="true" />
     <InnerTokenMapping LocalProfileToken="legacy" InnerProfileToken=
     "transformation_type" Literal="true" />
     <InnerTokenMapping LocalProfileToken="postscript" InnerProfileToken=
     "rendition_description" Literal="true" />
<\InnerProfile>
```

Thus, *commenting* is a coding device that disables processing within the profile, whereas *uncommenting* enables the processing.

To comment out a portion of a profile:

1. Check out the profile from the repository.
2. Open the file in an XML or text editor.
3. Locate the portion of the profile for which you want to disable the processing instructions.
4. Enclose the tags within the commenting syntax:

   ```
   <!-- <processing tags> -->
   ```

5. Save the file.
6. Check in the profile to the repository.
7. Restart the Transformation Services.

The tags you have commented out will now be ignored during processing.

To uncomment a portion of a profile:

1. Check out the profile from the repository.
2. Open the file in an XML or text editor.
3. Locate the portion of the profile that has been uncommented.
4. Remove the commenting syntax (`<!--` and `-->`) from the file.
5. Save the file.
6. Check in the profile to the repository.
7. Restart the Transformation Services.

The tags you have uncommented will now be included in processing.

4.4.4 Modified profiles did not load successfully

If you modified a profile and the Transformation Services error log indicates that they did not load successfully, check the following:

- Verify the modified profile against the Transformation Services DTD file. If the profile does not conform to the schema, it will not load.
- Verify that correct paths are listed in the `InnerProfile` tags of the profile. If you changed the file path of the profile folders in the repository, you must change the <*InnerProfile*> paths accordingly.

4.4.5 Using a profile description bigger than 192 bytes

The profile description length for new or edited profiles is up to 192 bytes. Note that Russian characters can be either 2 or 3 bytes. To use a description that exceeds 192 characters, perform the following procedure:

1. Run the following DQL queries against the repository:

   ```
   EXECUTE ALLOW_BASE_TYPE_CHANGES WITH ALLOW_CHANGE_FLAG=1
   ```

2. Alter type dm_sysobject modify subject char(300):

   ```
   EXECUTE ALLOW_BASE_TYPE_CHANGES WITH ALLOW_CHANGE_FLAG=0
   ```

3. For Second successive run we need to change it to (301):

   ```
   EXECUTE ALLOW_BASE_TYPE_CHANGES WITH ALLOW_CHANGE_FLAG=1
   ```

4. Alter type dm_sysobject modify subject char(301). (The reason for providing the size as 301 is, the size specified for the subject attribute should be greater than the previous value, else server throws an error)

   ```
   EXECUTE ALLOW_BASE_TYPE_CHANGES WITH ALLOW_CHANGE_FLAG=0
   ```

Caution

Use caution when running DQLs against the repository as any input errors may cause damage.

4.4.6 Restart the Transformation Services if profiles are modified

After modifying a system profile (such as register), the Transformation Services product must be restarted because the system profiles are cached during startup.

Restart the Transformation Services product using the Windows Services Utility, found at **Start > Settings > Control Panel > Administrative Tools > Services**.

4.5 Profiles Installed with Transformation Services products

This section lists both system and user profiles and provides the following information:

- Information about the plug-in(s) that are handling each profile. Where more than one plug-in is invoked by a single profile, you can change the order that the plug-ins are polled (except in the case of a chain profile).
- If you are adding a format to a Transformation Services, you must know whether the relevant plug-in is capable of handling the new format. Use this section to determine the plug-in that relates to the functionality desired for the new format.
- Descriptions of what each profile does. This information can be useful when profiles require modification, because you can determine which profile should be modified in order to effect a particular change.

The Transformation Services products are shipped with a number of predefined system and user profiles. System profiles are categorized based on the Transformation Services product.

To locate a profile by name, consult the index. All profile files are listed alphabetically in the index.

4.5.1 Transformation Services - Documents profiles

Table 4-1: Transformation Services - Documents profiles

Profile name	Action	Plug-in name
System profiles:		
addSignature_to_doc	Converts the provided document to PDF and adds a pre-defined or user-defined signature page to the output.	Doc4
attachment_to_pdf	Converts MSG attachments to PDF.	Doc4, Doc3, Doc6, Doc7, Image3
attachment_pdf	Adds specified content as attachment to the PDF.	Doc4

Profile name	Action	Plug-in name
autoGenProxy_msg	Generates a proxy.	
autoGenPreviewProxy_msg	Generates a preview proxy.	
content_to_pdf	Converts multiple documents to PDF.	Doc4, Doc3, Doc6, Doc7, Image3
document_extract_props	Extracts properties of Word, Excel, PDF, and PowerPoint source files.	Doc6, PDFStoryboard
document_registration	Registers source files; invokes convert_to_pdf and pdf_processing system profiles.	
document_registration_d2	Registers documents for client.	
document_registration_xcp	Registers documents for xCP.	
document_to_custom_pdf	Converts documents to PDF.	Doc4, Doc3, Doc6, Image3, Doc11 (optional)
document_to_html	Transforms source files to HTML format. **Note:** This type of transformation is available only for the following source formats: crtext, excel8book, excel8template, excel12book, excel12template, excel14book, excel14template, msw8, msw8template, msw12, msw12template, msw14, msw14template, and rtf.	Doc6, Doc9
document_to_html_ppt	Converts PPT and PPTX documents to HTML.	Doc6 > Doc9

Profile name	Action	Plug-in name
document_to_pdf	Transforms source files to PDF format. **Notes** • Following are the source formats: crtext, excel8book, excel8template, excel12mebook, excel12metemplate, excel12bbook, excel12book, excel12template, excel14book, excel14template, html, msw8, msw8template, msw12, msw12template, msw14, msw14template, msw12me, msw12metemplate, ppt8, ppt8_slide, ppt8slideshow, ppt8_template, ppt12, ppt12_slide, ppt12slideshow, ppt12template, ppt14, ppt14_slide, ppt14slideshow, ppt14template, rtf, odt, ott, ods, ps, eps, jpeg, gif, png, tiff, photoshop6, photoshop7, photoshop8, cr2, crw, wmf, bmp, mpp, vsdx, vsd, msg/ email, and pdf. • The Doc4 plug-in does not support TIFF file. • To support TIFF file, Ghostscript installation is mandatory (in the Image3 plug-in).	Doc4, Doc3, Doc6, Image3, Doc11 (optional)
document_to_ps	Transforms source files to PostScript format.	Doc6 > Doc8
document_to_xCP_pdf	Transforms source files to PDF for xCP.	Doc3, Doc6
headers_footers_bates_to_doc	Converts the provided document to PDF and adds the specified header, footer and bates stamping to the output.	Doc4
headers_footers_to_doc	Converts the provided document to PDF and adds the specified header and footer to the output.	Doc4
legacy_to_html	Transforms legacy source files to HTML. Invokes document_to_html system profile.	

Profile name	Action	Plug-in name
legacy_to_pdf	Transforms legacy source files to PDF. Invokes the document_to_pdf system profile. Can be configured to store PDF text files, PDF metadata, and PostScript renditions.	
merge_all_pdf	Merges several PDF documents to one PDF.	Doc4
msg_registration	Registers emails.	
msg_registration_d2	Registers emails for client.	
msg_registration_xcp	Registers emails for xCP.	
optimizePDF	Converts the provided document to a web-optimized PDF.	Doc6, Doc11
overlay_to_doc	Converts the provided document to PDF and adds an overlay using the specified image.	Doc4
pdf_processing	Invokes thumbnail, low-resolution rendition, and storyboard system profiles for PDF source files. Invokes a number of other system profiles.	
pdf_processing_d2	Processes PDFs for client.	
pdf_processing_xcp	Processes PDFs for xCP.	
pdf_to_pdf	Converts PDF to PDF. **Note:** The Doc3 plug-in does not generate PDF-to-PDF renditions if the input PDF has **Security Method** restrictions enabled or contains a digital signature.	Doc3, Doc4
register	Registers supported source files when they are imported to the repository. Invokes thumbnail, low-resolution, and storyboard renditions, and property extraction system profiles.	
register_d2	Automatically processes imported files from the client.	
register_legacy	Automatically processes imported files, replicating the behavior of Transformation Services in the previous releases.	
register_mobile	Automatically process imported files for Mobile client.	

