TEAMCENTER

Content Management — Deployment and Administration

Teamcenter 2412

SIEMENS

About Siemens Digital Industries Software

Siemens Digital Industries Software is a global leader in the growing field of product lifecycle management (PLM), manufacturing operations management (MOM), and electronic design automation (EDA) software, hardware, and services. Siemens works with more than 100,000 customers, leading the digitalization of their planning and manufacturing processes. At Siemens Digital Industries Software, we blur the boundaries between industry domains by integrating the virtual and physical, hardware and software, design and manufacturing worlds. With the rapid pace of innovation, digitalization is no longer tomorrow’s idea. We take what the future promises tomorrow and make it real for our customers today. Where today meets tomorrow. Our culture encourages creativity, welcomes fresh thinking and focuses on growth, so our people, our business, and our customers can achieve their full potential.

Support Center: **support.sw.siemens.com**

Send Feedback on Documentation: **support.sw.siemens.com/doc_feedback_form**

Contents

Managing objects for S1000D authoring

Creating schema objects

Creating topic and publication types

Creating a publication structure

Enabling references and links

Creating XML attribute mappings

Creating style types and stylesheets

Enabling graphics features

Enabling content editing

Enabling content publishing

1. What is Content Management?

Content Management is a component content management system for the creation of technical documentation. Technical publications consist of XML and SGML text components, which are stored in the Teamcenter database along with any referenced graphics used in the publications. The structure and content of a publication is validated by schemas, such as DITA or Docbook. The publication layout is implemented with stylesheets, which allows the same content to be formatted for different media, such as print, web, or interactive applications. Content developed as individual topics may be reused by reference in multiple publications.

Content Management works in conjunction with several publishing and editing tools, which enable publications to be generated for both online and print formats. Content Management includes support for multiple language versions and includes the ability to manage topics through translations and release cycles.

Three types of Content Management solutions are available:

- Base Content Management, which can be used with traditional and custom DTDs and schemas such as DocBook and Structured Product Labeling (SPL).
- S1000D Content Management, which enables management of documentation for either issue 2.2/2.3, 4.0, 4.1, 4.2, or 5.0 of the S1000D standard.
- DITA Content Management, which enables management of documentation for the DITA standard, through version 1.3, including specialization and support for DITA Open Toolkit (OT) version 2.4.6.

> Note:
>
> Editing tools are supported only on Windows clients. S1000D issue 2.3 is supported only on Linux and Windows. DITA publishing is supported only on Linux and Windows.

Separate from the Content Management solution, the document management functionality in Teamcenter enables you to manage documents created by other software applications, such as Microsoft Word, Microsoft Excel, and other programs.

Before Content Management can be used for authoring publications, the application administrator must complete several tasks to prepare Content Management for authors.

Pre-configured administration data is included in the *TC_ROOT***contmgmtbase_data\data**, *TC_ROOT***contmgmts1000d_data\data**, *TC_ROOT***contmgmts1000d40_data\data**, and *TC_ROOT***contmgmtdita_data\data** folders.

Two types of users access Content Management:

- Content Management administrators create and manage objects that enable content authoring.

- Content authors use Content Management to create and manage technical documentation.

2. Install Content Management using Deployment Center

Add Content Management application to your existing Teamcenter environment.

1. Log on to Deployment Center and select the environment to which you want to add Content Management.

2. Go to the **Applications** task. Click **Add or Remove Selected Applications**.

3. In the **Available Applications** panel, use the web browser search to find the following applications:

   - **Teamcenter→Content Management Active Workspace**
   - **Content and Document Management→Content Management Base**
   - (Optional) **Content and Document Management→Content Management DITA**
   - (Optional) **Content and Document Management→Content Management S1000D**
   - (Optional) **Content and Document Management→Content Management S1000D 4.X/5.X**

4. Select the application, and then click **Update Selected Applications**.

   Deployment Center automatically selects any additional dependent applications.

5. Go to the **Components** task.

6. In the **Selected Components** list, note any remaining components whose configuration status is not **100%**. Select each incomplete component, enter required parameters, and save component settings until all components in the environment show a configuration status of **100%**.

   When all components are fully configured, the **Deploy** task is enabled.

7. Go to the **Deploy** task. Click **Generate Install Scripts** to generate deployment scripts you will use to update affected machines.

   When script generation is complete, note any special instructions in the **Deploy Instructions** panel.

8. Locate deployment scripts, copy each script to its target machine, and then run each script on its target machine.

   For more information about running deployment scripts, see *Deployment Center — Usage*.

3. Install Content Management using Teamcenter Environment Manager

You can install Content Management using Teamcenter Environment Manager (TEM).

1. In the **Features** panel, select the following features:

   - **Base Install→Extensions→Content and Document Management→Content Management Base**
   - (Optional) **Base Install→Extensions→Content and Document Management→Content Management DITA**
   - (Optional) **Base Install→Extensions→Content and Document Management→Content Management S1000D**
   - (Optional) **Base Install→Extensions→Content and Document Management→Content Management S1000D 4.X/5.X**

2. To install Content Management for Active Workspace, select:

   - **Active Workspace→Client→Content Management Active Workspace**

4. Using Teamcenter XMetaL Client to manage content

Teamcenter XMetaL Client is an optional application that integrates XMetaL Author with Content Management for authoring and managing technical publications. Authors log in to Teamcenter using the XMetaL interface and can perform most authoring tasks and the workflow signoff task, without using the Teamcenter interface.

To install XMetaL Client, download and unzip*TC_INSTALL_DIR\additional_applications\xmetal_install.zip*. Open the *xmetal_install*folder and run *setup.exe*.

For configuration instructions, see the *Teamcenter XMetaL Client Administrator's Guide* located on the local drive where XMetaL Client is installed, for example: *c*:**\Program Files (x86)\Siemens\Teamcenter10\XMetaLPlugin\XMetal\CRCL\adapters\Teamcenter\docs**. For instructions on using Teamcenter XMetaL Client, see the *Teamcenter XMetaL Client Help* available from the **Help** menu in Teamcenter XMetaL Client.

If you are using Teamcenter XMetaL Client, do not install the XMetaL plugin in the Teamcenter interface.

> Note:
>
> Once installed, this plugin requires the user to associate the XMetaL application to the *aw___* file that is downloaded to the browser. A browser update may be needed. After that, the file should automatically launch into XMetaL without requiring the user to double click the *aw___*.

5. Basic concepts for using Content Management

Understanding the following basic concepts will help you use the features of Content Management.

Data objects are unique in Content Management

Items and item revisions are the fundamental data objects used to manage information in Teamcenter. Content Management uses specific types of items and item revisions, including topics, topic revisions, publications, and publication revisions in Base Content Management, and S1000D and DITA items in those solutions. Content Management also uses administration objects for defining the authoring environment, such as topic types, XML attribute mappings, and schemas.

Some administration objects—stylesheets, schemas, procedures, and translation offices—have item revisions. Content Management objects are managed like other Teamcenter objects.

You perform some tasks with these objects in the **Home** component view; however you use the **Publication Structure** view to create and maintain the contents of publications.

Topics are the basic component of publications

In Base Content Management, a topic is a component that may be part of one or more publications. A topic may contain text, tables, graphics, and subtopics. The term *topic* is also used in Content Management to refer to certain S1000D and DITA objects.

Content versions track edits to a topic

In addition to the revision ID and sequence ID appended to topics, each topic revision has a content version. The content version is a number that increases each time the content of the topic is edited. When a topic is first created, its content version is -1. After the content is edited once, the content version increases to 0, and then to 1, and so on. The content version of a topic revision is displayed in the **Summary** view. When a topic has a translation, you can compare the content version of the topic to the content version of the translation to determine if the topic has changed since the topic was translated. When you edit the content of a topic, you can save it without increasing the content version, which is useful when a topic has a related translated topic and you want the versions of the two to remain synchronized.

Publications are the complete structure and content of technical documents

In Base Content Management, a publication object holds all the contents of a traditional XML publication. A publication contains topics and subtopics, such as the cover page, back page, header, footer, and sections or chapters. DITA and S1000D Content Management solutions have unique publication-type objects that are used to create technical documentation.

The type of topics you can add to specific locations in a publication is defined by the Content Management administrator in an underlying hierarchical structure of publication types and topic types created according to the schemas your company uses.

Schemas define the building blocks of XML and SGML documents

Schemas define the basic structure of elements and attributes used in structured documents. When you relate publication and topic types to schemas in Content Management, the XML or SGML content is validated against the schema files that are defined by the Content Management administrator. A DTD is an example of a schema type.

Stylesheets control the format of documents

A *stylesheet* contains formatting information used to render, edit, or transform a topic or document. A *style type* is used to organize and manage the stylesheets associated with publications. These objects are defined by the Content Management administrator.

A graphic item may have several graphic options

A *graphic* item is the main object representing a set of different file types, or *graphic options*, for the same illustration. For example, a graphic item may be named *piston*, and it may have several associated graphic options named piston.jpg, piston.gif, and piston.bmp, which all represent the same illustration in different formats.

When you start an editing or publishing process, the Content Management system selects the applicable graphic option from the graphic, based on how the graphic options are defined by your application administrator. For example, during the publishing process, the graphic option selected may be a high-resolution .eps file registered with the **PDF** usage in the topic; and if the graphic item has a .png file registered with the **VIEW** usage, Content Management may display it in the topic for editing.

Integration with third-party tools enable you to author content

Content Management integrates tools, which are installed on each client, to support the following authoring tasks:

- **Publishing**

  Publishing tools render content to PDF, HTML, XHTML, and other proprietary formats, based on the stylesheets and applications you have available.

- **Editing**

  Editing tools are used to author and edit XML and SGML content and are launched when topics are opened for editing.

- **Comparing**

Comparing tools are used to compare differences in topics.

- **Viewing**

  Viewing tools are used to run an external tool to view topics. This is useful when a quick or remote preview session is needed, without the processing time required to generate a PDF document or other process-intensive stylesheet.

> Note:
>
> Cortona3D Teamcenter Integration is a third-party tool that provides the capability to author publications that are linked with assembly and part data in Teamcenter. The Cortona3D tool utilizes Content Management software but provides its own interface.

Composed documents assemble documents for reviewing

A composed document is created when you publish content. It contains a publication or topic with all its child topics collected in one file. A reviewer document is the same type of document, named differently so that it can be used to distinguish documents between those that have been processed through the review process (composed documents) and those that have not (reviewer documents). These documents can be exported and imported for content review outside of the system.

Workflows can aid in document review

You can design Teamcenter workflow process templates that incorporate your company's business practices and procedures. Your company can use these workflows to process topics through release cycles and between participants.

You can use Teamcenter workflows for document reviews by using Workflow Designer and Workflow Viewer.

You can manage content in different languages

Content Management manages multiple language versions of topics and publications, for example: English, Russian, and Simplified Chinese.

Several object types are used for translations:

- **Translations**

  A translation is an object created with the content of a topic in a different language than the original. Translations indicate the language and topic for which they are created. A translation is related to the source topic from which it is derived.

- **Translation orders**

A translation order is an object in the translation process used to provide information to the translator. It contains basic information, such as the translation office assigned and the requested delivery date.

- **Translation deliveries**

  A translation delivery contains the translation topics and has an associated .zip file that can be exported and sent to the translation company.

- **Graphic translations**

  Graphic translations are objects that are created when graphic options are imported. They store the language property associated with a graphic option when it is imported. Graphic translations are related to the original (master language) graphic items and to the graphic options with the associated translation languages.

6. Preparing Content Management for authors

Tasks in Content Management

Before authors can use Content Management, the Content Management administrator must complete the following tasks.

> Note:
>
> Although you can create these objects yourself, it is recommended that you import the Content Management **administration data** and modify the objects to meet your organization's needs. Depending on the needs of your organization, the administrator may also be required to complete tasks in addition to the ones listed here.

- Create at least one **style type** to organize multiple stylesheets based on a unique layout specification.
- Create **stylesheets**, which are used to manage formatting information for documents; and relate the stylesheets to the style type(s).
- Set up **editing**, **publishing**, **viewing**, and **comparing** tools, which are used to work with content in Content Management. Relate each tool to stylesheets when required for the tool.
- Create **topic types and publication types** and relate them to stylesheets. These are the objects that make up an XML publication structure and are selected by authors as they create publications and topics. In S1000D and DITA Content Management, topic types are predefined to match the objects for the standard, such as S1000D data modules and DITA concept topics. You still may need to create objects for them in Content Management, unless you import them from the Content Management **administration data**.
- (Optional) Create at least one **publication structure** with a publication type and related topic types, to define a structure that authors can use to create documents.
- Create **schema objects**, adding the schemas to the database; and relate them to publication and topic types. Schemas define the basic structure of elements and attributes used in a topic. For example, a DTD is a schema. DTDs contain the elements, attributes, entities, and notations used in XML documents.
- Create **XML attribute mappings** for the exchange of data between XML attributes and database object properties, and relate them to topic types.
- Create **reference topic types**, which identify the syntax used for building links and references to topics. Relate the reference topic types to XML attribute mappings and other topic types.
- **Enable navigation links**, so authors can create links between topics.

- **Enable graphics features**, so authors can use graphics in publications.
- (Optional) Create **translation offices and languages** to allow authors to manage content in different languages.
- (Optional) Create **transformation policies and procedures**, which are used for specialized handling of topics.
- Set Content Management preferences to determine behaviors for processing such as graphic filtering during import, publishing content to a file system, topic type filtering during topic creation, and XML attribute mapping.

Tasks in other Teamcenter applications

In addition to Content Management tasks, the administrator may complete some of the following tasks.

- Define your organization, including roles, groups, and users, in Organization.
- Control user access to data objects in Access Manager.
- Design Teamcenter workflow process templates that incorporate your company's business practices and procedures, in Workflow Designer.
- Define projects and programs and assign team members, in the Project and Program applications.
- Customize attributes and elements for Content Management objects, such as adding and updating business object properties and attaching naming rules to properties, in Business Modeler IDE.

> Caution:
>
> If you change naming rules, you must be sure to update all related configuration items to handle that change, such as stylesheets, attribute maps, transfer modes. If you want to use naming rules, ensure that you update every artifact that is affected by the naming rule.

7. Migrating Content Management data from a previous release

Update content reference type

In Teamcenter 11.2.3 and later, the relationship type between content references and graphic content references is changed from a GRM (generic relationship management) relation to a **PSOccurrence** relation. If you are upgrading from any prior release, you must run the **contmgmt_migration_1123** utility to update these relationships.

1. After the upgrade to the current release is complete, run the **contmgmt_migration_1123** utility.

2. Determine if any assigned relationships must be corrected, by reviewing the log file created by the utility in the temp directory. The log file is named **contmgmt_data_migration_1123#####.trc**, where **#####** is the date and time the log file was created.

Add the new relationship for topic references

In Teamcenter 10.0 and later, a new relationship exists between the **PSOccurrence** object between two related topics and the reference topic type that is between the two topic types. If you are upgrading from any prior release, you must run the **contmgmt_migration_100** utility to add this relationship to **PSOccurrence** objects. If more than one reference type exists between the two topic types, then the utility selects one and the selection is noted in a log file.

1. After the upgrade to the current release is complete, run the **contmgmt_migration_100** utility.

2. Determine if any assigned relationships must be corrected, by reviewing the log file created by the utility in the temp directory. The log file is named **contmgmt_data_migration_100#####.trc**, where **#####** is the date and time the log file was created.

Remove obsolete saved queries

In Teamcenter 10.1 and later, you may see saved queries listed in the Query Builder **Saved Queries** tree that are no longer needed.

- Use Query Builder to delete the saved queries with the prefixes **DC_** and **Civ0**.

8. Setting preferences for Content Management

Preferences related to Content Management

Some Teamcenter preferences must be set to control certain behaviors in Content Management.

- **AE_dataset_id_usage**

  This preference must be set to **OFF** for Content Management to function properly.

- **PLMXML_put_objects_in_newstuff_on_import**

  This preference can be created and set to **TRUE** to place imported objects, such as administration data, in a folder with the same name as the imported XML file under the **Newstuff** folder. If this preference is not created and set, you must search for the imported data.