Profile name	Action	Plug-in name
register_xcp	Automatically processes imported files from the xCP client.	
registerEx	Automatically processes imported files.	
reorder_registration	Reorders pages and registers again.	
storyboard	Generates a storyboard for PDF source files.	Image3
storyboard_msg	Generates a storyboard for messages.	PDFStoryboard, Doc7
storyboard_pdfstoryboard	Generates a storyboard for PDF source files. Image size is 200 x 200.	PDFStoryboard, Image3
storyboard_tiffstoryboard	Generates a storyboard for TIFF files.	Image3
thumbnail_msg	Generates a thumbnail for messages.	PDFStoryboard, Doc7
thumbnail_pdfstoryboard	Generates thumbnails of PDF source files only.	PDFStoryboard, Image3
tiff_registration	Registers TIFF files.	
tiff_registration_d2	Registers TIFF files for client.	
tiff_registration_xcp	Registers TIFF files for xCP.	
to_PDFA_doc	Converts the provided document to a PDF/A compliant PDF file with the specified options.	Doc4
to_PDFA_doc3	Converts the provided document to a PDF/A compliant PDF file with the specified options.	Doc3
ps_to_pdf	Converts a PostScript file to a PDF file. This profile supports password-protected documents.	Doc3
to_PDFX_doc.xml	Converts the provided document to a PDF/X compliant PDF file with the specified options.	Doc4
to_PDFX_doc3	Converts the provided document to a PDF/X compliant PDF file with the specified options.	Doc3
transform_mime	Transforms EML and MSG source files to rich text format.	Doc7
transform_msg	Converts MSG to PDF and extracts the attachments.	Doc7
transformMsgToXML	Converts messages to XML renditions. This profile serves real-time requests.	Doc7, Doc2

Profile name	Action	Plug-in name
transformXcpDocumentDirect	Creates PPT/PPTX and DOC/DOCX storyboards.	Doc6
transformXcpDocument	Converts to PDF format and create storyboards. **Note:** The conversion process can be carried out for a specific page range.	
transformXcpDocumentToXML	Creates XML renditions from documents. This profile serves real time requests. **Note:** The conversion process can be carried out for a specific page range.	
transformXcpPdf	Generates storyboards and low resolution JPEG files for specific page(s).	PDFStroryboard, Image3
transformXcpPdfToXML	Creates XML renditions from PDF. This profile serves real time requests. **Note:** The conversion process can be carried out for a specific page range.	Doc6, Doc2
transformXcpTiff	Converts TIFF files to PDF and creates storyboards.	PDFStoryboard, Image3
watermark_to_doc	Converts the provided document to PDF with a textual watermark.	Doc4
User profiles:		
addSignature_adts	Converts to PDF, and allows the user to add a signature page to the output. Advanced PDF options, such as security is available to users.	Doc6 > Doc4
attachment_adts	Converts to PDF, and allows the user to embed attachments to their PDF files. Advanced PDF options, such as security is available to users.	Doc6 > Doc4
embedMetadataToContent_adts	Writes object properties (attributes) to content.	Doc4, Doc6
extractMetadataToObject_adts	Writes content properties to object properties (attributes).	XMP, Doc6
extractMIME	Extracts MIME body and attachments from EML and MSG source files, and transforms files to CRTEXT/RTF format.	Doc7

Profile name	Action	Plug-in name
headers_footers_adts	Allows the user to add header and/or footer information and transforms files to PDF format.	Doc6 > Doc4
headers_footers_bates_adts	Converts the provided document to PDF and adds header and/or footer.	Doc6 > Doc4
markupPDF_adts	Converts the selected Microsoft Word (2003/2007) document to PDF with markup (comments and tracked changes such as insertions, deletions, and formatting changes).	Doc6 > Doc4
merge_virtualdoc_adts	Creates a merged PDF from a virtual document structure. Advanced PDF options, such as security is available to users.	Doc6 > Doc4
mergeCollection_adts	Merges files in a collection into a single PDF file. Advanced PDF options, such as security is available to users.	Doc6, Doc4
mergePDF_adts	Appends additional files (to a maximum of five) to the source file. Advanced PDF options, such as security is available to users.	Doc6 > Doc4
mergeRenditions_adts	Merges the specified rendition type into a single PDF file. Advanced PDF options, such as security is available to users.	Doc6
mime_to_PDFA	A chain profile, which converts the provided MIME document to a PDF/A compliant PDF file with the specified options. This profile supports files in the EML and MSG formats.	Doc 7 > Doc4 > Doc 3
optimizePDF_adts	Converts the provided document to a web-optimized PDF.	Doc6
overlay_adts	Creates a PDF and applies an image file as an overlay. Users can specify the alignment and file path of the overlay, and select advanced PDF options.	Doc6 > Doc4
pdf_split_adts	Converts the provided document to multiple PDF files.	Doc4
PDF_to_PDFA_adts	Converts PDF documents to PDFA-1a, PDFA-1b, PDFA-2a, PDFA-2b, PDFA-3a, or PDFA-3b and also allows web optimization option.	Doc6
pdf_to_ps	Converts PDF to Postscript.	Doc8

Profile name	Action	Plug-in name
ppt_conversion	Converts PowerPoint files.	PowerPoint1
reorderPages	Reorders the pages.	Doc5
to_PDFA_adts	Creates a PDF/A compliant PDF file, with user-defined options.	Doc6 > Doc3
to_PDFX_adts	Creates a PDF/X compliant PDF file, with user-defined options.	Doc6 > Doc3
transformMIMEToPDF	Transforms EML and MSG source files to PDF. Invokes transformEmlToPDF and transformMsgToPDF system profiles.	Doc7
transformMSGToPDFWithAttachment	Converts MSG/EML files to PDF with attachments.	Doc7
transformToText	Converts PDF files to Text format.	Doc2
transformToXMLInfo	Obtains the page-range (doc_token_pageRange) from the user and creates an XML rendition from the input PDF. The XML rendition contains information about the file path of all the content in the PDF, which can be used by web-based clients to search the PDF.	Doc2
watermark_adts	Creates a PDF with a textual watermark. User specifies font appearance and placement, as well as advanced PDF options.	Doc6 > Doc4

Note: For merging, the format of the source and merging document(s) must be supported by the document_to_pdf profile.

A linearized PDF is a PDF file that is structured to allow for fast web viewing. The first page of a linearized PDF is displayed in the browser before the entire file is downloaded from the web server.

4.5.2 Transformation Services - Media profiles

Multiple plug-ins are required for some transformations to deal with different source files. For example, resizing a Photoshop file invokes the Image1 plug-in, whereas a TIFF file is resized using the Image2 plug-in.

Table 4-2: Transformation Services - Media profiles

Profile name	Action	Plug-in name
System profiles:		
addText_direct	Adds text to images.	Image3

Profile name	Action	Plug-in name
addText_psd	Adds text to images.	Image3
AI_processing	Processes Adobe Illustrator files.	
AI_processing_mobile	Processes Adobe Illustrator files for mobile.	
autoGenPreviewProxy	Generates a preview proxy.	Image2, Image1, Image3, PowerPoint1
autoGenPreviewProxy_AI	Generates a preview proxy.	
autoGenPreviewProxy_pdfstoryboard	Generates a proxy.	Image3, PDFStoryboard
autoGenProxy	Creates a low-resolution JPEG rendition.	Image1, PowerPoint1, Image2, Image3
autoGenProxy_AI	Creates a low-resolution JPEG rendition for PDF source files.	
autoGenProxy_pdfstoryboard	Creates a low-resolution JPEG rendition for PDF storyboard source files.	PDFStoryboard, Image3
convertColorspace	Convert colorspace.	Image3
embedMetadataToContent_xmp	Adds copyright information to image properties.	XMP, EXIF
extractEXIFMetadata	Adds EXIF image metadata properties to object metadata.	EXIF
extractEXIFXMP	Extracts XMP metadata.	EXIF
extractXMP	Extracts XMP metadata; target output is XML.	XMP
img_extract_props	Extracts properties of an image file.	Image2
importExtractProperties	Extracts properties of files on import by invoking plug-in profiles (irw_extract_props, avi_extract_props, ppt_extract_props, img_extract_props, imw_extract_props).	
imw_extract_props	Extracts properties of an image file.	Image3
irw_extract_props	Extracts properties of an image file.	Image1
powerpointDefaultStoryboard	Generates a default PowerPoint storyboard in JPEG format.	PowerPoint1
powerpointRegistration	Registers PowerPoint formats.	PowerPoint1
powerpointRegistrationStoryboard	Generates a PowerPoint storyboard at the time of registration.	PowerPoint1

Profile name	Action	Plug-in name
powerpointStoryboard	Generates a PowerPoint storyboard in JPEG format.	PowerPoint1
ppt_assembler	Creates PowerPoint assemblies.	PowerPoint1
ppt_extract_props	Extracts the properties of a PowerPoint file.	PowerPoint1
ppt_slide_generator	Creates slide objects for PowerPoint files.	PowerPoint1
reset_thumbnail	Generates three thumbnails for all Transformation Services - Media formats.	Image1, Image2, Image3 PowerPoint1, PDFStoryboard
resize_AI	Resizes an AI file.	
resize_direct	Adjusts the size of an image.	Image1, Image2, Image3
resize_sct	Resizes the selected object to SCT format.	
resize_svg	Resizes an SVG image.	Image1, Image2, Image3
resize_viaPNG	Processes an image resize through either resize_direct or transformTo_direct system profiles.	Image1, Image2, Image3
resize_wbmp	Resizes the selected object to WBMP format.	
svg_resize_direct	Adjusts the size of an SVG image.	Image1, Image2, Image3
thumbnail	Generates thumbnails in JPEG format.	Image1, PowerPoint1, Image2, Image3
thumbnail_AI	Generates a thumbnail for an AI file.	
thumbnail_ppt	Generates thumbnail for PowerPoint formats.	PowerPoint1
tiff_to_xCP_jpg	Converts TIFF files to JPEG format for the xCP client.	Image3
tiff_to_xCP_pdf	Converts TIFF files to PDF format for the xCP client.	Image3
transform_sct	Converts the selected object to SCT format.	
transform_AI	Transforms AI files.	
transformAI_To_EPS	Transforms AI files to EPS format.	

Profile name	Action	Plug-in name
transform_wbmp	Converts the selected object to WBMP format.	
transformTo_direct	Transforms source files into a new format.	Image1, PowerPoint1, Image2, Image3
transformTo_viaPNG	Transforms source files into a new format; invokes transformTo_direct system profile.	
transformAI_To_EPS	Converts Illustrator files to EPS.	Image3
addMultilineWatermark_direct	Applies a multiline tiled watermark on images. Provides options to define watermark text color, text size, and rotation angle.	Image4
User profiles:		
addText	Creates a new layer and adds text to an image.	
autoContrast	Automatically adjusts contrast to improve an image's appearance.	Image3
autoLevels	Automatically adjusts levels in an image.	Image3
blendLinear	Blends an image and a solid color with a linear gradient.	Image3
blendRadial	Blends an image and a solid color with a radial gradient.	Image3
canvasSize	Sets canvas size by cropping or expanding canvas to meet required dimensions.	Image3
changeColorMode	Changes the ICC color profile of an image format.	Image3
changeResolution	Changes the resolution, or number of pixels displayed per unit.	Image3
copyright	Embeds copyright information as metadata.	
embedIPTCContent	Embeds IPTC metadata from object in image properties.	EXIF
embedMetadataToContent_mts	Embeds write object properties (attributes) to content.	XMP, EXIF
extractEXIFMetadataToObject	Adds EXIF image metadata properties to object metadata.	EXIF
extractMetadataToObject_mts	Writes content properties to object properties (attributes).	XMP

Profile name	Action	Plug-in name
flip	Flips image horizontally or vertically.	
ppt_conversion	Converts PowerPoint files.	
resize	Adjusts the size of an image.	Image1, Image2, Image3, PowerPoint1
resize_preserveRatio	Resizes image while preserving the aspect ratio.	Image1, Image2, Image3, PowerPoint1
rotate	Rotates image using a specific angle of rotation.	Image3
transformAIToPDF	Converts a file from AI (Adobe Illustrator) to PDF format.	Image3
transformEPS_To_PDF	Converts a file from EPS to PDF format.	Image3
transformPDF_To_EPS	Converts a file from PDF to EPS format.	Image3
transformTo	Converts content to a new format.	Image1, Image2, Image3
transformTo_pdf	Converts a file to PDF format.	Image3
trim	Trims an image by removing pixels on the top, bottom, left, and/or right sides of images that are of constant color.	Image3
watermark	Applies a watermark.	Image3
addMultilineWatermark	Adds multiline tiled watermark on image.	Image4

4.5.3 Transformation Services - Audio/Video profiles

The audio and video profiles are detailed in "Transformation Services - Audio/Video profiles" on page 194.

Table 4-3: Transformation Services - Audio/Video profiles

Profile name	Action	Plug-in name
System profiles:		
extractAudioMetadata	Extracts metadata from audio files.	Video1
render_subclip	Renders the subclip.	Video1
storyboard_video	Generates storyboard renditions for video formats.	Video1
video_registration	Registers video formats.	
video_registration_d2	Registers video formats for client.	

Profile name	Action	Plug-in name
video_registration_xcp	Registers video formats for xCP.	
addWatermarkVideo_direct	Applies a text watermark on video files. Provides options to define watermark text color, text size, and position.	Video1
User profiles:		
to_3gp	Converts to 3GPP and 3GPP2 format.	Video1
to_avi	Converts video files to AVI format.	Video1
to_DV_Stream	Converts video files to DV Stream format.	Video1
to_flash	Converts video files to FLV (Flash Video) format.	Video1
to_flash9	Converts video files to F4V format.	Video1
to_gxf	Converts source files to GXF format. Performs noise reduction, cropping, gamma correction, and video contrast.	Video1
to_m4v	Converts to MPEG4 video for Apple's iTunes.	Video1
to_mp4	Encodes source files to the MPEG 4 Simple profile.	Video1
to_mpeg1_layer3	Converts audio files to MP3 (MPEG 1) format or extracts audio from video to MP3 (MPEG 1).	Video1
to_mpeg1_SystemStream	Converts source files to MPEG 1 System Stream with MPEG 1 Layer 2 audio. Performs noise reduction, cropping, gamma correction, and video contrast.	Video1
to_mpeg2_layer3	Converts audio files to MP3 (MPEG 2) format or extracts audio from video to MP3 (MPEG 2).	Video1
to_mpeg2_ProgramStream	Converts source files to MPEG 2 Program Stream video format with MPEG 1 Layer 2 audio. Performs noise reduction, cropping, gamma correction, and video contrast.	Video1
to_mxf	Encodes source files to MXF media format.	Video1
to_QuickTime	Convert videos to QuickTime format.	Video1

Profile name	Action	Plug-in name
to_QuickTime_DVStd	Converts to generic QuickTime in Standard Definition DV quality, suitable for editing and for use in multimedia applications.	Video1
to_real	Converts source files to RealNetworks SingleRate format with RealAudio audio codec. Performs noise reduction, cropping, gamma correction, and video contrast.	Video1
to_wav	Converts audio files to WAVE format or extracts audio from video to WAVE.	Video1
to_webm	Converts to WebM simple profiles.	Video1
to_wma	Converts audio files to WMA format or extracts audio from video to WMA.	Video1
to_wmv	Converts to WMV, Microsoft's preferred format for streaming media.	Video1
addWatermark_video.xml	Applies a text watermark on video files. Provides options to define watermark text color, text size, and position.	Video1

4.5.4 XML Transformation Services transformation profiles

XML Transformation Services is shipped with a number of predefined system and user profiles. These profiles are detailed in “XML Transformation Services user profiles” on page 196.

Table 4-4: XML Transformation Services user profiles

Profile name	Action	Plug-in name
dita_transform	Transforms DITA source files into PDF, ZIP, HTML. Transforms DITAMAP to Compiled HTML Help, Eclipse Help, Java Help, RTF, and DocBook format.	XPUB
docbook_transform_chm	Transforms DocBook source files to Compiled HTML Help format.	XPUB

Profile name	Action	Plug-in name
docbook_transform_html	Transforms DocBook source files to HTML format.	XPUB
docbook_transform_pdf	Transforms DocBook source files to PDF format.	XPUB
xml_transform_pdf	Transforms XML to PDF.	XPUB
xml_transform_html	Transforms XML to HTML.	XPUB

4.6 Plug-ins Installed with Transformation Services

Each Transformation Services product includes a set of plug-ins, based on the features and formats supported by the type of transformation.

If you need to make any configuration changes to plug-ins, this information helps you to determine which plug-in configuration files need to be modified.

Table 4-5: Plug-ins installed with Transformation Services products

Plug-in name	Transformation type			
	Transformation Services - Documents	Transformation Services - Media	Transformation Services - Audio/Video	XML Transformation Services
Doc2	√			
Doc3	√			
Doc4	√			
Doc5	√			
Doc6	√			
Doc7	√			
Doc8	√			
Doc9	√			
Doc11	√			
Doc12	√			
EXIF		√		
IMAGE1		√		
IMAGE2		√		
IMAGE3	√	√		
IMAGE4		√		
PDFStoryBoard	√	√		

Plug-in name	Transformation type			
	Transformation Services - Documents	Transformation Services - Media	Transformation Services - Audio/Video	XML Transformation Services
POWERPOINT1		√		
POWERPOINT2		√		
VIDEO1			√	
XMP	√	√		
XPUB				√

Note: There are no plug-ins to embed metadata for BMP, HTM, and EPS file formats, but if the files already have XMP or EXIF metadata embedded upon import, then the metadata can be extracted.

Chapter 5

Creating a Transformation Services WebServices Activity

Install the latest version of the following software before creating a Transformation Services WebServices Activity:

- xCelerated Composition Platform (xCP) Designer
- Documentum CM Server

Note: Ensure that you deploy the Transformation Services WebServices (`transformation.ear` or `transformation.war` files) in a supported application server, according to the *OpenText Documentum Content Management - Transformation Services Installation Guide (EDCCT250400-IGD)*.

To create a Transformation Services WebServices Activity, first create endpoints, and then create a process with the Transformation Services WebServices Activity.

Endpoints represent a connection to external services, for example, an SMTP host or a web service URL. To create a WebServices Activity for invoking Transformation Services, create an endpoint for:

- ProfileService
- TransformationService

To create an endpoint:

1. In xCP Designer, navigate to **Application** and click **Endpoints**.
2. Click **+** and create two endpoints of type **WebService (SOAP)**.
3. For the first endpoint, in the dialog box, type a valid name and the WSDL URL to point to the Transformation Services ProfileService.

   When `transformation.ear` is deployed, `ProfileService wsdl` looks as follows:

   ```
   http://<CTSWS_HostName_or_IP>:<port>/services/transformation/
   ProfileService?wsdl/
   ```

   When `transformation.war` is deployed, `ProfileService wsdl` looks as follows:

   ```
   http://<CTSWS_HostName_or_IP>:<port>/transformation/services/transformation
   /ProfileService?wsdl/
   ```

4. For the second endpoint, in the dialog box, type a valid name and the WSDL URL to point to the Transformation Services TransformationService.