- **TcCheckoutReserveOnly**

  In releases prior to 9.0, setting this preference was recommended for Content Management classes to optimize performance. If this preference is set, users cannot cancel the checkouts of Content Management objects. If you are upgrading from a release prior to 9.0 and you want to enable the cancelation of checkouts, you must update the preference to remove any Content Management classes from the list of values after the upgrade to the current release.

- **Multiple_revise_dialog_visible**

  To enable users to edit the names of new topic revisions using the **Multiple Object Revise** dialog box, this preference must be set to **true**. If this preference is set to **false**, you may still revise multiple topics, but you cannot edit the names.

- **PSEVariantsMode**

  Content Management supports only classic variants. You can set this preference to **legacy** to ensure that the variant mode always opens in the classic mode to avoid having to change the mode that is displayed.

- **PSEShowUnconfigdVarPref**

  When you are working with a publication structure, you can set this preference to show or hide components containing variants.

- **PSM_classic_variants_text_families** and **PSM_classic_variants_numeric_families**

Variant options have a string type or a numeric type and a name. The type of option you can create depends on the settings of these preferences.

Content Management preferences

Use Content Management preferences to determine behaviors for processing such as graphic filtering during import, publishing content to a file system, topic type filtering during topic creation, and XML attribute mapping.

You can also set some of these preferences by choosing **Edit→Options→Content Management**.

- **Cdi0KeyrefValidation**

  Specifies whether keyrefs used in topics referenced in maps are always validated during import and when you save changes and close the file in an editing session.

- **ContentManagerColumnsShownPref**

  Specifies the list of columns that are displayed in the **Publication Structure** view.

- **ContentManagerDisplayNameWithTypeColumnsShownPref**

  Stores the internal names of properties that appear as columns in the **Publication Structure** view.

- **ContentManagerShownColumnWidthsPref**

  Specifies the column width for the list of columns defined in the **ContentManagerColumnsShownPref** preference.

- **ctm0Allow_Invalid_Content_Publish**

  Specifies if content with schema validation error is published or exported. If the value of the preference is **false**, publish or export operations are not allowed for content that is invalid against schema.

  If the value of the preference is **true**, publish or export operations are allowed for content that is invalid against schema.

- **Ctm0Allow_Multiple_Version_In_DITA_Structure**

  Allows mixing multiple DITA topic versions in a DITA structure.

- **ctm0DisableS1000DItemIdCheck**

  Disables the schema validation check on attributes that comprise the unique identifier for S1000D objects at the time the objects are created.

- **ctm0FileNameSeparator**

  Adds the revision ID and sets the separator that is used when creating file names for the XML files representing Content Management topics.

  The value must not be the same character used in any file name or item ID. For example, the value _ (underscore) cannot be used if a file is being imported as *engine_stand.xml*.

  When using the DITA standard on Linux, the value must not be $, %, or +, due to a limitation of DITA OT.

- **ctm0GraphicClassPref**

  Displays installed graphic classes for selection during import.

- **ctm0GraphicExtPref**

  Filters the files that are listed for selection when graphics are imported.

- **ctm0GraphicPref.values**

  Defines the values listed in the **Graphic Link Clipboard** preference list box. The values are the *publication_standard* part of the **ctm0GraphicPref.***publication_standard***.name** preferences. These values are used to define the default XML tagging used when adding graphics to content for specific technical publication standards.

- **ctm0GraphicPref.dita.name**

  Specifies the text to be displayed in the **Graphic Link Clipboard** preference list box. The value represents the option to use when working with the DITA standard for publications.

- **ctm0GraphicPref.dita.text**

  Specifies the default XML tagging that you want to use in the editing tool when a graphic is copied from Content Management and pasted to content in the editing tool. The value is used when working with the DITA standard for publications.

- **ctm0GraphicPref.docBook.name**

  Specifies the text to be displayed in the **Graphic Link Clipboard** preference list box. The value represents the option to use when working with the DocBook standard or a similar XML standard for publications.

- **ctm0GraphicPref.docBook.text**

Specifies the default XML tagging that you want to use in the editing tool when a graphic is copied from Content Management and pasted to content in the editing tool. The value is used when working with the DocBook standard or a similar XML standard for publications.

- **ctm0GraphicPref.s1000d.name**

  Specifies the text to be displayed in the **Graphic Link Clipboard** preference list box. The value represents the option to use when working with the S1000D 4.0 or 4.1 standard for publications.

- **ctm0GraphicPref.s1000d.text**

  Specifies the default XML tagging that you want to use in the editing tool when a graphic is copied from Content Management and pasted to content in the editing tool. The value is used when working with the S1000D 4.0 or 4.1 standard for publications.

- **ctm0GraphicUsagePref**

  Maps graphic option usages to graphic file extensions to provide a default selection when graphics are imported to Content Management. System-defined graphic usages include: ICON, PDF, PRINT, SOURCE, THUMBNAIL, VIEW, and WEB.

- **ctm0_processing_data**

  Maps processing data properties to content during the XML attribute exchange in the compose process.

- **ctm0RestoreStateOfPubStructure**

  Controls the save state of the **Publication Structure** view.

- **ctm0TopicTypeGroup**

  Limits the topic type groups that can be used for creating new topics. If one or more values are entered, only topic types in the listed groups are available for selection when new topics are created.

- **ctm0_translation_graphicpriority**

  Specifies the graphic priority preference assigned to translation orders when they are created in Content Management.

- **DCt_Language_fnd0isoLanguageCountryCode_expression**

  Defines the expression value of the run-time property **fnd0isoLanguageCountryCode** for the **DCt_Language** business object. It combines the values of the ISO language and country code on the language object.

For example, if the language code is **en** and the country code is **US**, the value in this run-time property on **DCt_Language** for the default value is **en-US**. However, this preference can be modified for the value to appear in another way, such as **enUS**, **en_US**, or **en.US**.

- **deploy_absolute_path**

  Specifies the file path to the folder specified in the **publish_to_file_system_loc** preference for published content. This path is platform specific.

- **deploy_link_prefix**

  Specifies the URL to connect to the folder storing published content.

- **publish_to_file_system**

  Enables publishing to a file system.

- **publish_to_file_system_loc**

  Specifies the name of the parent folder to store published content.

9. Importing and exporting administration data

Administration data included with Content Management

Administration data includes the database objects and relationships that are set up to enable authors to create and manage documentation in Content Management. You can import and export administration data to transfer it from one system to another. When you install Content Management, Teamcenter provides example administration data for Content Management.

> Note:
>
> Default settings in Teamcenter allow users within the same user group (such as dba and Engineering) to check out objects. Therefore, the administrative objects should be imported by a user with administration (dba) privileges.

The Content Management installation provides three different sets of administration data. The Base administration data provides example languages and graphic priority lists in the Organization application. It also creates example publishing and editing tools, graphic attribute mappings, and XML attribute mappings. Although you can create these objects yourself in Content Management, it is recommended that you import this data and modify the objects to meet your organization's needs.

The S1000D and DITA administration data sets contain pre-configured topic types, schemas, and for DITA, the DITA Open Toolkit (OT) stylesheets, as defined by these standards. You can use these as the starting point for S1000D and DITA content management.

> Tip:
>
> It is recommended that you maintain an unchanged version of the provided objects for testing and comparison purposes.

Import Content Management administration data

> Note:
>
> Verify that the tool paths assigned to the provided editing and publishing tools match the paths for your installation of the tools.

1. To place imported objects in the **Newstuff** folder, ensure the **PLMXML_put_objects_in_newstuff_on_import** preference is added. If this preference is not set, imported objects are saved in the database, but not displayed in My Teamcenter.

2. Choose **Tools→Import→From PLMXML**.

3. Click ... to the right of the **Importing XML Object** box, to browse for and select the objects to import.

   If you are importing the administration data provided with the Content Management installation, select one of the following:

   - *TC_ROOT***contmgmtbase_data\data\admin\contmgmtbase_admindata.xml**, to import Base Content Management administrative objects.
   - *TC_ROOT***contmgmtdita_data\data\admin\contmgmtdita_admindata.xml**, to import DITA 1.1 Content Management administrative objects.
   - *TC_ROOT***contmgmtdita_data\data\admin\contmgmtdita12_admindata.xml**, to import DITA 1.2 Content Management administrative objects.
   - *TC_ROOT***contmgmtdita_data\data\admin\contmgmtdita13_admindata.xml**, to import DITA 1.3 Content Management administrative objects.
   - *TC_ROOT***contmgmts1000d_data\data\admin\contmgmts1000d_admin.plmxml**, to import S1000D 2.2 and 2.3 Content Management administrative objects.
   - *TC_ROOT***contmgmts1000d40_data\data\admin\contmgmts1000d_40_admin.plmxml**, to import S1000D 4.0.1 Content Management administrative objects.

     > Note:
     >
     > If you imported this data in a release prior to 10.0, you must re-import it to update your data to the S1000D 4.0.1 standard.

   - *TC_ROOT***contmgmts1000d40_data\data\admin\contmgmts1000d_41_admin.plmxml**, to import S1000D 4.1 Content Management administrative objects.
   - *TC_ROOT***contmgmts1000d40_data\data\admin\contmgmts1000d_42_admin.plmxml**, to import S1000D 4.2 Content Management administrative objects.
   - *TC_ROOT***contmgmts1000d40_data\data\admin\contmgmts1000d_50_admin.plmxml**, to import S1000D 5.0 Content Management administrative objects.

4. From the **Transfer Mode Name** list, select **ctm0_ContMgmtImportAdmin**.

5. Click **OK**.

> Tip:
>
> To verify the import, search for some of the object types you imported, or look in the **Newstuff** folder if your preferences are set to place new objects there.

Export Content Management administration data

You can export administrative objects such as schemas, topic types, stylesheets, and XML attribute mappings.

1. Select the object to be exported.
2. Choose **Tools→Export→To PLMXML**.
3. Click ... to the right of the **Export Directory** box, to browse for and select the select the directory in which to place the file.
4. In the **Export Filename** box, type the name of the export file.
5. From the **Transfer Mode Name** list, select **ctm0_ContMgmtExportAdmin**.
6. (Optional) Select **Open PLM XML File** to view the file when the export operation is complete.
7. (Optional) Select the **Perform Export In Background** check box to run the export in the background.
8. Click **OK** to export the object in the **Object List** list and close the dialog box.

   When the export completes, a dialog box appears.

   If the export was successful, you can click **Yes** to view the log file for the export.

   If there were errors during the export, the dialog box lists the errors. Click **Yes** to view information about the errors that occurred.

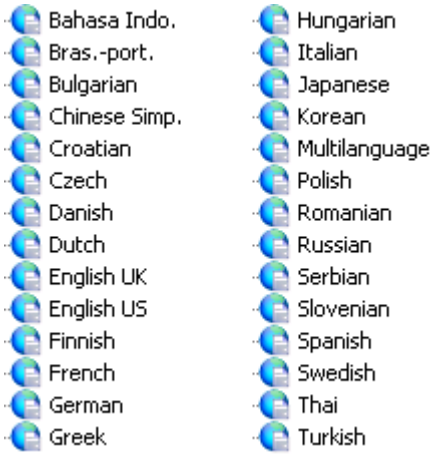
10. Pre-configured Base Content Management objects

Base Content Management can be used with traditional and custom DTDs and schemas, such as DocBook and Structured Product Labeling (SPL). Teamcenter provides Base administration objects and relationships as examples of how administration data works. To begin creating the required objects for managing documentation for Base Content Management, you can import the Base **administration data**, which includes the following example objects:

> Tip:
>
> It is recommended that you maintain an unchanged version of the provided objects for testing and comparison purposes.

Object type	Included objects
Editing tool	**XMetaL 9.0** Used for editing topics in XMetaL 9.0.
Publishing tool	• **InternalViewer** An internal tool that publishes XML content to the **Preview** view. The view displays XML in ASCII format, with tags, etc. • **FOP** Supports Formatting Objects Processor, a Java application included with Teamcenter that converts XSL Formatting Objects files to PDF. • **PDF serverside** Supports Antenna House XSL Formatter, one of many available publishing applications that can be purchased from a third-party supplier. If you were to use this tool, you would install the publishing application on a server for publishing content to PDF. • **serverside HTML publish** An internal tool that publishes content to HTML, using XSLT or CSS scripts, which are processed on the server.
Graphic attribute mapping	• **Default Graphic Attribute Mapping**

Object type	Included objects
	In this graphic attribute mapping, when a graphic is imported, all characters after the first period in the file name become the file extension. • **Default Graphic Attribute Mapping without suffix** In this graphic attribute mapping, when a graphic is imported, all characters after the last period in the file name become the file extension.
Graphic priority list	The Base administration data provides these graphic priority lists in the Organization application, which you can use to associate with graphics and editing and publishing tools as desired: • **EDIT** Includes graphics with these uses, in this priority: **VIEW**, **WEB PDF**, **PRINT**. In the example data, this is used by the **XMetaL 9.0** editing tool. • **PUBLISH TO HTML** Includes graphics with these uses, in this priority: **VIEW**, **WEB**. In the example data, this is used by the **serverside HTML publish** publishing tool. • **PUBLISH TO PDF** Includes graphics with these uses, in this priority: **VIEW**, **WEB**. In the example data, this is used by the **PDF serverside** publishing tool. • **REVIEW** Includes graphics with these uses, in this priority: **VIEW**, **WEB**. • **VIEW**

Object type	Included objects
	Includes graphics with these uses, in this priority: **VIEW**, **WEB**, **PDF**, **PRINT**, **THUMBNAIL**. In the example data, this is used by the **FOP** and **InternalViewer** publishing tools.
Language	The Base administration data configures 28 languages in the Organization application. You can add and remove languages as necessary. To minimize the menu options users must scroll through, it is recommended that you remove any languages that you do not use. 

11. Relating administrative items to enable authoring

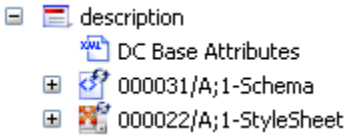
Relating administrative items

Administrators relate items such as stylesheets to style types, stylesheets to tools, schemas to topic types, and XML attribute mappings to topic types. Schemas and stylesheets have revisions; therefore when you relate items to schemas and stylesheets, you relate the item to the *revisions*.

You can relate an item to many parent items, creating multiple instances of the same item. Whenever the item is modified at one location, the modifications are reflected in all locations where the item is reused.

When you relate administrative items, you build the constraints that govern how authors can create and work with publications.

The following shows an example of a topic type with a related XML attribute mapping, schema revision, and stylesheet revision.



When you select an item in the **Home** component view, its related child items are listed in the **Details** view, which also shows the type of relation between the two items.

Relate one administrative item to another

> Note:
>
> To relate a topic type to itself, you must use the **Copy→Paste** option instead of dragging the topic type.

1. In the **Home** component view, locate the item you want to relate and drag it to the parent item you want to relate it to.

2. When the parent item appears highlighted, release the mouse button.

   The child item appears in the structure below the parent item.

12. Managing objects for DITA authoring

Creating objects for DITA authoring

DITA (Darwin Information Typing Architecture) is an XML-based architecture for developing technical publications. It includes an architectural specification, language specification, and DTD and schema implementations of the language. DITA Content Management is an optional Teamcenter solution.

To create the required objects for managing documentation for the DITA standard, you can import the DITA **administration data**. The DITA objects are related to version 1.0/1.1, 1.2, or 1.3 of the DITA DTDs. If you prefer to import modified versions of these DTDs or the DTD schemas, you must create and configure additional objects and import those DTDs.

> Note:
>
> - It is recommended that you maintain an unchanged version of the provided objects for testing and comparison purposes.
> - DITA publishing is supported only on Linux and Windows and relies on the DITA Open Toolkit (OT).

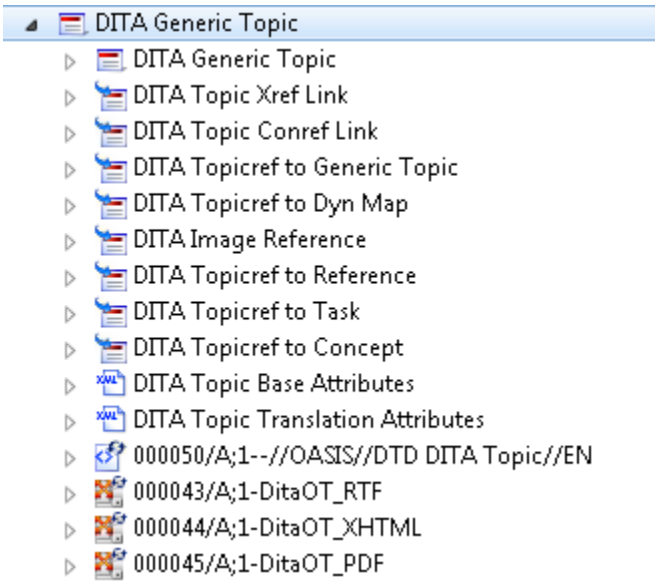
Objects you create include:

- DITA topic types
- DITA maps
- DITA composites
- DITA value filters
- Reference types
- Schemas
- Stylesheets
- XML attribute mappings
- DITA OT publishing tools

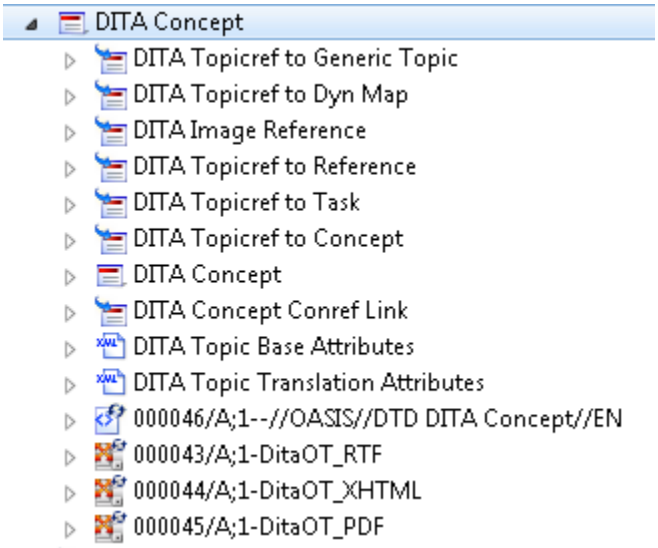
DITA 1.0/1.1 topic authoring objects

The DITA **administration data** includes the following topic types for DITA 1.0/1.1 topics, shown here with their related objects. You manage these topic types, along with the related objects required for them to function, so that authors can create and manage DITA topics.

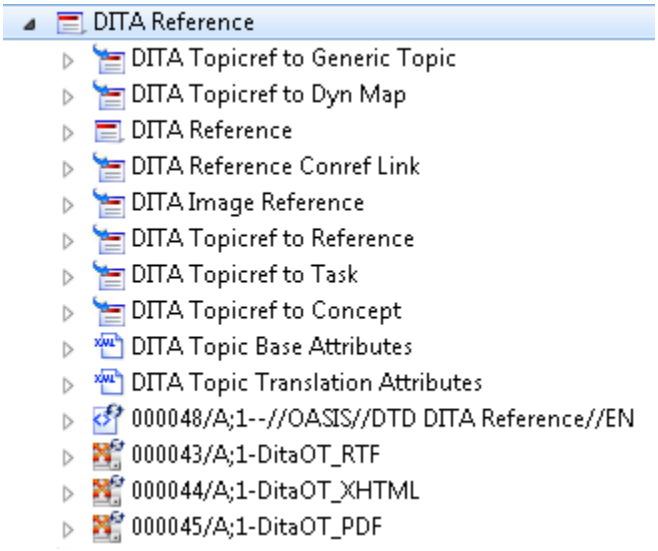
- DITA Generic Topic



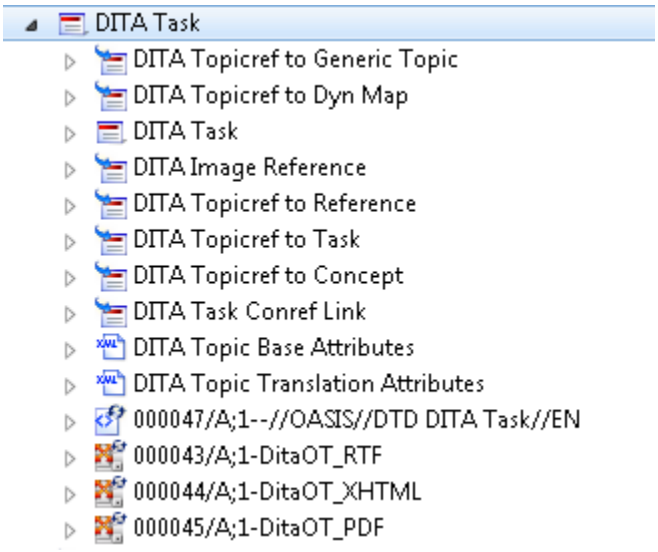
- DITA Concept



- DITA Reference



- DITA Task

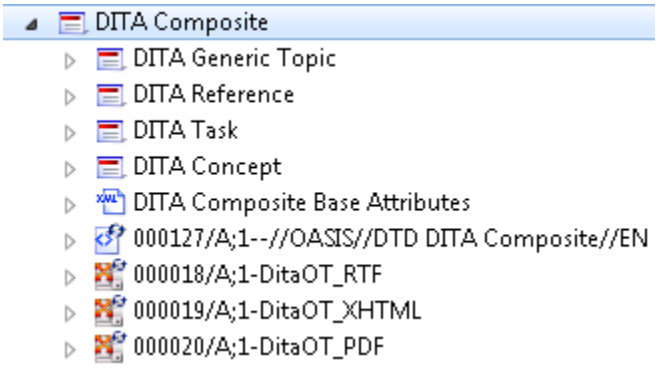


DITA 1.0/1.1 composite authoring objects

The DITA **administration data** includes a topic type for DITA 1.0/1.1 composites. The DITA composite, or *DITA base*, object provides a top-level container for multiple topics when you create single content documents. It allows the creation of any sequence of generic topics, tasks, concepts, and references.

> Note:
>
> An XML attribute mapping for defining the translation attribute mappings is not necessary on DITA composite publication types, because DITA composites are not translated. Translations are performed on the topics inside the composite.

When using the content of a DITA composite, the reference should be made to the top level topic within the DITA composite, not the composite itself. Only one top level topic should be included; all others should be nested or referenced.



DITA 1.0/1.1 map and bookmap authoring objects

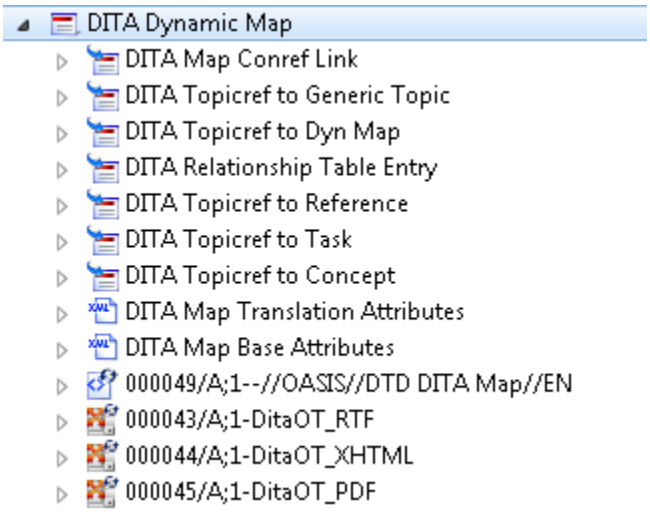
The DITA **administration data** includes topic types for DITA 1.0/1.1 maps. A DITA map is an object that contains references to DITA topics and organizes the topics into hierarchies, tables, or groups. A map can contain any combination of references to DITA topics, DITA tasks, DITA concepts, DITA references, and other DITA maps. Maps are used to create documents that result in separate XML files in output processing. The <map> element is used in the XML to define a map. Authors can work with two types of maps: dynamic and static.

Authors can organize topics in a DITA dynamic map in Content Management and edit contents in an editing tool. Topic references in a DITA static map can be edited in an editing tool but not in Content Management. A DITA bookmap is a type of dynamic map. The <bookmap> element is used in the XML to define a bookmap.

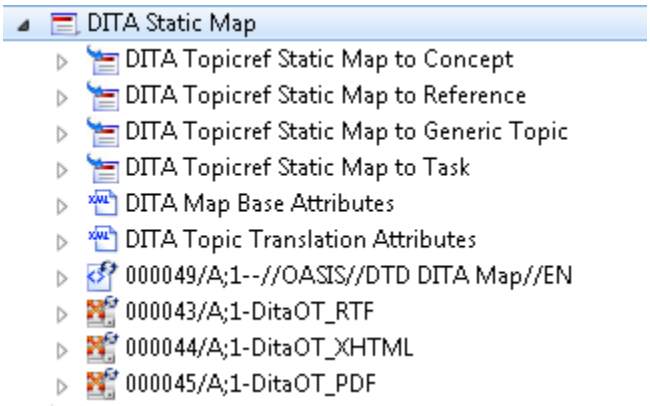
bi

The sample data includes the following topic types for DITA maps, shown here with their related objects.

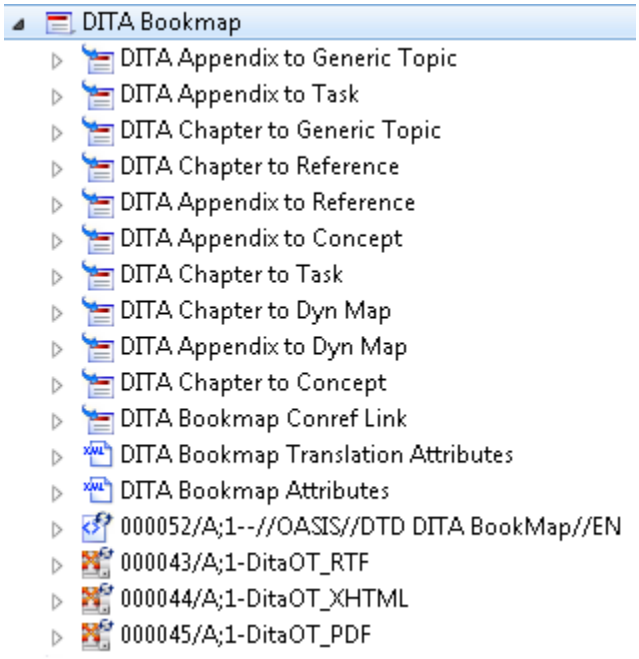
- DITA Dynamic Map



- DITA Static Map



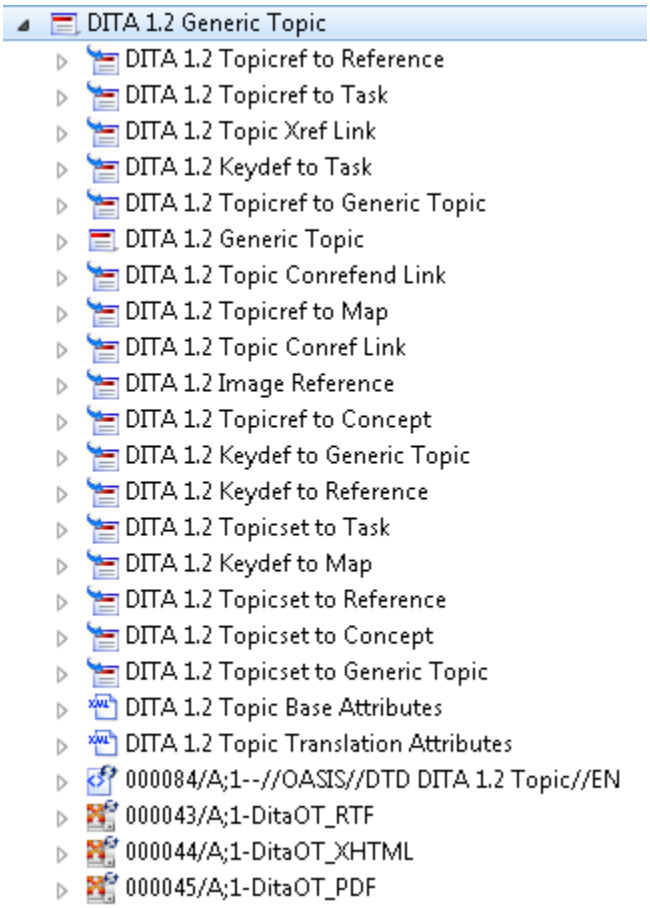
- DITA Bookmap



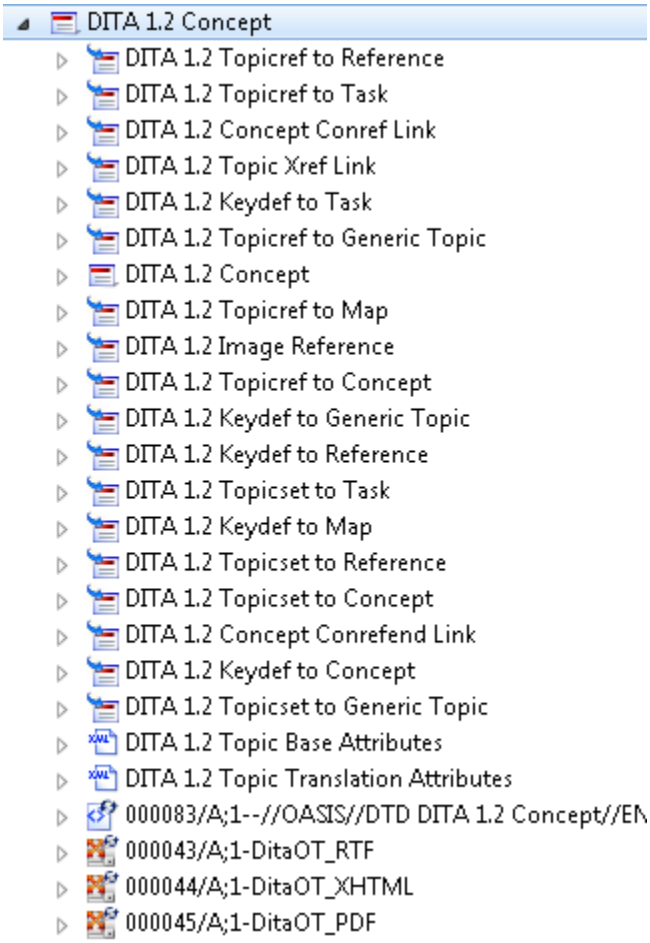
DITA 1.2 topic authoring objects

The DITA **administration data** includes the following topic types for DITA 1.2 topics, shown here with their related objects. You manage these topic types, along with the related objects required for them to function, so that authors can create and manage DITA topics.