   When `transformation.ear` is deployed, TransformationService wsdl looks as follows:

```
http://<CTSWS_HostName_or_IP>:<port>/services/transformation/TransformationService?
wsdl/
```

When `transformation.war` is deployed, TransformationService wsdl looks as follows:

```
http://<CTSWS_HostName_or_IP>:<port>/transformation/services/transformation
/TransformationService?wsdl/
```

Note: The *xCP Designer Help* contains detailed information for providing valid names to endpoints.

To create a process with a Transformation Services WebServices Activity:

1. In xCP Designer, navigate to **Processes** and create a new process.

2. Click **Process Properties** to open a dialog box.

3. Under **Data**, add a new package with the **Name** `sourcedoc` and the **Type Base Content**.

4. Create the process variables as shown in the following table:

   Table 5-1: Process Variables

Process variable name	Value description
profilename	Default value: <*document_to_pdf*> (can be any profile name which has to be executed). Type: String
profileid	Leave the value empty. Type: String
parametername	Default value: <*any parameter you want to pass*>. Type: String
parametervalue	Default value: <*any default value for the parameter*>. Type: String
targetformat	Default value: <*pdf*> (can be any value depending on the profile selected). Type: String
objectidvaluetype	Set the default value to `OBJECT_ID`. Type: String

   Note: The variable names, “parametername” and “parametervalue” in the table, are provided as examples. If the profile chosen requires more parameters, you must supply all parameter names and values, as appropriate.

5. Create a **Call Web Service** activity using drag and drop of the icon from **Activities** to **Process Designer**.

6. Double-click **Call Web Service** in **Process Designer** to open the **Activity Inspector** dialog box.

7. Choose the endpoint that you created for **profileservice**. In the **Operation** drop-down list, choose **getProfileByName** and click **Next**.

8. Authentication for a Transformation Services web service can be done in a single or chained manner. The *xCP Designer Help* contains instructions for setting up the SOAPHeader required for calling a OpenText™ Documentum™ Content Management Foundation Java API-based web service.

   The following code example uses a single authentication process by providing a <*ServiceContext*> to each web service call. In the “SOAPHeaderElement[0]” create a “Concat” function and append the following two strings, changing the userName, password, and repositoryName as appropriate.

   ```
   <ServiceContext xmlns="http://context.core.datamodel.fs.documentum.emc.com/"
   xmlns:ns2="http://properties.core.datamodel.fs.documentum.emc.com/"
   xmlns:ns3="http://profiles.core.datamodel.fs.documentum.emc.com/"> <identities
   xmlns:xsi="http://www.w3.org/2001/XMLSchema=instance" repositoryName="<RepoName>"
   password="<Password>" userName="<UserName>" xsi:type="RepositoryIdentity"/>
   </ServiceContext>
   ```

9. Map the profilename process variable created to SOAPBody > parameters > getProfileByName > profileName.

10. Map the ProcessData > Execution Data > docbase > name to SOAPBody > parameters > getProfileByName > repository.

11. Click **Next** to view the Output Message Mapping in the **Activity Inspector** dialog box. Map the SOAPBody > parameters > getProfileByNameResponse > return > ProfileId > ObjectId > id to ProcessData > Variables > profileid.

12. Create a second **Call Web Service** activity using drag and drop of the icon from **Activities**. Double-click **Call Web Service** to open the **Activity Inspector** dialog box.

13. Choose the endpoint that you created for **transformationservice**. In the **Operation** drop-down list, choose **addJob** and click **Next**.

14. In **Input Message Mapping**, choose “SOAPHeaderElement[0]” and create a “Concat” function in it. In the Concat function, add the following strings to provide the repository name, user name, and password, as appropriate:

   ```
   <ServiceContext xmlns="http://context.core.datamodel.fs.documentum.emc.com/"
   xmlns:ns2="http://properties.core.datamodel.fs.documentum.emc.com/"
   xmlns:ns3="http://profiles.core.datamodel.fs.documentum.emc.com/"> <identities
   xmlns:xsi="http://www.w3.org/2001/XMLSchema=instance" repositoryName="<RepoName>"
   password="<Password>" userName="<UserName>" xsi:type="RepositoryIdentity"/>
   </ServiceContext>
   ```

15. Map the parameters as shown in the following table:

Table 5-2: Process Data Parameters

Process data parameters	SOAP input
ProcessData > packages > sourcedoc > Object ID	SOAPBody > parameters > addJob > jobticket > @sourceObjectId
ProcessData > packages > sourcedoc > Content > Format	SOAPBody > parameters > addJob > jobticket > @sourceFormat
ProcessData > Variables > profileid	SOAPBody > parameters > addJob > jobticket > Profile > ProfileId > ObjectId > @id
ProcessData > Variables > targetFormat	SOAPBody > parameters > addJob > jobticket > @targetFormat
ProcessData > Variables > parametername	SOAPBody > parameters > addJob > jobticket > ParamProperties[0] > @name
ProcessData > Variables > parametervalue	SOAPBody > parameters > addJob > jobticket > ParamProperties[0] > @value
ProcessData > Variables > objectIdValueType	SOAPBody > parameters > addJob > jobticket > Profile > ProfileId >@valueType
ProcessData > Variables > profileName	SOAPBody > parameters > addJob > jobticket > Profile > @name

16. Click **Next**. You can map the response to any process variables you prefer or leave it empty.

17. Deploy the process as part of the xCP application, using the xMS server or a different method (according to the xCP documentation). The process or workflow is ready for execution using any client application (for example, Documentum Administrator).

Chapter 6

Troubleshooting

This chapter provides troubleshooting information for errors encountered from Transformation Services products. *OpenText Documentum Content Management - Transformation Services Installation Guide (EDCCT250400-IGD)* provides additional information regarding items that can affect the installation of your products.

6.1 Troubleshooting tips for Transformation Services products

This section includes troubleshooting tips and topics for Transformation Services products.

6.1.1 Storing the intermediate output of chain profile to repository

There is an option to store the output of an intermediate profile to the repository.

Note: This option is available only for chain profiles.

Add the following as one of the *InnerTokenMapping* to the InnerProfile entries of required profiles:

```
<InnerTokenMapping LocalProfileToken="true"
InnerProfileToken="store_result_in_repo" Literal="true"/>
```

To store the intermediate PDF output from the document_to_pdf profile which is part of the document_registration_xcp chain profile, add *InnerTokenMapping* to document_to_pdf InnerProfile as shown in the following code snippet:

```
<ProfileChain>
<InnerProfile path="/System/Media Server/System Profiles/document_to_pdf"
waitOnCompletion="true" useTargetFormat="true">
<InnerTokenMapping LocalProfileToken="true"
InnerProfileToken="store_result_in_repo" Literal="true"/>
...
</InnerProfile>
...
</ProfileChain>
```

6.1.2 Preserving temporary or intermediate files to debug a problem

There is a configuration option to preserve temporary or intermediate files. This configuration should only be used when debugging a problem and disabled as soon as it is no longer required.

The configuration can be overwritten (when “false”) using profile. For the profile, the values are:

0 – NO (do not keep temp files)

1 – YES (keep temp files)

The configuration element is in `CTSServerService.xml` (*`<Documentum Content Transformation Services_HOME>`*`\Config`):

```
<KeepTempFiles>NO</KeepTempFiles>
```

By switching this value to YES, input files as well as plug-in results will not be deleted after a transformation. Also, this value can be overwritten by a profile parameter:

```
<Parameter name="keep_temp" label="Keep temporary files" controltype="list"
datatype="string" default="0" isRequired="false">
<ValueList>
<Value label="No">0</Value>
<Value label="Yes">1</Value>
</ValueList>
</Parameter>
```

The output and source files can be found at the following file paths:

- for outputs – *`<Documentum Content Transformation Services_HOME>`*`\cache`
- for source files – *`<Documentum Content Transformation Services_HOME>`*`\docbases\`*`<docbase_name>`*`\config\temp_sessions`

Notes

- The default file path for *`<Documentum Content Transformation Services_HOME>`* is `C:\Documentum\CTS`.
- To reflect changes made to `CTSServerService.xml`, Transformation Services must be restarted.

6.1.3 Error events and log files

Any time Transformation Services fails to process a particular item, it queues an event (dm_mediaserver_error) to the Inbox of the repository user defined as the system operator in the server configuration file. The first step in troubleshooting is to check the system operator's Inbox for any messages that might explain the error.

Additionally, review the `CTS_Log.txt` log file. This log file contains errors and exceptions that are specific to the server. The default file path of the log file on the Transformation Services host, is in the following file path:

```
<Documentum Content Transformation Services_HOME>\logs\
```

Plug-in error exceptions are sent to individual log files that are specific to each plug-in. These plug-in log files allow you to quickly troubleshoot exceptions that are specific to each plug-in. For example, errors relating to the Image3 plug-in are found in the `Image3_Log.txt` log file.

The plug-in log files are located in the `<Documentum Content Transformation Services_HOME>\logs\` file path on the Transformation Services host.

Note: If logging is enabled separately in `log4j2.properties`, log files will be in the `<Documentum Content Transformation Services_HOME>\docbases\<docbase name>\config\logs\` folder. Otherwise, they are in the main logs file path at `<Documentum Content Transformation Services_HOME>\logs\`.