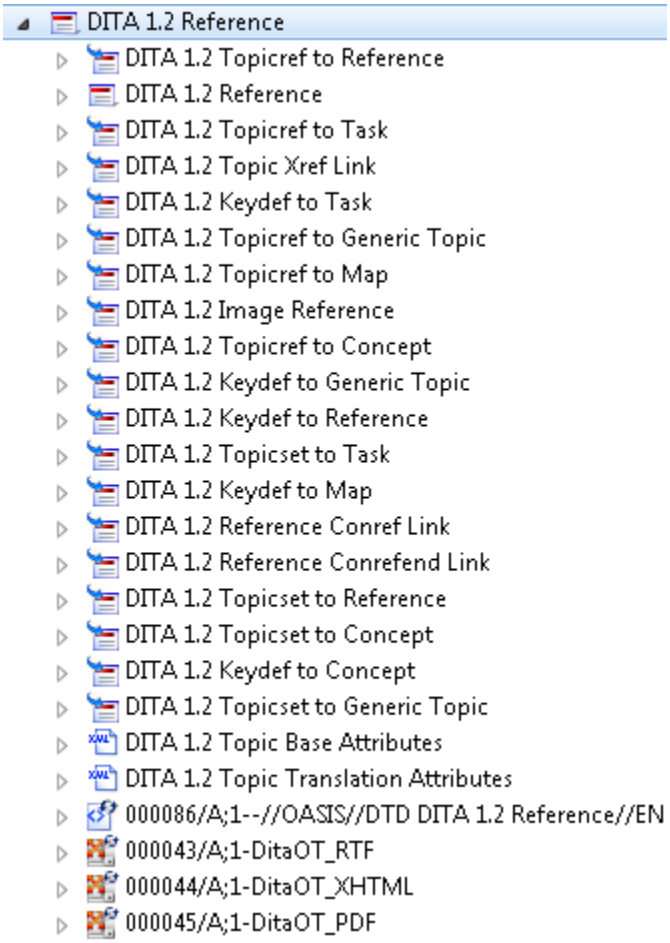
- DITA Generic Topic



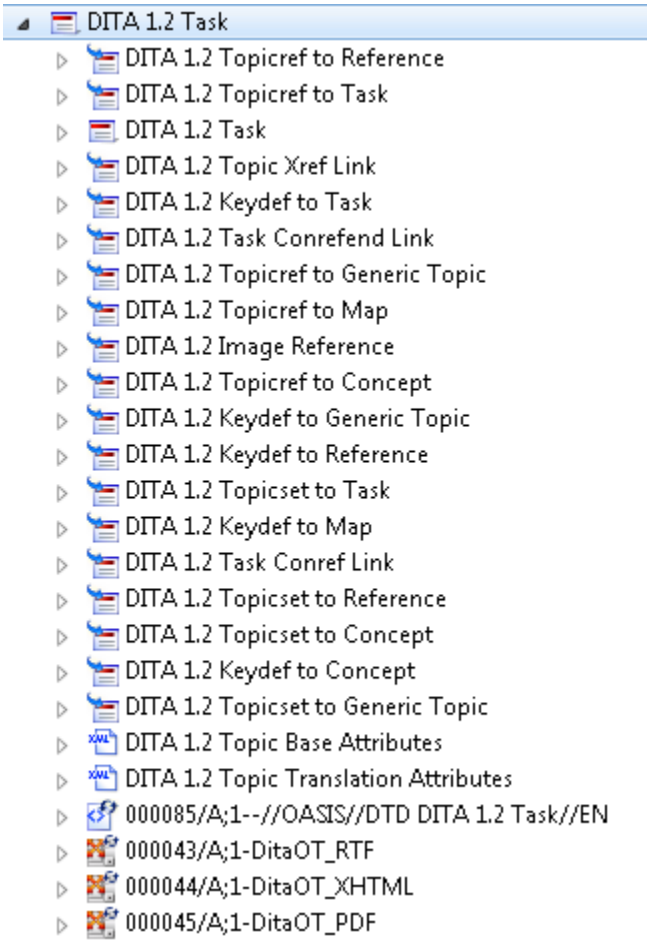
- DITA Concept



- DITA Reference



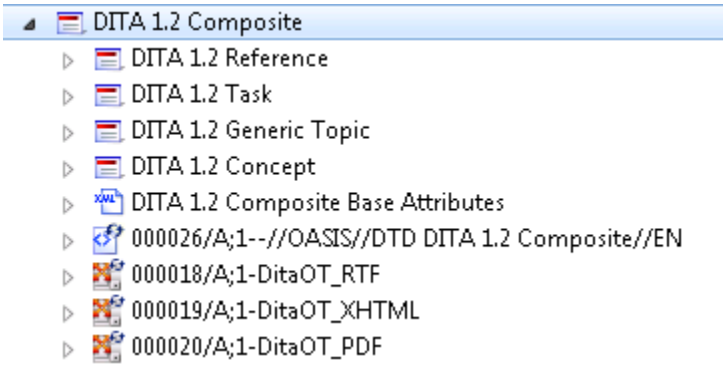
- DITA Task



DITA 1.2 composite authoring objects

The DITA **administration data** includes a topic type for DITA 1.2 composites. The DITA composite, or *DITA base*, object provides a top-level container for multiple topics when you create single content documents. It allows the creation of any sequence of generic topics, tasks, concepts, and references.

> Note:
>
> An XML attribute mapping for defining the translation attribute mappings is not necessary on DITA composite publication types, because DITA composites are not translated. Translations are performed on the topics inside the composite.
>
> When using the content of a DITA composite, the reference should be made to the top level topic within the DITA composite, not the composite itself. Only one top level topic should be included; all others should be nested or referenced.



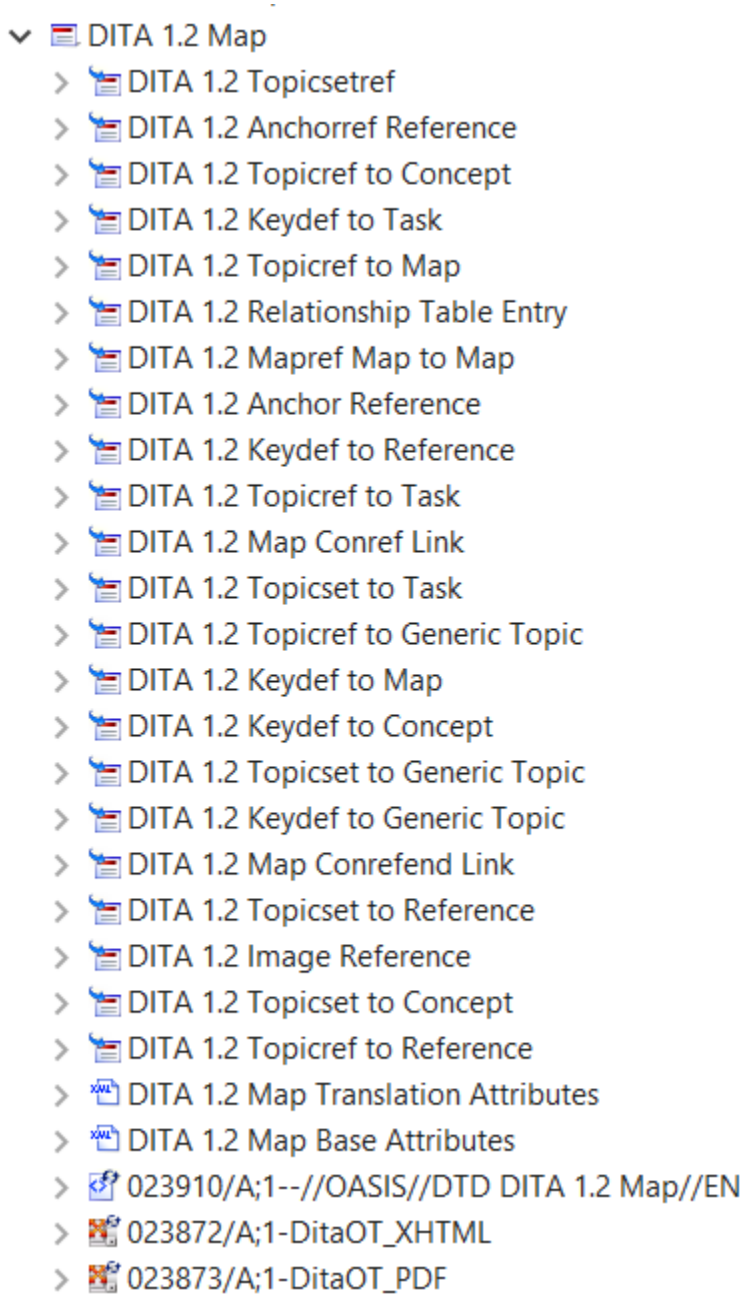
DITA 1.2 map authoring objects

The DITA **administration data** includes topic types for DITA 1.2 maps. A DITA map is an object that contains references to DITA topics and organizes the topics into hierarchies, tables, or groups. A map can contain any combination of references to DITA topics, DITA tasks, DITA concepts, DITA references, and other DITA maps. Maps are used to create documents that result in separate XML files in output processing. The <map> element is used in the XML to define a map.

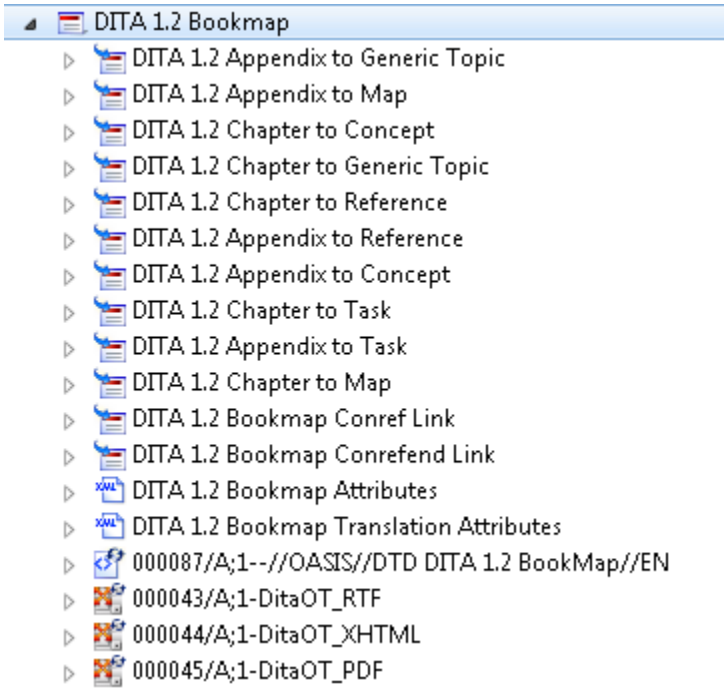
Authors can organize topics in a DITA dynamic map in Content Management and edit contents in an editing tool. A DITA bookmap is a type of dynamic map. The <bookmap> element is used in the XML to define a bookmap.

The sample data includes the following topic types for DITA maps, shown here with their related objects.

- DITA Dynamic Map

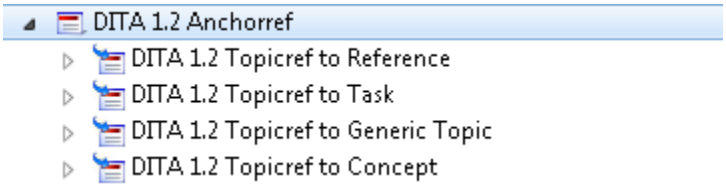


- DITA Bookmap



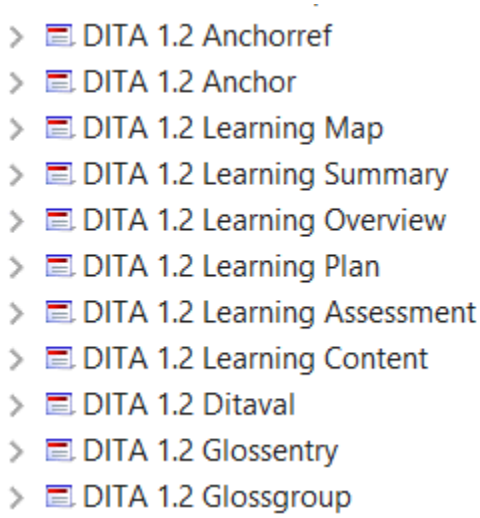
DITA 1.2 anchor and anchorref objects

The DITA **administration data** includes the DITA 1.2 anchor and anchorref topic types, shown here with their related objects. The anchorref objects enable authors to define a map fragment that is copied to the location defined by an anchor object.



Other DITA 1.2 topic types

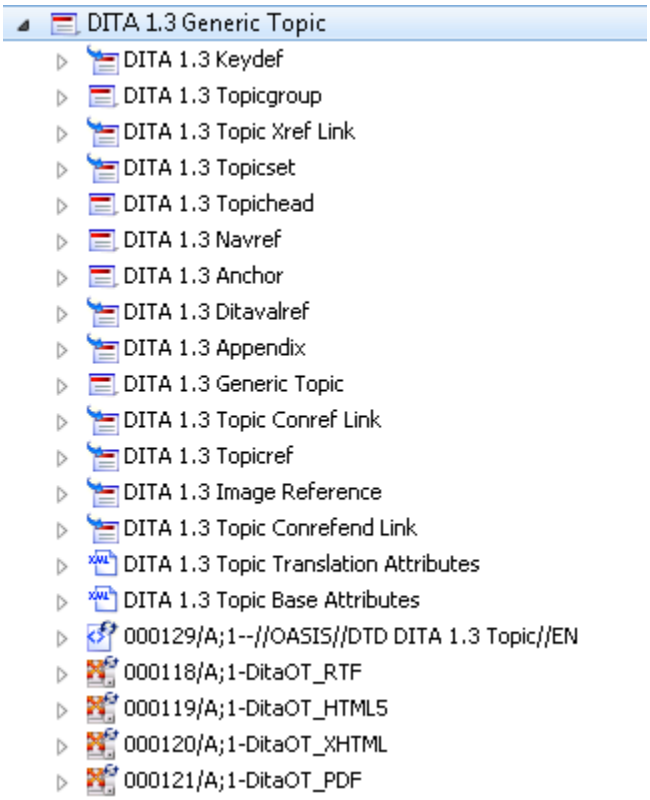
The DITA **administration data** includes several other DITA 1.2 topic types, shown here.



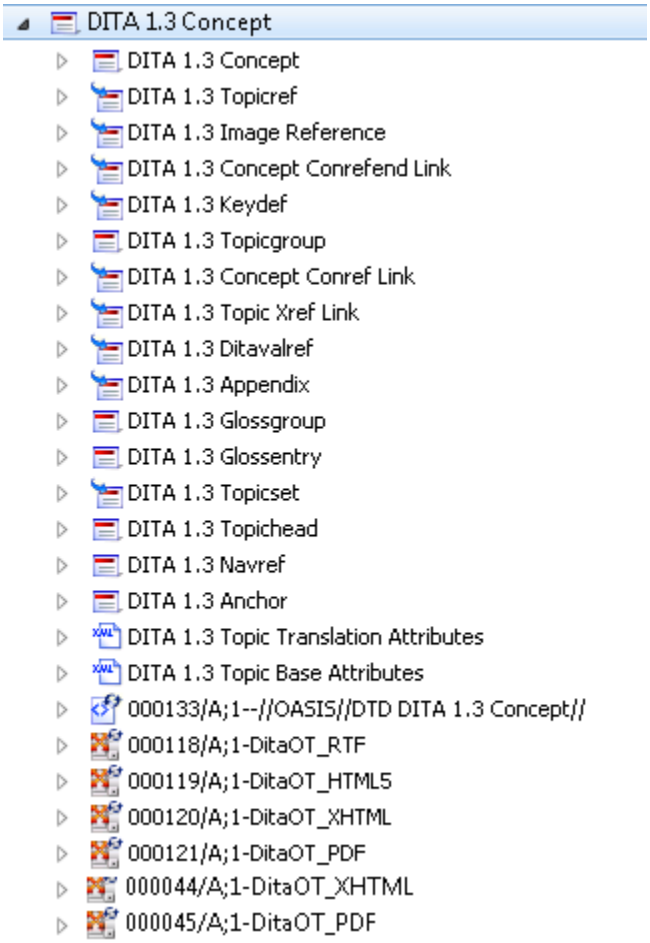
DITA 1.3 topic authoring objects

The DITA **administration data** includes the following topic types for DITA 1.3 topics, shown here with their related objects. You manage these topic types, along with the related objects required for them to function, so that authors can create and manage DITA topics.

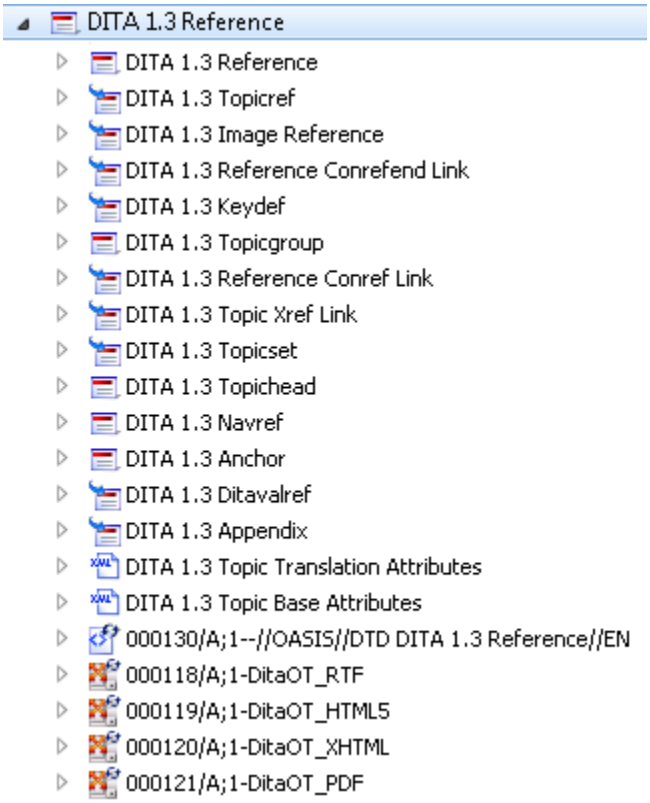
- DITA Generic Topic



- DITA Concept



- DITA Reference

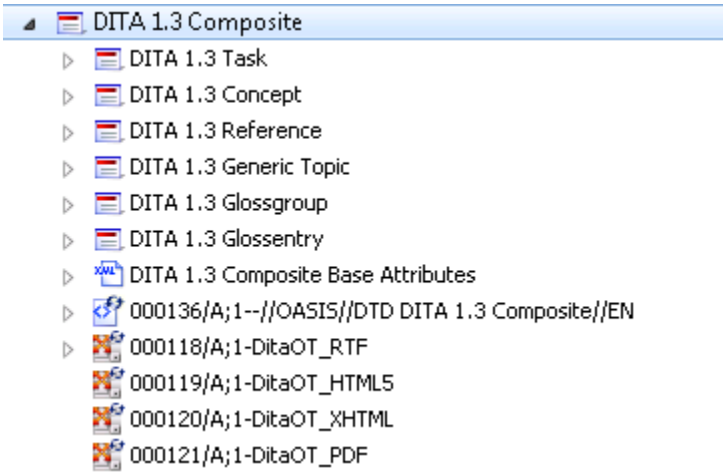


- DITA Task

DITA 1.3 composite authoring objects

The DITA **administration data** includes a topic type for DITA 1.3 composites. The DITA composite, or *DITA base*, object provides a top-level container for multiple topics when you create single content documents. It allows the creation of any sequence of generic topics, tasks, concepts, and references.

> Note:
>
> An XML attribute mapping for defining the translation attribute mappings is not necessary on DITA composite publication types, because DITA composites are not translated. Translations are performed on the topics inside the composite.
>
> When using the content of a DITA composite, the reference should be made to the top level topic within the DITA composite, not the composite itself. Only one top level topic should be included; all others should be nested or referenced.



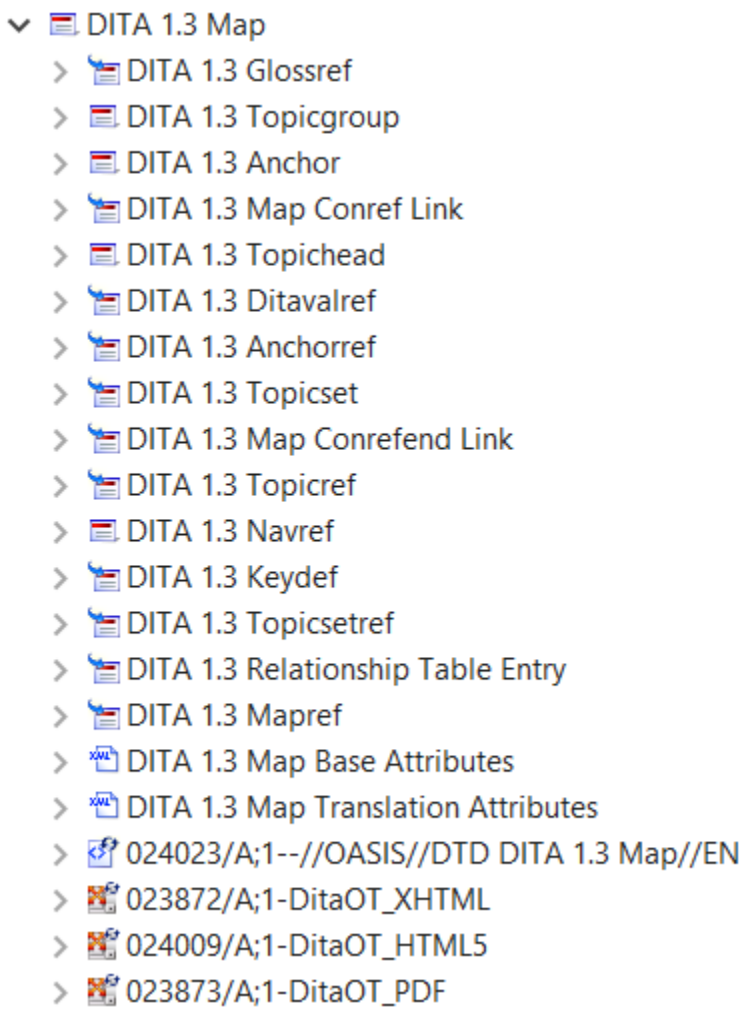
DITA 1.3 map authoring objects

The DITA **administration data** includes topic types for DITA 1.3 maps. A DITA map is an object that contains references to DITA topics and organizes the topics into hierarchies, tables, or groups. A map can contain any combination of references to DITA topics, DITA tasks, DITA concepts, DITA references, and other DITA maps. Maps are used to create documents that result in separate XML files in output processing. The `<map>` element is used in the XML to define a map.

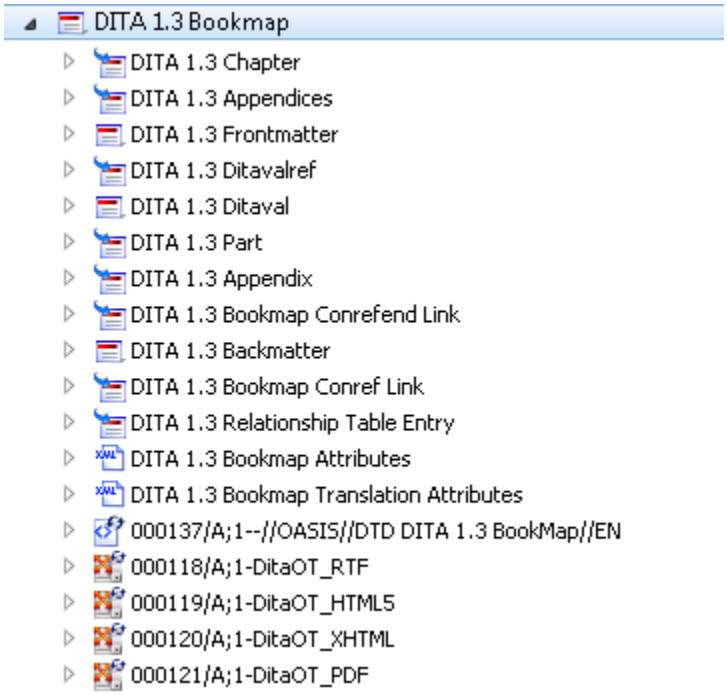
Authors can organize topics in a DITA dynamic map in Content Management and edit contents in an editing tool. A DITA bookmap is a type of dynamic map. The `<bookmap>` element is used in the XML to define a bookmap.

The sample data includes the following topic types for DITA maps, shown here with their related objects.

- DITA Dynamic Map

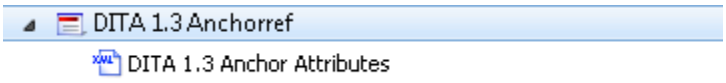


- DITA Bookmap



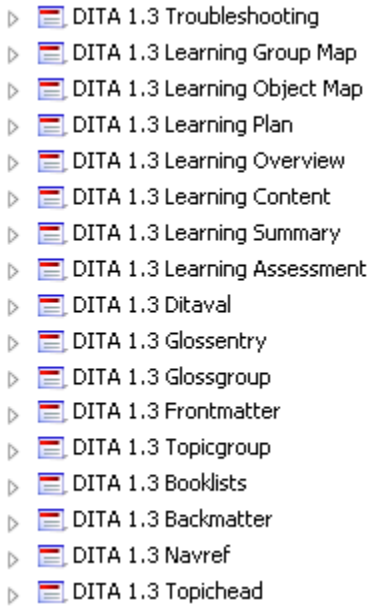
DITA 1.3 anchor and anchorref objects

The DITA **administration data** includes the DITA 1.3 anchor and anchorref topic types, shown here with their related objects. The anchorref objects enable authors to define a map fragment that is copied to the location defined by an anchor object.



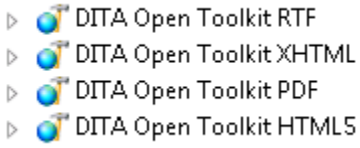
Other DITA 1.3 topic types

The DITA **administration data** includes several other DITA 1.3 topic types, shown here.



DITA publishing tools

The DITA **administration data** includes the following publishing tools, used for the DITA 1.0/1.1, 1.2, and 1.3 standards. Content Management supports version 2.4.6 of the DITA Open Toolkit (OT) publishing tools. You manage these publishing tools so that authors can view and publish DITA content.



Publishing Tool	Purpose
DITA Open Toolkit XHTML	Publishes DITA content to XHTML.
DITA Open Toolkit PDF	Publishes DITA content to PDF.
DITA Open Toolkit HTML 5	Publishes DITA content to HTML 5.

Note:

- Set the path for `DITA_ANT_HOME` in the *Dispatcher_Root***Module\Translators\-contmgmtpublish\config\contmgmtpublish_config.properties** file. The path must be set to the Ant directory in the DITA OT.

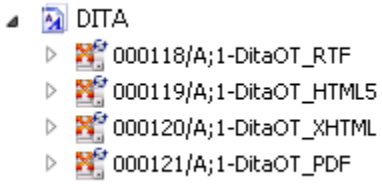
- To use an older version of DITA OT, copy the older version of DITA OT to: the *Dispatcher_Root***Module\Translators\contmgmtpublish\lib** folder.

DITA OT publishing tools support the following image formats, although .tif and .png formats may not be supported for PDF output:

- .jpg
- .gif
- .bmp
- .tif
- .eps
- .svg
- .png

DITA stylesheets

The DITA **administration data** includes the stylesheet objects for Content Management, under the **DITA** style type, used for the DITA 1.0/1.1, 1.2, and 1.3 standards.



Stylesheet	Used for
DitaOT_HTML5	Publishing DITA composite topics and maps to PDF, with the resulting content in HTML 5 format.
DitaOT_XHTML	Publishing DITA composite topics and maps to PDF, with the resulting content in XHTML format.
DitaOT_PDF	Publishing DITA composite topics and maps to PDF, with the resulting content in PDF format.

The actual stylesheet files are included with the DITA Open Toolkit installation on the Dispatcher and are stored in this directory:

```
Dispatcher\Module\Translators\contmgmtpublish\lib\dita-ot
```

The dataset attached to the stylesheets is the **build.xml** file, which is used by the ANT command. The ANT command is executed by the Dispatcher, which uses the stylesheets in the above directory. The **build.xml** file is located here:

```
\tc_root\contmgmtdita_data\data\admin\contmgmtdita_admindata\build.xml
```

You can modify the stylesheets for your organization's needs for viewing and publishing DITA content. To use different stylesheets for the same tool, you must create new style types. For example, you may have a style type for parts manuals that includes all the stylesheets used for that type of publication and another style type and stylesheets for work instructions.

Create a DITA value filter

A DITA value filter contains a table of values that are used to filter topics in a DITA map when you publish it, based on certain attributes in the topics. For example, you can exclude topics whose Audience attribute is Advanced or Expert, so that the map includes only those topics whose Audience is Beginner.

Administrators create these types of DITA value filters. Authors may create DITA value filters by creating a topic with the **Ditaval** topic type.

1. Select the folder in which you want to create the new topic type or publication type, and then either click **New Administrative Item...**, or choose **File→New→New Administrative Item**.

2. In the **New Administrative Item** dialog box, expand **Complete List**, click **DITA Value Filter**, and then click **Next**.

3. In the dialog box, in the **Name** box, type a name for the filter.

4. Click **Finish**.

5. Open the **Summary** view and select the DITA value filter.

6. Under **DITA Value Filter Table**, click **Add**.

7. In the dialog box, do the following to add a filter action:

From this list	Select this
DITA Action	The type of action on which the filter will be used.
DITA Attribute Name	The attribute on the objects that you want the filter to use.
DITA Value	The value assigned to the attribute that you want the filter to use. Note: Additional values can be added using Business Modeler IDE. Caution: DITA filter values cannot contain spaces.

8. Click **Finish**.

9. For each filter action you want to add, repeat the above two steps.

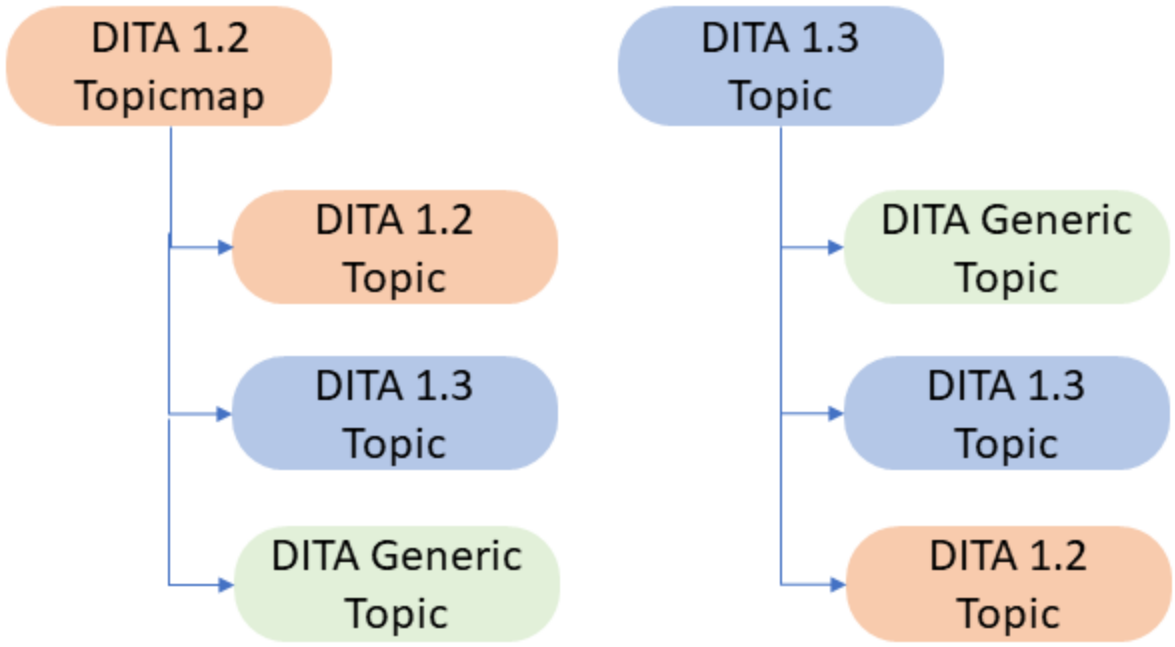
Enable support for multiple DITA topic versions in a DITA structure

By default, a DITA topic structure can contain only the same version of DITA topic. For example, a DITA 1.2 topicmap can contain only DITA 1.2 topic types.

To enable multiple DITA topic versions in a DITA structure:

- Set the value of the **Ctm0Allow_Multiple_Version_In_DITA_Structure** preference to **true**.

Example:



Editing DITA object lists of values (LOVs)

When authors create DITA items or apply DITA value filters, they can choose from predefined values for audience, platform, product, and other properties. Administrators set the allowed values for these properties using Business Modeler IDE.

Caution:

Values in DITA LOVs cannot contain spaces.

13. Managing objects for S1000D authoring

Creating objects for S1000D authoring

S1000D is an international specification for technical publications utilizing a Common Source Database. S1000D Content Management is an optional Teamcenter solution.

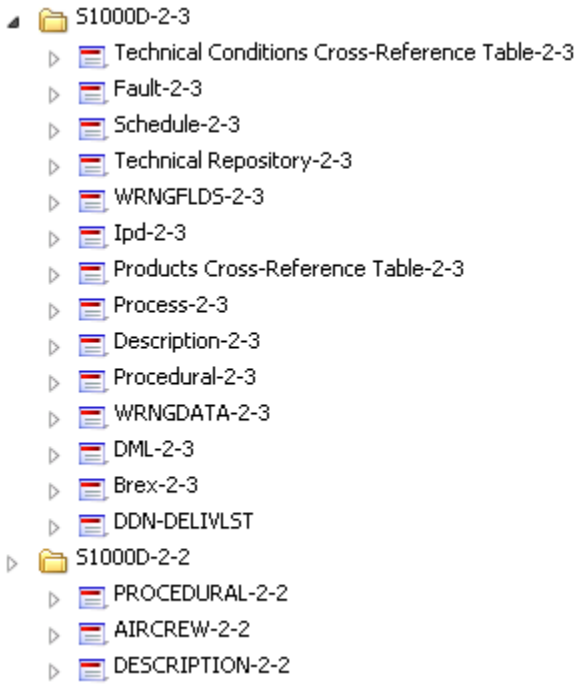
To create the required objects for managing documentation for the S1000D standard, you can import the S1000D **administration data** for S1000D issues 2.2/2.3, 4.0, 4.1, 4.2, or 5.0. These administration objects support the content types defined in the S1000D standard for these versions.