6.1.4 A transformation request fails

Occasionally, a transformation request may fail to process, or may process incorrectly. Typically, an examination of any error messages will indicate the problem.

The following actions can be performed to determine the cause of a transformation request failure.

To troubleshoot the failure of a transformation request:

1. Check for error messages in the system operator's Inbox indicating that the source file may not adhere to system specifications. For example, the file may be corrupted, or compression or other similar features (such as codec) of the file may not be supported by Transformation Services.

2. Check that disk space and memory on the Transformation Services server host, and the Documentum CM Server host, is sufficient to handle the requested file transformation.

3. Restart the Transformation Services server host and attempt the transformation again.

   Restarting the server will not affect the queue.

6.1.5 A profile error occurs

Errors related to profiles may occur. For example, if a profile was edited and updated manually, there may be issues with the XML file.

There are a number of ways to determine the cause of a profile error.

To identify the cause of a profile-related error:

1. Check the Inbox of the system operator for error messages. Error messages here may indicate the cause of the error.

2. Check the `CTS_log.txt` log file or the main Documentum log file, `log4j.log`, for exceptions.

3. If you suspect the profile itself to be the cause of the error, use Documentum Administrator or WDK client to ensure you have three folders in the repository for profiles:

   - `/System/Media Server/Profiles/`
   - `/System/Media Server/System Profiles/`
   - `/System/Media Server/command line Profiles/`

   These folders should be created when you install Transformation Services. Ensure the profile is in the correct folder.

4. Ensure that the command line file referenced by the profile (dm_media_profile object) exists in the appropriate folder (preceding list) and that the name in the profile is identical to the actual file name. For example, if a dm_media_profile object *resize* contains the following link to its command line file,

   ```
   /System/Media Server/command line Profiles/resize.xml
   ```

5. Ensure that the `ProfileSchema.dtd` is stored in the repository folder:

   ```
   /System/Media Server/Profiles
   ```

6. If a new user profile is not appearing in the WDK application, but the log file indicates that the profile was successfully imported, ensure that the following tags appear in the profile file (typically after the `Formats` section):

   ```
   <Filters>
   .....
   .....
   .....
   <Filter name="CTSProduct" value="<CTS product initialism>" />
   <Filter name="Visibility" value="Public" />
   .....
   .....
   .....
   </Filters>
   ```

   Caution

   The value of <*CTSproduct*> must only be that product that is configured to the repository, which means that, it can be any one of the following

> values: MTS, ADTS, XTS, and AVTS. If Transformation Services - Media is configured and the value of <*CTS product initialism*> is set to “AVTS”, then this profile will not be executed by the Transformation Services - Media server.

7. Restart the Transformation Services.

6.1.6 Cannot add a rendition to a particular format

If Transformation Services cannot add a rendition to a particular format, first check the Inbox of the system operator for error messages, and then check the `CTS_log.txt` file. Look for a detailed message that describes the problem. For example, the message “Object with id=...does not exist” would indicate that the object was deleted from the repository before the server could add a rendition to it.

6.1.7 If Transformation Services server cannot communicate to the repository

Transformation Services connects to the repository using a repository SuperUser account. If the Transformation Services server cannot log in to the repository, ensure that a valid repository SuperUser is identified for the product to use.

To identify the cause of a Transformation Services server not logging in to the repository:

1. Ensure that the repository is running.

2. Check your network connections.

3. Check that the connection broker is running.

4. Check that the `dfc.properties` file points to the connection broker that recognizes your repository.

5. Log in to the Transformation Services host as an administrator.

6. Open the Transformation Services file path, located in the file path:

   ```
   <Content Transformation Services_HOME>\config\
   ```

7. Open the `SessionService.xml` file in any text or XML editor.

8. Check the values of the MediaServerName attributes *userName* and *passwordFile* of the Transformation Services element:

   ```
   <CTSServer AttributeName="userName" AttributeValue="<admin_user"/>
   <CTSServer AttributeName="passwordFile" AttributeValue="C:\
   DOCUME~1\CTS\docbases\<repository_name>\config\pfile\
   mspassword.txt"/>
   ```

9. Ensure that you have configured the `SessionService.xml` file to the proper user in the repository who has SuperUser access. If necessary, check with the repository administrator to verify user name and password information.

“Administering and Configuring Transformation Services through Documentum Administrator“ on page 31 provides more information on the repository user name and password.

10. Save and close the `SessionService.xml` file.

11. Restart the Transformation Services server and/or the repository.

6.1.8 Transformation Services fails to function

If Transformation Services fails to function and you receive an error message when attempting transformations, the error may occur if the Global Registry does not have the current version of Transformation Services DARs configured against it. The Global Registry is a common repository where commonly used DAR files are installed.

Ensure that Rich_Media_Services and Transformation DAR files are run against the Global Registry repository. *OpenText Documentum Content Management - Transformation Services Installation Guide (EDCCT250400-IGD)* provides more information.

6.1.9 Transformation Services Administration Agent gives AGENT_INACCESSIBLE message

If the Transformation Services Administration node in Documentum Administrator is showing an AGENT_INACCESSIBLE message, it indicates that Documentum Administrator was unable to connect to the agent. This could be as a result of a number of reasons:

- The server method on the Documentum CM Server is not running.

  Check the service on the Documentum CM Server machine. There should be a service called Documentum Java Server Method.Documentum Administrator

- The Transformation Services Administration Agent is not running.

  There is a service called Documentum CTS AdminAgent on the Transformation Services machine for the agent. If this service is running, you should be able to ping the agent by opening your browser and pointing it to http://machinename:<*port*>/CTSAgent/CTSAgent (the default port is 9095). It should respond with a page saying “Documentum Content Transformation Services Agent” if it is running.

  If the AdminAgent is not running, then follow the instructions in “Starting and stopping the Transformation Services Administration Agent” on page 32 to start the Documentum CTS AdminAgent.

- The server method on the Documentum CM Server is not there.

  You can look for the server method using Documentum Administrator (**Administration > Job Management > Methods**). Do a search for CTSAdminMethod. Documentum CM Server must be version 6.x or later.

- There is no space left on the Transformation Services Administration Agent machine.

  Check the machine to ensure that there is available space.

- If you see following error in the `server.log` file of the Documentum CM Server host, then you must turn off Windows firewall in the Transformation Services machine:

  ```
  ERROR [default task-16] com.documentum.mthdservlet.DoMethod - Exception invoking
  CTSAdmin.
  ```

  Check AdminAgent URL in `cts_instance_info`. You should be able to ping AdminAgent from Documentum CM Server host.

6.1.10 Transformation Services fails to transform any document to PDF (using Tools->Transform) that is imported to a web cabinet as WebPublisher admin user

A transformation request reaches the Transformation Services server but fails with an exception.

This occurs because WebPublisher Documents work with 'application_code'. So any session that does not have any application_code associated with it will be treated as a Read only session.

To resolve this issue, in the `dfc.properties` files, set dfc.application_code=dm_wcm. This will ensure all sessions created will have application_code associated with it.

6.1.11 Transformation Services is not responsive after a set of transformations

Transformation Services is not responsive after a set of transformations and does not poll new queue items from the repository, even if the QueueProcessorContext threads are active and running.

The issue happens when Transformation Services fails to find the `cts_resources.properties` and `dam_resources.properties` files under the *`<DFC User Directory>`*`\config\` folder. This is fixed so that during startup, Transformation Services checks to see if these files are available in the specified folder, and if not found, will throw an exception in the `CTS_log.txt` file and stop the service. The log file will give more information about the expected file path of these files. This issue normally happens if Transformation Services is installed in a 'non-default' file path.

For older version of the product, resolve this issue as follows:

- Check to see if there is a `stderr.log` file created under *`<Content Transformation Services_HOME>`*`\config\` folder or perform a search on the host for this file.

- If the file has any exceptions logged, there could be an issue:

```
java.lang.RuntimeException: cts_resources.properties file not found:
C:\Documentum\CTS\lib\documentum\config\cts_resources.properties (The system
cannot find the path specified)
```

- Place these files under the folder specified, and then restart the Transformation Services.

The issue occurs mainly because the system is not getting the correct “DFC User Directory” file path through the Foundation Java API call. This is something configured in the `dfc.properties` file either through `dfc.data.dir` or `dfc.user.dir` property. In case these entries are missing in the `dfc.properties` file, Foundation Java API will be using the *`<current working directory>`*`\documentum\` as the Foundation Java API user file path.

6.1.12 Manual re-configuration of Transformation Services instance if the repository is deleted

If a repository configured with Transformation Services is permanently unavailable, some manual re-configuration is necessary. For example, if the repository cannot be recovered after a crash, or if the repository is deleted without removing all instances of Transformation Services, this procedure is required. Without manual re-configuration, the Transformation Services fails to start for *all* repositories that the Transformation Services server was configured against, not just the deleted repository. You will see the following error in the `CTS_log.txt` file:

```
11:28:08,406 INFO [ main] CTSServerHandlerImpl - ***************
11:28:08,718 ERROR [ main] CTSServerHandlerImpl - Unable to start
 the default handler.
com.documentum.cts.exceptions.internal.CTSServiceManagerException:
 Unable to start the default handler.
Cause Exception was: A new instance of ICTSService with SUBTYPE =
 CTSPLUGIN could not be created
com.documentum.cts.exceptions.internal.CTSServiceException: Unable
 to start the default handler.
at com.documentum.cts.impl.services.CTSServiceBaseImpl.startAll
(CTSServiceBaseImpl.java:930)
at com.documentum.cts.impl.services.CTSServiceBaseImpl.start
(CTSServiceBaseImpl.java:565)
at com.documentum.cts.services.CTSServiceManager.getServiceBySubType
(CTSServiceManager.java:545)
at com.documentum.cts.impl.services.ctsserver.CTSServerHandlerImpl.
getMediaPluginHandler(CTSServerHandlerImpl.java:1074)
at com.documentum.cts.impl.services.ctsserver.CTSServerHandlerImpl.
start(CTSServerHandlerImpl.java:541)
at com.documentum.cts.impl.services.CTSServiceBaseImpl.startAll
(CTSServiceBaseImpl.java:925)
at com.documentum.cts.impl.services.CTSServiceBaseImpl.start
(CTSServiceBaseImpl.java:565)
at com.documentum.cts.services.CTSServiceManager.registerServices
(CTSServiceManager.java:268)
at com.documentum.cts.services.CTSServiceManager.bootstrapServiceFramework
(CTSServiceManager.java:161)
at com.documentum.cts.services.CTSServiceManager.getInstance
(CTSServiceManager.java:118)
at com.documentum.cts.services.ctsserver.CTSServerStartup.main
(CTSServerStartup.java:107)
```

To fix this, perform the following procedure on the Transformation Services host:

1. Navigate to `<Documentum Content Transformation Services_HOME>\config\` (for example, `C:\Documentum\CTS\config`).
2. Make a backup copy of the following files:
   - `CTSProfileService.xml`
   - `CTSServerService.xml`
   - `SessionService.xml`
3. In any text or XML editor, open the `CTSProfileService.xml` file and remove the entire `ProfileManagerContext` tag for the repository that is no longer available.
4. In any text or XML editor, open the `CTSServerService.xml` file and remove the entire `QueueProcessorContext` tags for the repository that is no longer available. There are two of these tags for each repository that has been configured, so you will need to remove both entries for the repository that is no longer available.
5. In any text or XML editor, open the `SessionService.xml` file and remove the entire `LoginContext` tag for the repository that is no longer available.
6. Delete the folder for the repository that is no longer available under `<Documentum Content Transformation Services_HOME>\docbases\` (for example, `<Documentum Content Transformation Services_HOME>\docbases\`).
7. Open the Windows Registry Editor (go to **Start > Run > regedit**).
8. For the Transformation Services installed, delete the folder with the repository name that is no longer available under:

   ```
   HKEY_LOCAL_MACHINE\SOFTWARE\Documentum\CTS\MTS\Docbases
   ```

   For example, the full path for Transformation Services, including the repository name, will appear as:

   ```
   HKEY_LOCAL_MACHINE\SOFTWARE\Documentum\CTS\MTS\Docbases\<docbase-name>
   ```
9. Stop the Transformation Services.
10. Delete the Transformation Services log files.
11. Start the Transformation Services.

6.1.13 Profile modifications

When there are changes made to the profile, the changes will be loaded automatically to the profile. However, restarting the Transformation Services server ensures that the cache, which Transformation Services maintains for all profiles, is created afresh.

6.1.14 Rendition failures due to expired login tickets

In some cases, users may experience rendition failures resulting from expired login tickets. The following error message appears in the log file:

```
[DM_SESSION_E_START_FAIL]
[DM_SESSION_E_LDAP_AUTHENTICATION_FAILED]
[DM_SESSION_E_LDAP_BIND]
```

A KeepSessionTimeOut node exists in the `SessionService.xml` file, to prevent the caching of login tickets.

However, if the KeepSessionTimeOut node is changed from its default of 0, this error may occur. Ensure that the value for the KeepSessionTimeOut node is *less than* the ticketed session expiry timeout value in the `server.ini` file for a given repository.

6.1.15 Transformation Services Queue Item Cleanup – Pre-processing delay

If a delay in starting up the Pre-processing becomes noticeable, it can be fixed by updating the markerInterval entry to have the same or lesser value of queueInterval in the `CTSServerService.xml` file:

```
<CTSServer AttributeName="markerInterval" AttributeValue="10"/>
<CTSServer AttributeName="queueInterval" AttributeValue="10"/>
```

6.1.16 After restarting Documentum CM Server, Transformation Services throws a “server communication failure” exception

After Documentum CM Server is restarted, Transformation Services throws “server communication failure” exception and has to be restarted to work. Configuration options are available to help improve the Transformation Services response rate for reconnecting if the Documentum CM Server is restarted.

Note: Try updating only the Transformation Services host first, then the Documentum CM Server host if necessary.

There are two main configuration elements:

1. Add the following to the `dfc.properties` file:

   ```
   dfc.session.keepalive.enable = true
   ```

2. Add the following registry entry:

   ```
   HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameter\KeepAliveTime
   ```

   And, set the DWORD value to 300000(Decimal).

   Two additional operating system settings to consider adjusting are *Keepalive interval* and *Keepalive retry*.

Note: 'Keepalive' is a networking concept, not a Documentum feature. More details on how these setting work and how to set them can be obtained from operating system documentation or other resources.

6.1.17 Users with version permission get Inbox notification on legacy transformations

To fix this problem, perform the following procedure:

1. Check out the `legacy_to_pdf` system profile from the repository, in the folder: `\System\Media Server\System Profiles\`.

2. Modify ALL of the innerprofile entries to have an additional `innertokenmapping` tag for "add_rendition_properties" token in it as follows.

   ```
   <InnerProfile path="/System/Media Server/System Profiles/document_to_pdf"
    waitOnCompletion="true" useTargetFormat="true">
   ..
   ..
       <InnerTokenMapping LocalProfileToken="false" InnerProfileToken=
   "add_rendition_properties" Literal="true"/>
   </InnerProfile>
   ```

   If LocalProfileToken is set to false, properties will NOT be added to the rendition. The default value will be true. If the tag is missing or if the value is "true" properties will be added. If the user has only VERSION permission, set this attribute value to "false" so that at least the renditions will be created as in previous versions.

3. Check in the `legacy_to_pdf` system profile.

6.1.18 Transparency in PNG renditions lost for Illustrator files

PNG renditions made from AI files lose their transparency when using Transformation Services.

For example, backgrounds lose their transparency and appear as white when the images appear on a webpage that has a non-white background color.

To maintain an image's transparency remove the -flatten option from the profile command line file.

6.1.19 Preserving ICC profiles during transformations

To allow Transformation Services to preserve the ICC profile associated with ICC-aware files (for example, Photoshop files), remove the following from the command line file generating the desired rendition(s):

-strip and +profile icc -profile “doc_token_cmyk_profile” -profile “doc_token_rgb_profile”

6.1.20 Text wrapping problems with large double byte characters

When double-byte characters are entered and given a large font size using the Add Text profile, some of the text is lost as it does not wrap.

6.1.21 Embedding metadata for date datatype other than default

If you want to embed metadata to content for a date datatype and you want to specify a date format other than the default which is MM/dd/YYYY HH:mm:ss, add an attribute dateFormat to the MetadataMapper line in the embedMetadataToContent.xml command line file specifying the format to use. For example:

```
<MetadataMapper name="date_created" value="cch_photo.date_created" datatype="date"
dateFormat="yyyy-MM-dd" token="doc_metadata_date_created"/>
```

where cch_photo is the custom type created and date_created is the attribute for the cch_photo type.

6.1.22 Troubleshooting storyboard issues generated by the PDFStoryboard plug-in

When you attempt to use the PDFStoryboard plug-in to generate storyboards from PDFs that contain image content only, which were generated by the Doc6 plug-in, it is seen that the images do not have content, and are blank.

Hence, it is recommended that you use the Image3 plug-in to generate storyboards, since the PDFStoryBoard plug-in cannot produce images at a resolution higher than 96 dpi.

6.1.23 Resolving issues with email content containing double byte characters

When you import emails with double byte characters in the subject, content, and attachment names, it is seen that in the subject and attachment names, the double byte characters are corrupted and appear as “???”. For the renditions to be created without character corruption, it is recommended that you install Arial Unicode Microsoft font on the Transformation Services server machine.

6.1.24 Performance degradation due to add_rendition_properties settings

You may observe a degradation in performance when add_rendition_properties is set to “true” for profiles. To improve performance, add the following inner token mapping to the inner profiles:

```
<InnerTokenMapping Literal="true" InnerProfileToken="add_rendition_properties"
LocalProfileToken="false"|>
```

If the tag is missing or if the value is set to “true”, properties will be added.