Note:

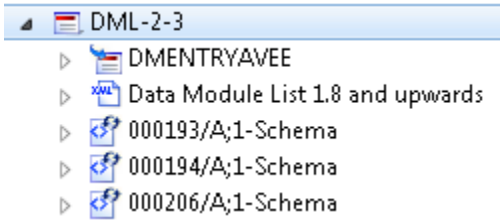
- Since S1000D data modules requires business rules exchange (BREX) reference to validate the data modules, ensure that you create BREX data modules so that you can successfully create S1000D data modules.
- It is recommended that you maintain an unchanged version of the provided S1000D objects for testing and comparison purposes.

S1000D 2.2 and 2.3 topic authoring objects

The S1000D **administration data** includes the following topic types in the **S1000D-2-3** and **S1000D-2-2** topic type groups. You manage these topic types, along with the related objects required for them to function, so that authors can create and manage S1000D topics.

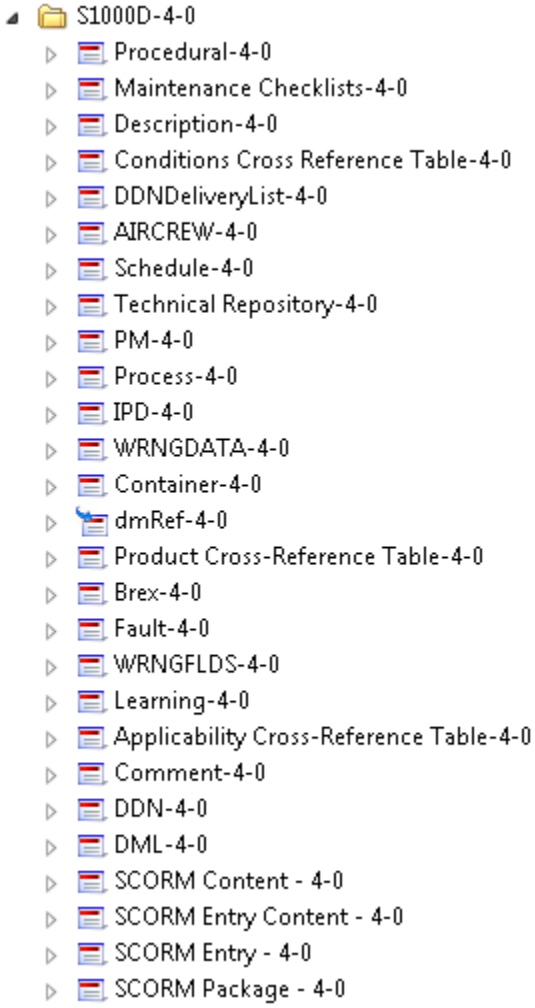


Each topic type has related objects. For example the data module topic type for version 2.3 has a related reference topic type, XML attribute mapping, and schemas:

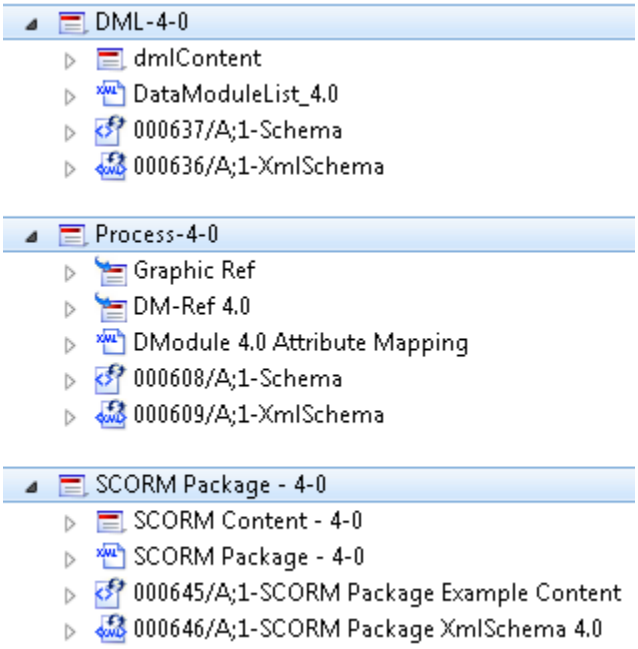


S1000D 4.0.1 topic authoring objects

The S1000D **administration data** includes the following topic types for the **S1000D-4-0** topic type group. You manage these topic types, along with the related objects required for them to function, so that authors can create and manage S1000D topics.

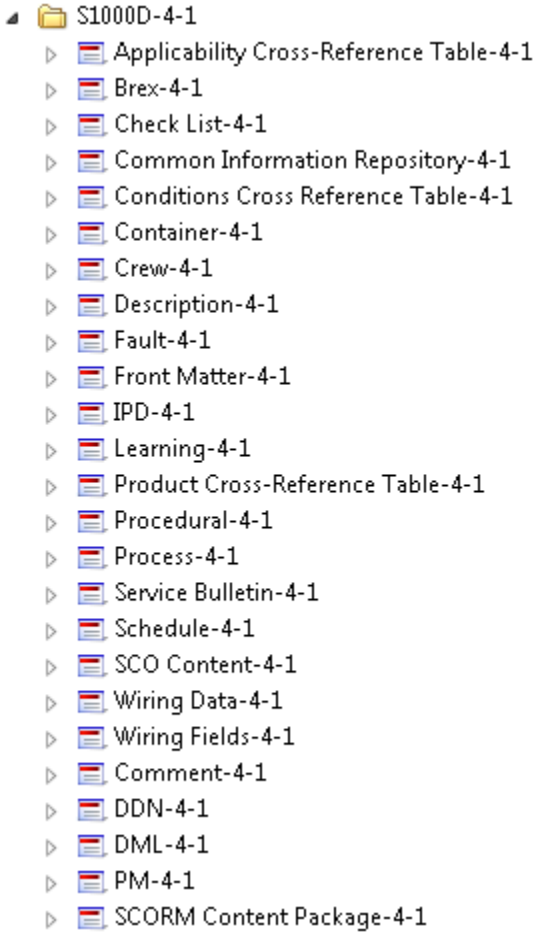


Each topic type has related objects. For example the data module, process, and SCORM package topic types for version 4.0.1 have related reference topic types, XML attribute mappings, and schemas:

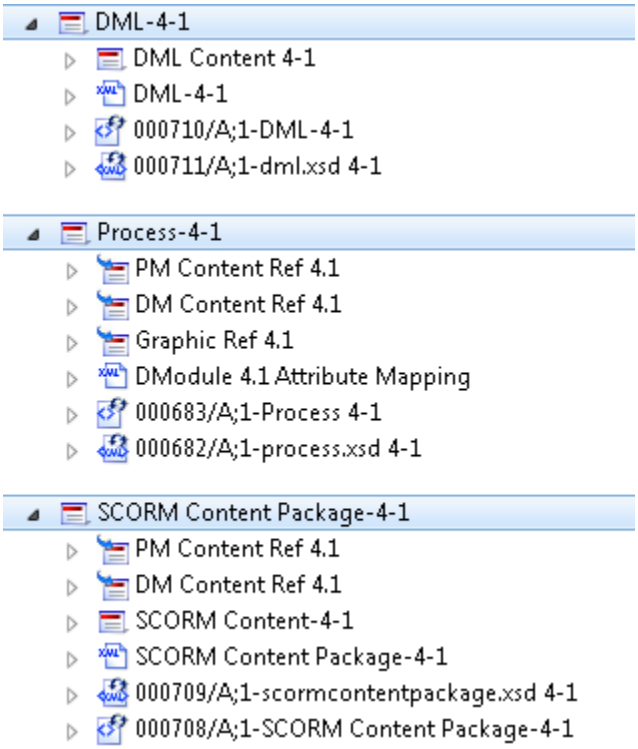


S1000D 4.1 topic authoring objects

The S1000D **administration data** includes the following topic types for the **S1000D-4-1** topic type group. You manage these topic types, along with the related objects required for them to function, so that authors can create and manage S1000D topics.



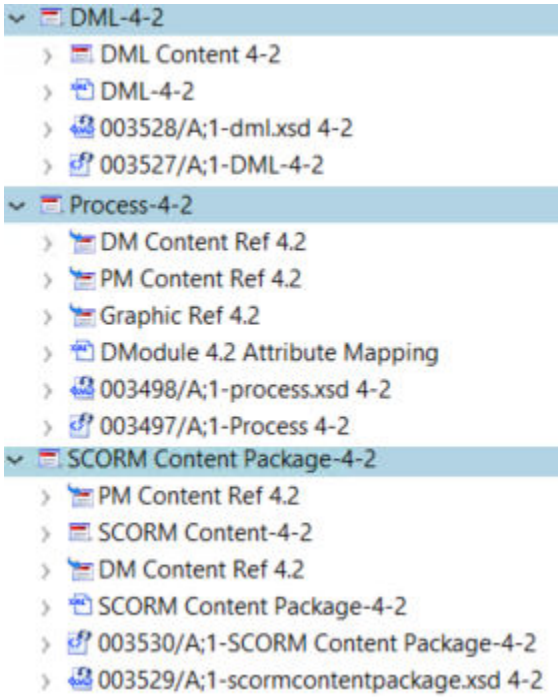
Each topic type has related objects. For example the data module, process, and SCORM content package topic types for version 4.1 have related reference topic types, XML attribute mappings, and schemas:



S1000D 4.2 topic authoring objects

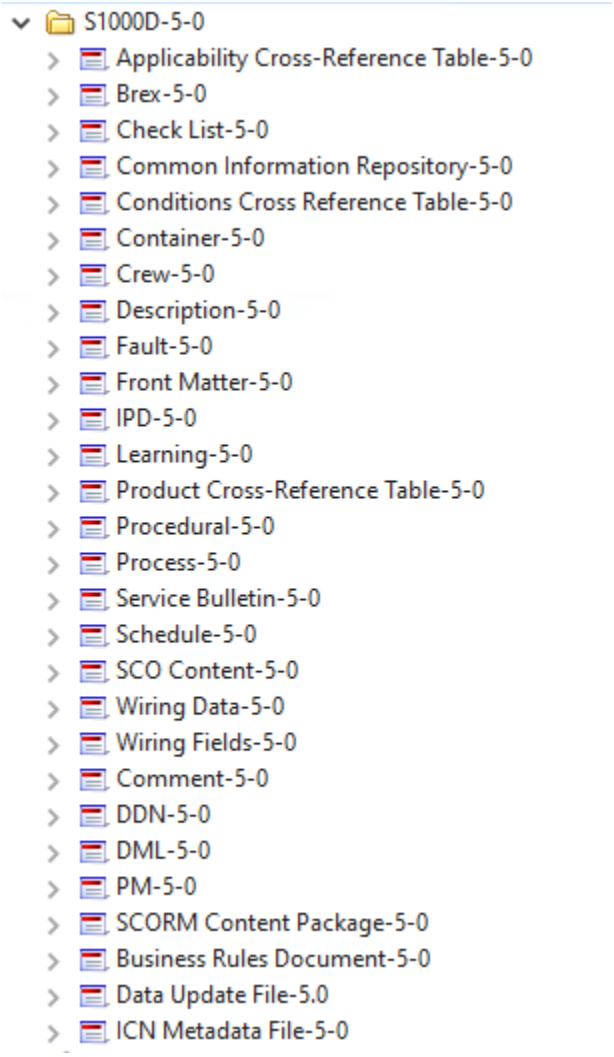
The S1000D **administration data** includes the following topic types for the **S1000D-4-2** topic type group. You manage these topic types along with the related objects required for them to function. This enables authors to create and manage S1000D topics.

Each topic type has related objects. For example, the data module, process, and SCORM content package topic types for version 4.2 have related reference topic types, XML attribute mappings, and schemas:

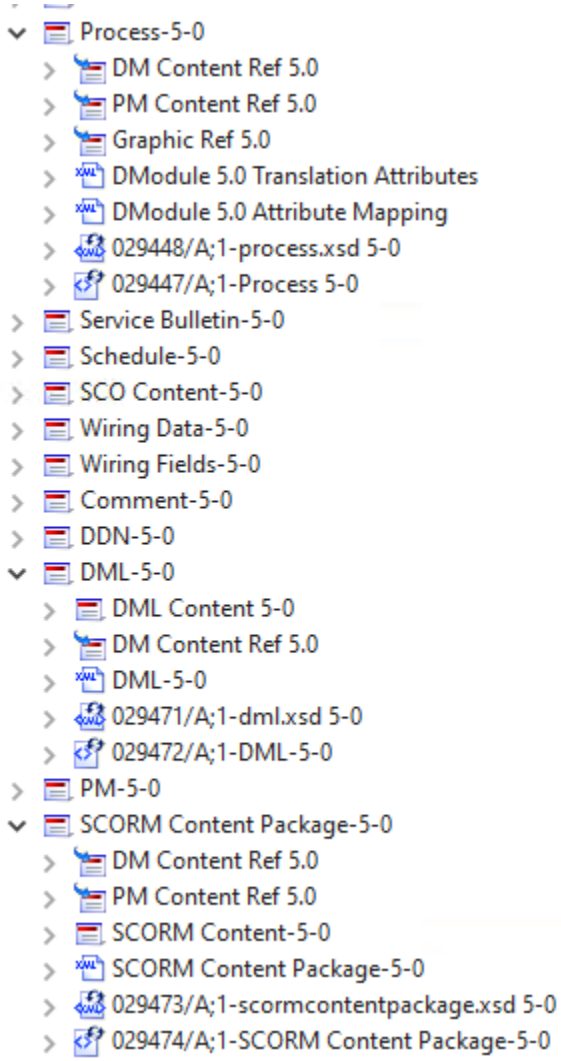


S1000D 5.0 topic authoring objects

The S1000D **administration data** includes the following topic types for the **S1000D-5-0** topic type group. You manage these topic types along with the related objects required for them to function. This enables authors to create and manage S1000D 5.0 topics.



Each topic type has related objects. For example, the data module, process, and SCORM content package topic types for issue 5.0 have related reference topic types, XML attribute mappings, and schemas:



Using S1000D standard numbering system (SNS) structures

An S1000D standard numbering system (SNS) structure defines the applicable product structure users document and the numbering scheme used in data modules for specific S1000D 4.0, 4.1, 4.2, and 5.0 projects. When you create a new data module in Content Management, you select a node of an SNS structure to apply its *data module code* (DMC) values and other data module detail values to the data module.

When you import data modules that use SNS codes, they are related to the appropriate node in the SNS structure based on that portion of their DMC. Imported graphics are also related to the node based on their SNS. Objects that don't use an SNS, such as data dispatch notes, publication modules, and graphics with commercial and government entity (CAGE) codes, are related to the top node of the SNS structure.

The SNS typically consists of a structure with the following nodes:

- Root node that contains the data module code values and other data module values, such as the originator name and responsible party details
- System nodes that specify the titles, codes, and descriptions for the various system levels of the structure
- Subsystem nodes that specify the titles, codes, and descriptions for the various subsystem levels of the structure
- Assembly nodes that specify the titles, codes, and descriptions for the various assembly levels of the structure
- Disassembly nodes that specify the titles, codes, and descriptions for the various disassembly levels of the structure

An SNS also defines data module *information codes*, which specify the data module types used in a project. In Content Management, information code objects include the S1000D standard three-character code, a name, a description, and the topic type reference for each data module type. For example, the information code for a 4.1 procedural data module is **121**. You may edit the name and description values for each information code.

Create, import, and edit S1000D SNS structures

You can create an S1000D standard number system (SNS) structure in Content Management, node by node, or create an SNS structure XML file and import it to Content Management.