Appendix A. Formats supported by Transformation Services - Documents

Table A-1: Document Transformations Supported for Source and Target Formats

Source format	Target format
BMP	PDF, PDF/A, PDF/X
DOC, DOT, DOCX, DOTX	PDF, PDF/A, PDF/X, HTML, PS, PDF TEXT
DWG, DXF, DGN, DWF, DWFX	PDF, PDF/A, PDF/X
EML	PDF, PDF/A, TXT
EPS	PDF, PDF/A, PDF/X
GIF	PDF, PDF/A, PDF/X
HTML	PDF, PDF/A, PDF/X, PS, PDF TEXT
JPEG	PDF, PDF/A, PDF/X
MPP	PDF
MSG	PDF, PDF/A,TXT
OTT, ODS, ODT	PDF, PDF/A, PDF/X
PDF	PDF, PDF/A, PDF/X, HTML, PS, PDF TEXT
PNG	PDF, PDF/A, PDF/X
POT, POTX(12,14), PPTX(12,14)	PDF, PDF/A, PDF/X, PS, PDF TEXT
PPT	PPS, PDF, PDF/A, PDF/X, PS, PDF TEXT
PS	PDF, PDF/A, PDF/X, PDF TEXT
RTF, TXT	PDF, PDF/A, PDF/X, HTML, PS, PDF TEXT
TIFF (single and multi-page)	PDF, PDF/A, PDF/X
VSDX, VSD	PDF, PDF/A, PDF/X, HTML, PS, PDF TEXT
XLS, XLT, XLSX, XLTX	PDF, PDF/A, PDF/X, HTML, PS, PDF TEXT

Note: Transformation Services supports Microsoft Office 2019 renditions through the Doc6 plug-in.

Table A-2: Capabilities Supported for Document Formats

Format	Creates thumbnail*	Creates storyboard*	Creates low-resolution rendition*	Writes XMP to content	Extracts XMP from content
BMP	Y		Y		
DOC, DOCX	Y	Y	Y		
DOT, DOTX	Y	Y	Y		
DWG, DXF, DGN, DWF, DWFX	Y	Y	Y		
EML			Y		
GIF	Y		Y		
HTML	Y	Y	Y		
JPEG	Y		Y		
MPP	Y	Y	Y		
MSG	Y	Y	Y		
ODS	Y	Y	Y		
ODT	Y	Y	Y		
OTT	Y	Y	Y		
PDF	Y	Y	Y	Y	Y
PNG	Y	Y	Y		
POT, POTX	Y	Y	Y		
PPT, PPTX	Y	Y	Y		
PS	Y	Y	Y		
RTF	Y	Y	Y		
TIFF	Y	Y	Y		
TIFF (multipage)	Y	Y	Y		
VSDX, VSD	Y	Y	Y		
XHTML	Y	Y	Y		
XLS, XLSX	Y	Y	Y		
XLT, XLTX	Y	Y	Y		

Y indicates that the capability is supported for the format.

* Thumbnail, storyboard, and low-resolution renditions are not available out of the box. Some configuration is required.

Appendix B. Formats supported by Transformation Services - Media

Table B-1: Media Transformations Supported for Source and Target Formats

Source format	Target format
AI [1]	BMP, GIF, JPEG, JPEG2000, PBM, PGM, PNG, PNM, PPM, PSD, TIFF, WBMP
BMP	PSD, JPEG, GIF, PNG, TIFF, BMP, PNM, PGM, PPM, PBM, PCX, TGA,
CR2	PSD, JPEG, GIF, PNG, TIFF, BMP, PNM, PGM, PPM, PBM, PCX, TGA, SGI, SCT, JPEG2000
DNG	PSD, JPEG, GIF, PNG, TIFF, BMP, PNM, PGM, PPM, PBM, PCX, TGA, SGI, SCT, JPEG2000
EPS [1]	PSD, JPEG, GIF, PNG, TIFF, BMP, PNM, PGM, PPM, PBM, PCX, TGA, SGI, SCT, JPEG2000, WBMP
GIF	PSD, JPEG, GIF, PNG, TIFF, BMP, PNM, PGM, PPM, PBM, PCX, TGA, SGI, SCT, JPEG2000, WBMP
JPEG	PSD, JPEG, GIF, PNG, TIFF, BMP, PNM, PGM, PPM, PBM, PCX, TGA, SGI, SCT, JPEG2000, WBMP
JPEG2000	PSD, JPEG, GIF, PNG, TIFF, BMP, PNM, PGM, PPM, PBM, PCX, TGA, SGI, SCT, WBMP
PDF [1]	PSD, JPEG, GIF, PNG, TIFF, BMP, PNM, PGM, PPM, PBM, JPEG2000, WBMP
PGM	PSD, JPEG, GIF, PNG, TIFF, BMP, PNM, PGM, PPM, PBM, JPEG2000, WBMP
PNG	PSD, JPEG, GIF, PNG, TIFF, BMP, PNM, PGM, PPM, PBM, PCX, TGA, SGI, SCT, JPEG2000, WBMP
PNM	PSD, JPEG, GIF, PNG, TIFF, BMP, PNM, PGM, PPM, PBM, JPEG2000, WBMP
PPM	PSD, JPEG, GIF, PNG, TIFF, BMP, PNM, PGM, PPM, PBM, JPEG2000, WBMP
PSD [2]	PSD, JPEG, GIF, PNG, TIFF, BMP, PNM, PGM, PPM, PBM, PCX, TGA, SGI, SCT, JPEG2000, WBMP
SCT	PSD, JPEG, GIF, TIFF, BMP, PCX, TGA, SGI, SCT, JPEG2000
SGI	PSD, JPEG, GIF, TIFF, BMP, PCX, TGA, SGI, SCT, JPEG2000
SVG	PSD, JPEG, GIF, PNG, TIFF, BMP, SVG, PNM, PGM, PPM, PBM, WBMP, JPEG2000
TIFF	PSD, JPEG, GIF, PNG,TIFF, BMP, PNM, PGM, PPM, PBM, WBMP, JPEG2000

Source format	Target format
WBMP	PSD, JPEG, GIF, PNG,TIFF, BMP, PNM, PGM, PPM, PBM, WBMP, JPEG2000
WMF	JPEG, GIF, PNG, TIFF, BMP, PNM, PGM, PPM, PBM, PCX, TGA, SGI, SCT, JPEG2000, WBMP

1. Transformation Services requires that the supported version of Ghostscript be installed to perform transformations where the source or target format is AI, EPS, or PDF. The product *Release Notes* on My Support (support.opentext.com) provides more details.
2. When performing a transformation in which PSD (Photoshop) is both the source and target format (PSD to PSD), all of the layers are flattened.

Table B-2: Capabilities for Supported Media Image Formats

Input format	Create thumbnail and low-resolution rendition	Extract properties	Write XMP to content	Extract XMP from content	Extract EXIF metadata	Resize
AI	Y			Y		Y
BMP	Y	Y				Y
CR2	Y					Y
DNG	Y	Y				Y
EPS	Y			Y		Y
GIF	Y	Y		Y		Y
JPEG	Y	Y	Y	Y	Y	Y
JPEG2000	Y	Y				Y
PCD	Y	Y				Y
PCX	Y	Y				Y
PGM	Y					Y
PNG	Y	Y	Y	Y		Y
PNM	Y	Y				Y
PPM	Y					Y
PSD	Y	Y	Y	Y		Y
SCT	Y	Y				Y
SGI	Y	Y				Y
SVG	Y	Y		Y		Y
TGA	Y	Y				Y
TIFF	Y	Y	Y	Y	Y	Y

Input format	Create thumbnail and low-resolution rendition	Extract properties	Write XMP to content	Extract XMP from content	Extract EXIF metadata	Resize
WBMP	Y	Y				Y
WMF	Y					Y
Y indicates that the capability is supported for the format.						

Appendix C. Formats supported by Transformation Services - Audio/Video

Table C-1: Audio and Video Transformations Supported for Source and Target Formats

Source format	Target format
Video	
3G2	3GP, 3GP2, AVI, DV, F4V, FLV, GXF, M4V, MPEG-4, MP3, QuickTime, MPEG, MPEG2, WAV, WMA, WMV, MXF, RM, WebM
3GP	3GP, 3GP2, AVI, DV, F4V, FLV, GXF, M4V, MPEG-4, MP3, QuickTime, MPEG, MPEG2, WAV, WMA, WMV, MXF, RM, WebM
ASF	3GP, 3GP2, AVI, DV, F4V, FLV, GXF, M4V, MPEG-4, MP3, QuickTime, MPEG, MPEG2, WAV, WMA, WMV, MXF, RM, WebM
DV	3GP, 3GP2, AVI, DV, F4V, FLV, GXF, M4V, MPEG-4, MP3, QuickTime, MPEG, MPEG2, WAV, WMA, WMV, MXF, RM, WebM
F4V	3GP, 3GP2, AVI, DV, F4V, FLV, GXF, M4V, MPEG-4, MP3, QuickTime, MPEG, MPEG2, WAV, WMA, WMV, MXF, RM, WebM
FLV	3GP, 3GP2, AVI, DV, F4V, FLV, GXF, M4V, MPEG-4, MP3, QuickTime, MPEG, MPEG2, WAV, WMA, WMV, MXF, RM, WebM
GXF	3GP, 3GP2, AVI, DV, F4V, FLV, GXF, M4V, MPEG-4, MP3, QuickTime, MPEG, MPEG2, WAV, WMA, WMV, MXF, RM, WebM
LXF	3GP, 3GP2, AVI, DV, F4V, FLV, GXF, M4V, MPEG-4, MP3, QuickTime, MPEG, MPEG2, WAV, WMA, WMV, MXF, RM, WebM
M4V	3GP, 3GP2, AVI, DV, F4V, FLV, GXF, M4V, MPEG-4, MP3, QuickTime, MPEG, MPEG2, WAV, WMA, WMV, MXF, RM, WebM
MPEG	3GP, 3GP2, AVI, DV, F4V, FLV, GXF, M4V, MPEG-4, MP3, QuickTime, MPEG, MPEG2, WAV, WMA, WMV, MXF, RM, WebM
MPEG2	3GP, 3GP2, AVI, DV, F4V, FLV, GXF, M4V, MPEG-4, MP3, QuickTime, MPEG, MPEG2, WAV, WMA, WMV, MXF, RM, WebM
MPEG-4	3GP, 3GP2, AVI, DV, F4V, FLV, GXF, M4V, MPEG-4, MP3, QuickTime, MPEG, MPEG2, WAV, WMA, WMV, MXF, RM, WebM

Source format	Target format
MSS	3GP, 3GP2, AVI, DV, F4V, FLV, GXF, M4V, MPEG-4, MP3, QuickTime, MPEG, MPEG2, WAV, WMA, WMV, MXF, RM, WebM
MXF	3GP, 3GP2, AVI, DV, F4V, FLV, GXF, M4V, MPEG-4, MP3, QuickTime, MPEG, MPEG2, WAV, WMA, WMV, MXF, RM, WebM
Quicktime	3GP, 3GP2, AVI, DV, F4V, FLV, GXF, M4V, MPEG-4, MP3, QuickTime, MPEG, MPEG2, WAV, WMA, WMV, MXF, RM, WebM
RM	3GP, 3GP2, AVI, DV, F4V, FLV, GXF, M4V, MPEG-4, MP3, QuickTime, MPEG, MPEG2, WAV, WMA, WMV, MXF, RM, WebM
WebM	3GP, 3GP2, AVI, DV, F4V, FLV, GXF, M4V, MPEG-4, MP3, QuickTime, MPEG, MPEG2, WAV, WMA, WMV, MXF, RM, WebM
WMV	3GP, 3GP2, AVI, DV, F4V, FLV, GXF, M4V, MPEG-4, MP3, QuickTime, MPEG, MPEG2, WAV, WMA, WMV, MXF, RM, WebM
Audio	
MP3	3GP, 3GP2, AVI, F4V, FLV, M4V, MPEG-4, MP3, QuickTime, MPEG, MPEG2, WAV, WMA, WMV, RM, WebM
WMA	3GP, 3GP2, AVI, F4V, FLV, M4V, MPEG-4, MP3, QuickTime, MPEG, MPEG2, WAV, WMA, WMV, RM, WebM
WAV	3GP, 3GP2, AVI, F4V, FLV, M4V, MPEG-4, MP3, QuickTime, MPEG, MPEG2, WAV, WMA, WMV, RM, WebM

Table C-2: Capabilities Supported for Audio and Video Formats

Format	Creates thumbnail	Creates storyboard	Creates default rendition	Extracts properties	Streamable*
Video Formats					
3G2	Y	Y	Y	Y	
3GP	Y	Y	Y	Y	
ASF	Y	Y	Y	Y	
AVI	Y	Y	Y	Y	
DV	Y	Y	Y	Y	
F4V	Y	Y	Y	Y	
FLV	Y	Y	Y	Y	
GXF	Y	Y	Y	Y	

Format	Creates thumbnail	Creates storyboard	Creates default rendition	Extracts properties	Streamable*
LXF	Y	Y	Y	Y	
M4V	Y	Y	Y	Y	
MPEG	Y	Y	Y	Y	
MPEG2	Y	Y	Y	Y	
MPEG-4	Y	Y	Y	Y	
MSS	Y	Y	Y	Y	
MXF	Y	Y	Y	Y	
QuickTime	Y	Y	Y	Y	Y
RM	Y	Y	Y	Y	
WMV	Y	Y	Y	Y	Y
WebM	Y	Y	Y	Y	
Audio Formats					
MP3			Y	Y	
WAV			Y	Y	
WMA			Y	Y	
Y indicates that the capability is supported for the format. *A properly configured Streaming Server and a client are required. For more information, see your product documentation.					

Appendix D. Formats supported by XML Transformation Services

Table D-1: XTS Transformations Supported for Source and Target Formats

Source format	XML standard	Target formats
DITA	DITA	PDF, HTML, ZIP
DITAMAP	DITA	PDF, HTML, Compiled HTML Help (CHM), Java Help, Eclipse Help, RTF, and DocBook
XML	DITA	PDF, HTML, ZIP
	DocBook	PDF, HTML, Compiled HTML Help (CHM)
XML (custom)	XML Schema	PDF, HTML

Appendix E. Audio and Video Codecs

Table E-1: Audio and Video codecs supported for input formats

Format	Video codec	Audio codec
3GP/3G2	H.263 H.264/AVC MPEG-4	AAC-LC AMR - NB
ASF	WM V7 WM V8 WM V9WM9 VC-1	Windows Media Audio 9.2 WMA V9 ProWMA V9 Voice
AVI	Cinepak Radius	MP3 MS CCITT A-Law MS CCITT U-Law PCM
AVI	FFmpeg MPEG 1/2 FFmpeg MPEG-4H.264/AVC IBM Motion JPEG Intel I.263/H.263 Intel Indeo iYUV 4:2:0 Microsoft Video 1 Uncompressed AVI Stream XviD MPEG-4 Video	Intel ADPCM MP3 MS ADPCMMS CCITT A-Law MS CCITT U-Law PCM
AVI	Huffman Lossless	Intel ADPCM MP3 MPEG Audio Layer2 MS ADPCM MS CCITT A-Law MS CCITT U-Law MS GSM 6.10 PCM
DV	DV25/50 DVCPRO25/50	PCM
F4V	AVC	AAC-LC
FLV	Flash 8/VP6	MP3
FLV	Flash Screen Video Flash Video (Sorenson Spark) H264/AVC	AAC-LC MP3 PCM
GXF	DV25/50 MPEG2	PCM
LXF	MPEG2	PCM
QuickTime	H.264/AVC Sorenson 1	AAC-LC AMR A-lawMP3 U-law

Format	Video codec	Audio codec
QuickTime	Cinepak Radius Sorenson 1 Sorenson3	MACE 3:1 MACE 6:1
QuickTime	AVID ABVB AVID Meridian InterlacedAVID Meridian progressive AVIDNuVistaCinepak Radius Cinewave 8-bit YUV H264/AVC MotionJPEG A MotionJPEG B MPEG4 Video PhotoJPEG Sorenson3Uncompressed	AAC-LC QDesign Music 2 QualcommPureVoice PCM
QuickTime	DNxHD DV25/50 DVCPRO25/50DVCPRO HD IMX30/40/50 ProRes/ProResHQ	PCM
M4V	H.263/AVC H.263/AVC Intel MPEG4	AAC-LC
MP3	–	MPEG1 Layer3 MPEG2 Layer3
MP4	H.263 H.264 MPEG4	AAC-LC AMR MP3
MPEG1	MPEG1 MPEG2	MPEG1 Layer2 MPEG2 Layer3
MPEG2	MPEG2 ML@MP MPEG2 4:2:2	Dolby5.1 AC3 MPEG1 Layer2
MSS	MPEG2	MPEG1 Layer2
MXF	MPEG2 DVCPRO/DV XDCAM HD D-10 MPEG2 30/40/50	PCM
WAVE	–	PCM PCM A-law PCM mu-law
WebM	VP8	Vorbis
WMA	–	Windows Media Audio 9.2 WMA V9 ProWMA V9 Voice

Format	Video codec	Audio codec
WMV	WM V7 WM V8 WM V9WM9 VC-1	PCM Windows Media Audio 9.2 WMA V9 Pro WMA V9 Voice

Table E-2: Audio and Video Codecs Supported for Output Formats

Format	Video codec	Audio codec
3GP/3G2	H.264/AVC H.263 MPEG-4	AAC-LC AMR - Narrow Band AMR- Wide Band
AVI	H.263 H.264/AVC Microsoft MPEG-4 V3 Microsoft Video1 Raw Video (RAW I420)	PCM
DV	DV 25	PCM
F4V	H.263	MPEG1 Layer3
FLV	Sorenson Spark	MPEG1 Layer3
GXF	DV MPEG2 4:2:2	PCM
M4V	H.264/AVC	AAC-LC
MP3	–	MPEG1 Layer3 MPEG2 Layer3
MP4	H.264/AVC MPEG4 Video	AAC-LC
MPEG1	MPEG1	MPEG1 Layer3
MPEG2	MPEG2 ML@MP MPEG2 4:2:2	MPEG2.5 Layer3
MXF	D-10 MPEG2 30/40/50	PCM
QuickTime	DV 25	PCM
QuickTime	H.264/AVC MPEG-4 Photo JPEG Raw video (I420) Sorenson Video 1	AAC-LC (MPEG4 Audio) PCM
WAV	–	PCM
WebM	VP8	Vorbis

Format	Video codec	Audio codec
WMV	Windows Media Video 7 (wmv1) Windows Media Video 8 (wmv2)	WMA1 WMA2
WMA	–	WMA1 WMA2