A sample SNS structure and SNS schema are included in the administration data provided with the Content Management installation:

*TC_ROOT***contmgmts1000d40_data\data\admin\BIKESNS.xml**

*TC_ROOT***contmgmts1000d40_data\data\admin\sns.xsd**

Import an S1000D SNS structure

When you import an SNS structure, the entries in the XML file define the objects and relationships created in the database, including the SNS root node, related nodes, and information codes used for the specific project.

1. Select the folder in which you want to place the imported SNS structure, and then choose **Tools→Import→S1000D Project Data→Standard Numbering System (SNS) Structure**.
2. In the **Bulk Import** dialog box, browse to and select the file you want to import.
3. Click **OK**.

Import an S1000D SNS structure

1. Go to a folder and choose **More Commands** ••• > **New** > **Import Content**.
2. In the **Import Content** panel, from the **CONTENT TYPE** list, select **SNS**.
3. Click **Choose File** and select the SNS structure.
4. Click **Import**.

The SNS structure is imported into the folder you selected.

Create an S1000D SNS structure

1. Select the folder in which you want to create the new SNS.
2. Either either click **New Administrative Item...**, or choose **File→New→New Administrative Item**.
3. In the **New Administrative Item** dialog box, expand **Complete List**, click **Standard Numbering System Root Node**, and then click **Next**.
4. In the dialog box, enter the data for the SNS root node you are creating, according to the S1000D standard.

   For more information on the S1000D standard, see *www.S1000D.org*.
5. Click **Finish**.
6. Add nodes to the structure using **File→New→New Administrative Item→Standard Numbering System Node**.

Edit an SNS structure

1. In the **Home** component view, select the SNS structure you want to edit.
2. Open the **Summary** view.
3. Check out the SNS structure.
4. Edit the properties of the SNS structure in the **Summary** view.
5. Check in the SNS structure.

Enable publishing of CGM graphics in S1000D content

Graphics in the computer graphic metafile (CGM) format are used in S1000D publications. If you are using the publishing tools provided in the Content Management installation, you must take steps for CGM graphics to display in published output. If you are using other publishing tools, these steps may not apply.

Enable graphics to display in XHTML output

- To enable graphics to display in XHTML output, install a CGM viewer on the client.

Enable graphics to display in PDF output

1. Download these JAR files:

   > Note:
   >
   > These links are current as of the time of publication. They may differ from the links listed here.

   - **jai_imageio_windows_i586.jar**

     This file is available from:
     http://www.oracle.com/technetwork/java/javasebusiness/downloads/java-archive-downloads-java-client-419417.html#7380-JAI-1.1.1_01-oth-JPR

     Select **Windows Signed Auto-Install** (**jai_imageio-1_0_01-windows-i586-jar.zip**).

   - **jcgm-core-0.2.0.jar**

     This file is available from: **http://jcgm.sourceforge.net/download.html**

     Select **jcgm-core-0.2.0-bin.zip**.

   - **jcgm-image-0.1.1.jar**

     This file is available from: **http://jcgm.sourceforge.net/download.html**

     Select **jcgm-image-0.1-1-bin.zip**.

2. Save the JAR files to this folder in the dispatcher root directory:

   ```
   Dispatcher\Module\Translators\contmgmtpublish\lib
   ```

Configure Publication Structure columns for S1000D content

When authors work with the S1000D standard, you can configure the **Publication Structure** view so that it contains default columns that are useful for working with S1000D objects. To do this, you can import the preferences XML file provided with the Content Management installation that defines the three preferences for controlling the column configuration in the view. You can modify this file before you import it to meet your organization's needs. Authors can also change the column configuration in the **Publication Structure** view to meet their individual needs. If authors change their column configurations before you import the preference file, the imported preferences do not affect the columns configurations in their environments.

1. Review the preference XML file for the definitions it contains, at this location:

   *TC_ROOT***contmgmts1000d40_data\data\admin\contmgmts1000d40_column_heading_preference_override.xml**

   The file sets these columns as the default:

   - Item Id
   - Document Title
   - In Work
   - Issue Number
   - Topic Type

2. Modify the file if necessary to meet your organization's needs.
3. Choose **Edit→Options**.
4. In the **Options** dialog box, click **Search** at the bottom of the dialog box.
5. In the **Preferences by Search** pane, click the **Import** tab.
6. From the **Import File Name** box, browse for and select the preference file:

   *TC_ROOT***contmgmts1000d40_data\data\admin\contmgmts1000d40_column_heading_preference_override.xml**

7. From the **To Location** box, select **Site**.
8. Click **Import**.

Tip:

To verify the import, search for the **ContentManagerColumnsShownPref** preference and verify that the values represent the columns defined in the XML preference file.

Modifying S1000D stylesheets

S1000D XSL stylesheets are used to transform S1000D documents (data modules or publication modules that contain data modules) to PDF or XHTML for viewing and printing. XSL stylesheets are provided with the Content Management installation and stored on Teamcenter Dispatcher.

When you publish S1000D documents, the Dispatcher generates output as follows:

PDF output:

1. The S1000D XML is transformed into DocBook XML using the **s1000dtodb** stylesheet.
2. The DocBook XML is transformed into an XSL-FO document using the **dbtofo** stylesheet.
3. The XSL-FO document is processed into the PDF file, using a formatting program such as Formatting Objects Processor (FOP).

XHTML output:

1. The S1000D XML is transformed into DocBook XML using the **s1000dtodb_xhtml** stylesheet.
2. The DocBook XML is transformed into XHTML using the **dbtoxhtml** stylesheet and the Xalan program.

To customize the stylesheets, update or add stylesheets on the Dispatcher in the following subfolders of the **...\Translators\contmgmtpublish\lib\s1000dpublish** folder.

- For PDF output, use the stylesheets in the **s1000dtodb** and **dbtofo** folders.
- For XHTML output, use the stylesheets in the **s1000dtodb** and **dbtoxhtml** folders.

If the build process must be modified, update the **build_pdf.xml** or **build_xhtml.xml** files by editing the **S1000D PDF** or **S1000D XHTML** stylesheets in Content Management, by using **Edit→Edit Stylesheet**.

14. Creating schema objects

Overview of schemas in Content Management

You use schemas to define the structure of elements and attributes that can be used in the XML or SGML files created by the authors in your organization. In Content Management, a schema object corresponds to a schema file that you import to the database when you create the schema object. XML topic types are not required to have a related schema, but it is recommended to always relate at least one. The topic type is related to the schema *revision*. A schema may also be referenced as an external entity with another schema. Each SGML topic type must have three related schemas: an SGML declaration, an SGML DTD, and an SGML catalog file.

You can create your own schemas or use standard schemas, such as those from the S1000D standard. You import the S1000D standard schemas when you import the S1000D **administration data**.

Content Management supports several methods for defining XML and SGML document structure. You may choose from the following schema types:

- **DTD**

  A DTD is used to validate XML content. It is optional for XML topic types, and it is not used for SGML topic types (see SGML DTD below). If no DTD is present, XML content is not validated, and some editing tools may not work correctly.

- **Character entity map**

  A character entity map enables the replacement of certain characters with character entities during attribute mapping. It is optional for both XML and SGML. If no character entity map is present, a default character entity map is used.

- **Example content**

  Example content is used as default content for newly created topics. It is optional for both XML and SGML. If no example content is present, newly created topics will be based on the DTD only.

- **SGML DTD and declaration file**

  The SGML DTD and SGML declaration file are used to validate SGML content. They are required for SGML topic types. They are not used for XML topic types.

- **SGML catalog file**

  The SGML catalog file lists all supporting files necessary to parse an SGML file. It is required for SGML topic types. It is not used for XML topic types.

- **XML Schema**

  An XML schema is used to validate XML content similarly to DTDs, with additional capabilities.

Importing a schema

You can import a schema two ways:

- Import the schema when you **create a schema object**.
- Import the schema using the PLM XML method. This is useful for transferring between test and production systems. The data models must be the same between the two systems.

Create a schema object

1. (Optional) Select the folder in which you want to create the new schema.
2. Either click **New Administrative Item...** , or choose **File→New→New Administrative Item**.
3. In the **New Administrative Item** dialog box, expand **Complete List**, click **Schema**, and then click **Next**.
4. Do the following:

For this option	Do this
ID	Either type an ID for the schema, or leave the box blank so that the next available item ID is automatically assigned.
Revision	Either type a revision for the schema, or leave the box blank so that the next available revision is automatically assigned.
Name	Type the name for the schema.
Public ID	Type the public ID for the schema, which must correspond to the public ID as it is defined in the schema file. This cannot be modified after the schema has been created.
Schema Type	Select the type of schema you are creating.
Select Content	Click **Browse** to select the content file of the schema.

5. Click **Finish**.

Create an XML schema object

1. (Optional) Select the folder in which you want to create the new XML schema.

2. Either click **New Administrative Item...** , or choose **File→New→New Administrative Item**.

3. In the **New Administrative Item** dialog box, expand **Complete List**, click **XML Schema**, and then click **Next**.

4. Do the following:

For this option	Do this
ID	Either type an ID for the XML schema, or leave the box blank so that the next available item ID is automatically assigned.
Revision	Either type a revision for the XML schema, or leave the box blank so that the next available revision is automatically assigned.
Name	Type the name for the XML schema.
Public ID	Type the public ID for the XML schema, which must correspond to the public ID as it is defined in the schema file. This cannot be modified after the schema has been created.
Default Prefix	Prefix that represents the namespace URI, if a namespace URI is used.
Schema File Name	File name of the schema document that has a target namespace, if a namespace is used.
Select Content	Click **Browse** to select the content file of the XML schema.

5. Click **Finish**.

15. Creating topic and publication types

Overview of topic and publication types

In Base Content Management, a publication type is the topmost object in the underlying XML structure that defines how authors can assemble publications. A topic type is a component of a publication type that defines a single node in a publication structure definition. You create publication and topic types for each type of topic and publication that authors may create.

In S1000D and DITA Content Management, topic types are predefined to match the objects for the standard, for the versions of the standard that are supported, for example, an S1000D data module or a DITA concept topic. However, before content authors can create these objects, you must create these topic types in your system or import them from the Content Management **administration data** or data from a prior installation.

The structures of the publication and topic types that you create must match the XML tag structure defined in the DTDs or schemas. The schema revision that is related to a topic or publication type defines the valid structure of the content when it is edited and published.

The dialog box you use to create a topic type is the same for all types, Base, S1000D, and DITA. To specify the type of object, you select the appropriate class in the **Apply classname** option in the dialog box.

A topic type is related to other topic and publication types, **stylesheet revisions**, an **XML attribute mapping**, and a **schema revision**. Topic types are related to a different stylesheet revision for each method in which they are published. For example, a topic type may be related to one stylesheet revision for publishing to HTML and another for publishing a PDF.

> Note:
>
> To relate a topic type to itself, you must use the **Copy→Paste** option instead of dragging the topic type.

You can also create reference types used to contain extra data used for building links and references to topics, such as for the functioning of graphics within content.

> Tip:
>
> Reference types are required for the functioning of graphics within content. You must have a **Content Graphic Reference** reference type related to each topic type.

You can create topic type groups to use with the **ctm0TopicTypeGroup** preference to limit the topic types that can be used for creating new topics.

Create a topic type or publication type

1. Select the folder in which you want to create the new topic type or publication type, and then either click **New Administrative Item...** , or choose **File→New→New Administrative Item**.

2. In the **New Administrative Item** dialog box, expand **Complete List**, click **Topic Type** or **Publication Type**, and then click **Next**.

3. In the dialog box, do the following:

For this option	Do this
Name	Type the name of the topic type or publication type to appear in the structure tree.
Root Element Name	Type the root tag name of the topic type or publication type, which must correspond to the root tag as it is defined in the schema related to the topic type. When a document is parsed, this identifies where to start the new topic object and the type of object to create. If multiple topic types within the same structure have the same **Local Tag Name**, enter the **Topic Type Condition** below, to distinguish the topic types.
Usage	Select one of the following: • **SYSTEM** The topic type will *not* appear in selection lists in dialog boxes, such as those for creating topics. Note: This is used for special purposes only, such as for adding new objects that may not be ready for general use or for objects to be used only by system processes, such as workflows. • **USER** The topic type will appear in selection lists in the appropriate dialog boxes, such as those for creating topics.
Validate Incoming	Select either **True** or **False**, to specify whether data being saved to the database is validated when content is decomposed. If **True** is selected, invalid content produces an error.

For this option	Do this
	Typically, this type of validation is not required if your editing tool validates content, and it may affect processing time.
Validate Outgoing	Select either **True** or **False**, to specify whether data being retrieved from the database is validated when content is composed. If **True** is selected, invalid content produces an error. Typically, this type of validation is not required if your editing tool validates content, and it may affect processing time.
Validate Example Content	If the topic type is related to a schema with the EXAMPLE_CONTENT schema type, select either **True** or **False**, to specify whether the example content is validated when it is used during compose. If **True** is selected, invalid content produces an error. Typically, this type of validation is not required if your editing tool validates content, and it may affect processing time.
Class Name Applied	Select the Teamcenter subclass of topic that is associated with this topic type. The S1000D and DITA entries are available only if the corresponding solution is installed. • **Topic** (Base topic) • **Publication** (Base publication) • **S1000D Data Dispatch Note** • **S1000D Data Module** • **S1000D In Process Review Form** • **S1000D Publication Module** • **S1000D Data Module List** • **S1000D Commentary** • **DITA Dynamic Map** • **DITA Static Map**

For this option	Do this
	• **DITA Topic** • **DITA Task** • **DITA Base** • **DITA Reference** • **DITA Concept** • **Non-topic Item** Note: This supports Teamcenter items and subclasses.
Non-topic Class Names	Add the Teamcenter subclass of objects that are associated with this topic type.
(For DITA solution only) **Transfer Mode**	Select the appropriate transfer mode for the topic type, to control how referenced topics are included during compose: • For DITA map topic types, select the **cdi0_DITAMultiContent** transfer mode. • For DITA composite, topic, concept, reference, and task topic types, select the **cdi0_DITASingleContent** transfer mode.
File Extension	Select the appropriate extension to be applied to topics of this topic type.
Namespace URI	Type the URI that serves as a unique identifier for the namespace. Tip: Namespaces allow different XML vocabularies to be used in a single XML document and provide an XML parser a means to categorize each vocabulary. Although this is a URI, it is not used to identify and retrieve a Web address.

For this option	Do this
Default Namespace Prefix	Type the prefix that represents the namespace URI. The prefix is added to the beginning of element names to place them in the namespace.
(When creating a topic type) **Topic Type Condition**	If multiple topic types within the same structure have the same **Local Tag Name**, type a **topic type condition** value to restrict the topic type, using either **containspath**, **insidepath**, or **containsvalue**, along with a specified value or path. If the **Local Tag Name** of all topic types in a structure is unique, this attribute is not used.

4. Click **Finish**.

5. **Relate** the following objects to the topic or publication type:

   - Relate the topic or publication type to other topic and publication types to create a **publication structure**.

     > Tip:
     >
     > To relate a topic type to itself, you must use the **Copy→Paste** option instead of dragging the topic type.

   - Relate the **schema revision** that applies to the topic or publication type.
   - Relate the **stylesheet revision** that applies to the topic or publication type.
   - Relate the **XML attribute mappings** that apply to the topic or publication type.

Topic type conditions

In cases where you want to define multiple topic types using the same tag name, you must use a topic type condition to distinguish one topic type from another when you **create a topic or publication type**. This enables the system to identify topic types that begin with the same initial element and delineate them based on lower level elements or attributes.

If a topic type condition is defined, it will be used by the system to determine which topic types are assigned to topics when they are imported or when they are decomposed to be saved to the database after being created in an XML editor.

In all cases, the syntax used within the condition field is the name of the condition, followed by parenthesis which contain the values. Examples are provided below.

You may use the following topic type conditions:

Condition	Function	Example
containspath	This condition causes the system to assign a topic type if the XML in a topic has a specific series of elements or tags that contain a specific path.	To differentiate the topic type **dmodule-age** with element **dmodule** from the alternative **dmodule-avee**, this topic type condition is assigned in the **Topic type condition** box: Example: `containspath(/dmaddress/dmc/avee)` If a topic with the following XML is imported, since it contains the path specified in the condition, the **dmodule-avee** topic type is assigned: `<dmodule> <idstatus> <dmaddress> <dmc> <avee> <modelic>S1000DBIKE</modelic> </avee> </dmc>` Example: `containspath(/dmaddress/dmc/age)` If a topic with the following XML is imported, since it contains the path specified in the condition, the **dmodule-age** topic type is assigned: `<dmodule> <idstatus> <dmaddress> <dmc> <age> <modelic>S1000DBIKE</modelic> </age> </dmc>`
containsvalue	This condition causes the system to assign a topic type if the XML in a topic has an element or attribute that contains a specific value. For example:	To differentiate the topic type **dmodule-dmc** with the element **dmodule** from the alternative **dmodule-arv**, this topic type condition is assigned in the **Topic type condition** box: `containsvalue(/ddnfilen,DMC.*)`

Condition	Function	Example

	`containsvalue(/tag/subtag/subtag/subtag/@attribute,ref)`
	or
	`containsvalue(/path/tag,textstring)`
		If a topic with the following XML is imported, since it contains a **ddnfilen** element that contains text that starts with DMC, the **dmodule-dmc** topic type is assigned.
		`<dmodule><ddnfilen>DMC-111-333-444-555.SGM</ddnfilen>`

insidepath	This condition causes the system to assign a topic type if the XML in a topic has a specific element or tag that is inside a specific parent element or a specific path.	For the topic type **refdmavee** with the element **avee**, this topic type condition is assigned:
		`insidepath(/refdm)`
		If a topic with the following XML is imported, since it contains the **avee** element inside the **refdm** element, the **refdmavee** topic type is assigned.
		`<refdm><avee><modelic>S1000DBIKE</modelic><sdc>AAA</sdc></avee></refdm>`

Create a topic type group

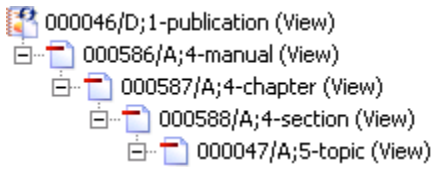
You can create topic type groups to use with the **ctm0TopicTypeGroup** preference to limit the topic types that can be used for creating new topics.

1. Select the folder in which you want to create the new topic type group, and then either click **New Administrative Item...**, or choose **File→New→New Administrative Item**.
2. In the **New Administrative Item** dialog box, expand **Complete List**, click **Topic Type Group**, and then click **Next**.
3. In the dialog box, type a name for the group in the **Name** box.
4. Click **Finish**.
5. Relate topic types to the topic type group so that the topic type group contains all the topic types you want to be available when creating a topic.

16. Creating a publication structure

What is a publication structure?

A *publication structure* is generally described as a set of topics related together to make a larger document, such as a publication with a header and footer, chapters, sections, and components, as in this example:



Administrators relate publication and topic types to define the structures that authors can use to create documents. The structure must match the composition of schemas related to the corresponding topic types. If a topic type has no related schema, you cannot edit or publish topics at that level in the structure.

Once a publication structure is created, when content authors create a new publication with the publication type defined in the structure, they are restricted to adding only topics with the defined topic types at the allowed locations in the structure.

Define a structure for publication authoring

1. In the **Home** component view, **create the topic and publication types** that match the elements in your organization's schemas.

2. **Relate** the publication and topic types to create a structure that matches the hierarchy of the schemas.

17. Enabling references and links

Overview of reference types

Create reference topic types for use with XML content to identify the syntax used for building links and references to topics. Reference topic types are related to topic types and XML attribute mappings and hold values that include the file or content and context of the target of a link. Several reference types are supplied with the DITA and S1000D **administration data**. The syntax of these reference types match the syntax requirements of the related DTDs and schemas.

You can create these reference topic types:

- **Content Topic Reference**

  A reference type typically used for **enabling navigation links**. The reference is created by pasting a link generated in the Teamcenter client, into content in the editing tool. Authors generate links in the Teamcenter client by using the fixed and floating navigation links copy options.

  Content references are also used for S1000D documentation.

  > Example:
  >
  > To create a content reference, you create reference type REFDMAVEE with the tag name AVEE, which is a tag name used in data modules. You then relate reference type REFDMAVEE to topic type REFDM. You create an XML attribute mapping for a specific function and relate it to REFDMAVEE. Then when you create a data module with the topic type REFDM, it contains the reference AVEE as specified by the XML attribute mapping.

- **Content Cross Reference**

  A reference type with the same functionality as **Content Topic Reference**. You can use this if you need to differentiate between two different types of content references.

- **Composable Topic Reference**

  A reference type typically used to enable the creation of DITA dynamic maps and S1000D data dispatch notes, data module lists, and publication modules. Authors create these types of references when they relate topics in the Teamcenter client. References are added to content during the compose process.

- **Content Graphic Reference**

  A topic-to-graphic reference type. The reference is created by pasting a link generated in the Teamcenter client, into the content in the editing tool. Authors generate links in the Teamcenter client by using the **Copy For Editor** option.

This reference type is required for the functioning of graphics within content. You must have a **Content Graphic Reference** reference type related to each topic type. A graphic reference type is included in the **administration data**; however you may need to modify it to work with your organization's schemas depending on the element, attribute, or reference type that you use to create those links.

- **Composable Graphic Reference**

  A topic-to-graphic reference type. The reference is created by relating the topic and graphic in the Teamcenter client. The reference is then added to the content by the compose process.

Create a reference topic type

1. Select the folder in which you want to create the new reference type, and then either click **New Administrative Item...**  , or choose **File→New→New Administrative Item**.

2. In the **New Administrative Item** dialog box, expand **Complete List**, click **Reference Topic Type**, and then click **Next**.

3. In the dialog box, do the following:

For this option	Do this
Name	Type a name to assign to this reference type. This name appears in the **Reference Topic Type** box when authors create a DITA object and relate DITA objects, and it can be edited later in the object's properties.
Local Tag Name	Type the root tag name to be used for the reference. The tag name must be valid for in the topic type(s) in which you insert the reference. If multiple topic types within the same structure have the same **Local Tag Name**, enter the **Topic Type Condition** below, to distinguish the topic types. If the reference uses fragments instead of a full tag, type a virtual tag name. The reference is then defined by the list of **Fragment Tag Names** below.
System Usage	Select one of the following: • **SYSTEM** The reference type will *not* appear in selection lists in dialog boxes, such as those for creating topics.

For this option	Do this
	Note: This is used for special purposes only, such as for adding new objects that may not be ready for general use or for objects to be used only by system processes, such as workflows. • **USER** The reference type will appear in selection lists in the appropriate dialog boxes, such as those for creating topics.
Validate Incoming on Parse	Select either **True** or **False**, to specify whether data being saved to the database is validated when content is decomposed. If **True** is selected, invalid content produces an error. Typically, this type of validation is not required if your editing tool validates content, and it may affect processing time.
Validate Outgoing on Parse	Select either **True** or **False**, to specify whether data being retrieved from the database is validated when content is composed. If **True** is selected, invalid content produces an error. Typically, this type of validation is not required if your editing tool validates content, and it may affect processing time.
Validate Example Content on Parse	If the topic type is related to a schema with the EXAMPLE_CONTENT schema type, select either **True** or **False**, to specify whether the example content is validated when it is used during compose. If **True** is selected, invalid content produces an error. Typically, this type of validation is not required if your editing tool validates content, and it may affect processing time.
Reference Type	Select the reference type you are creating: • **Composable Topic Reference** A topic-to-topic reference type. The reference is created by relating the topics in the Teamcenter client and is added to the content by the compose process. • **Composable Graphic Reference**

For this option	Do this
	A topic-to-graphic reference type. The reference is created by relating the topic and graphic in the Teamcenter client and is added to the content by the compose process. • **Content Topic Reference** A topic-to-topic reference type. The reference is created by pasting a link generated in the Teamcenter client, into the content in the editing tool. This reference type used to **enable fixed and floating navigation links**. • **Content Graphic Reference** A topic-to-graphic reference type. The reference is created by pasting a link generated in the Teamcenter client, into the content in the editing tool. Note: This reference type is required for the functioning of graphic options and graphic items within content. You must have a **Content Graphic Reference** reference type related to each topic type. A graphic reference type is included in the Base **administration data**; however you may need to modify it to work with your organization's schemas. • **Content Cross Reference** A reference type with the same functionality as **Content Topic Reference**. You can use this if you need to differentiate between two different types of content references.
Namespace URI	Type a URI that serves as a unique identifier for the namespace. Namespaces allow different XML vocabularies to be used in a single XML document and provide an XML parser a means to categorize each vocabulary. Although this is a URI, it is not used to identify and retrieve a web address.

For this option	Do this
Default Namespace Prefix	Type a prefix that represents the namespace URI. The prefix is added to the beginning of element names to place them in the namespace.
Topic Type Condition	If multiple topic types within the same structure have the same **Local Tag Name**, type a **topic type condition value** to restrict the topic type, using either **containspath**, **insidepath**, or **containsvalue**, along with a specified value or path. For example, insidepath(/REFDM) specifies that the reference is used inside the topic type REFDM. If the **Local Tag Name** of all topic types in a structure is unique, this attribute is not used.
Fragment Tag Names	If the topic type reference you are creating uses fragments instead of a full SGML tag with start and end tags, such as in a S1000D delivery list, type each tag name that appears in the fragment, in order of appearance. If you are using this option, enter a virtual tag name for the **Local Tag Name** above, which can be any unique string. Leave this option empty for standard topic types.

4. Click **Finish**.

5. Create an **XML attribute mapping** to define what data is read from the reference, and relate it to the reference type. The attribute mapping must have at least one entry for a key attribute, with the REFERENCE function.

Enabling navigation links

Overview of navigation links

Content authors use navigation links to add XML references with text to topics, which when published, create links. When clicked in published output, the user is directed to another topic or another location within the same topic. Before content authors can use navigation links, you must enable the ability to create the links, by creating a reference type and XML attribute mapping and relating those to the topic types in your system. You can create one reference type with the XML attribute mapping containing all the required attributes for the different types of links used by authors, or you can create a reference type with an attribute mapping for each link type. These steps enable both fixed and floating navigation link functionality. When the content author creates a navigation link and chooses the type of link, the resulting markup for the link in the content includes the reference, corresponding to the link type, and the text that is seen in the output.

Authors can insert two types of navigation links into a topic:

- **Floating navigation link**

  This navigation link type always links to the most current version of a topic.

  > Example:
  >
  > At the time you create a link to a topic, the topic's version is A.1, and the link accesses version A.1 of the topic. Later, when the topic's version increases to B.1, the link then accesses version B.1 of the topic.

- **Fixed navigation link**

  This navigation link type always links to a specific version of a topic.

  > Note:
  >
  > Fixed navigation links are not supported for S1000D content.

  > Example:
  >
  > At the time you create a link to a topic, the topic's version is A.1, and the link accesses version A.1 of the topic. Later, when the topic's version increases to B.1, the link still accesses version A.1 of the topic.

For both types of links, authors can also create a link to content below the topic level, for example, to a table or figure within a topic.

This example shows a floating link from a paragraph in one Base topic to a paragraph in another Base topic.

Topic 1

```
<component id=0001>
<title> Topic 1 Title</title>

<p>
      <xref href="0002.xml/p0002"
      linktyp="floating"
      version="CURRENT">
      </xref>
</p>

</component>
```

Topic 2

```
<component id=0002>
<title> Topic 2 Title</title>

<p id=p0002>
      This paragraph is in Topic 2.
</p>

</component>
```

Enable fixed and floating links and conrefs

1. Create a reference topic type, entering the following:

For this option	Do this
Topic Type Name	Type a name to assign to the reference type, for example, Link Component to Component.
Local Tag Name	Type the root tag name to be used for the reference, for example, xref. The tag name must be valid for the topic type(s) in which you insert the reference. When enabling a DITA conref, type the top level element and **+**. Example: topic + map + concept +
Reference Type	Choose **Content Topic Reference** Note: You may choose **Content Cross Reference** if you are configuring your system to have another distinguishable type of reference.

2. Relate the reference type to each topic type in your system, ensuring that you have a reference type between each of the topic types that can be linked.

3. Create an XML attribute mapping, with the required attributes, such as in the following examples.

   Cross-references (**xref**) in DITA Content Management:

Attribute Name	Function	Path	Field Separator	Constant Value
item_id	Reference	/@href	#*/	
/cdi0sourceAnchorFragId	Get	/@id		
linkType	Floating link	/@base		floating
version	Floating link	/@rev		CURRENT
linkType	Fixed link	/@base		fixed
item_revision_id	Fixed link	/@rev		
item_id	Get	/@href	#*/	
linkType	Get	/@base		
/ctm0destAnchorFragId	Get	/@href	/*	
item_revision_id	Get	/@rev		
DocumentTitle	Compose	/		

Content references (**conref**) in DITA Content Management:

Attribute Name	Function	Path	Field Separator	Constant Value
/cdi0sourceAnchorFragId	Reference	/@id		
/ctm0destAnchorFragId	Reference	/@conref	/*	
item_id	Reference	/@conref	#*/	
linkType	Floating link	/@base		floating
/ctm0versionNoOnRelation	Floating link	/@rev		CURRENT

Base Content Management:

Attribute Name	Function	Path	Field Separator	Constant Value
DocumentTitle	Compose	/		
item_id	Reference	/@href	*/	
linkType	Fixed link	/@linktyp		fixed
item_revision_id	Fixed link	/@version		
linkType	Floating link	/@linktyp		floating
version	Floating link	/@version		CURRENT
linkType	Get	/@linktyp		
item_revision_id	Get	/@version		
/ctm0destAnchorFragId	Reference	/@href	/*	

The Base example XML attribute mapping contains the following entries:

Attribute Name and Function	Purpose of the entry
DocumentTitle/Compose	The content of the **DocumentTitle** attribute from the topic you are linking

Attribute Name and Function	Purpose of the entry
	to appears in the link in the topic you are linking from, when the link is created.
item_id/Compose	The **item_id** attribute from the topic you are linking to appears in the link's **href** attribute in the topic you are linking from, when the link is created.
item_id/Reference	The **item_id** attribute from the topic you are linking to appears in the link's **href** attribute in the topic you are linking from, when the reference is created. The item ID appears before the field separator.
item_id/Get	Retrieves the **item_id** attribute from the **href** attribute in the content. The item ID appears before the field separator.
linkType/Fixed link	The attribute **linktyp** in the topic you are linking from contains the constant **fixed**, when a fixed link is used.
item_revision_id/Fixed link	The attribute **version** in the topic you are linking from contains the revision ID and sequence ID of the topic you are linking to, when a fixed link is used.
linkType/Floating link	The attribute **linktyp** in the topic you are linking from contains the constant **floating**, when a floating link is used.
version/Floating link	The attribute **version** in the topic you are linking from contains the constant **CURRENT**, when a floating link is used.
linkType/Get	Retrieves the **linkType** attribute from the attribute **linktyp** in the content.
item_revision_id/Get	Retrieves the **item_revision_id** attribute from the **version** attribute in the content.
/ctm0destAnchorFragId/Get	Retrieves the attribute **ctm0destAnchorFragId** from the **href** attribute after the field separator in the content.

4. Relate the XML attribute mapping to the reference topic type.

5. **Enable link validation**.

Enable link validation

When a link is inserted into a topic, it references an element from another topic. You can enable the validation of links so that if a referenced element is removed from a topic, an error message appears when the author tries to save the topic.

To enable this feature:

- Add the **ctm0destAnchorFragId** attribute to the XML attribute mapping that is related to the topic types in your system, as shown in the **DC Base Attributes** XML attribute mapping.

  > Note:
  >
  > This attribute is already included in the **DC Base Attributes** XML attribute mapping as well as the DITA XML attribute mappings. If those XML attribute mappings are already related to the topic types in your system, you do not have to add the attribute.

18. Creating XML attribute mappings

Overview of XML attribute mappings

Information is exchanged between a database property of an object and an attribute in the object's XML or SGML. This exchange of information takes place during many processes, such as compose, decompose, and initial creation. An attribute mapping defines the values that are exchanged, the user actions that cause the exchange, and the direction of the exchange (from database to attribute value or from attribute value to database). For example, an attribute mapping may transfer a title attribute from a topic's XML to the title property in the database whenever the topic is decomposed.

Each topic type must be related to at least one XML attribute mapping. You may have multiple XML attribute mappings for different topic types, or you may have the same XML attribute mapping related to every topic type, such as the **DC Base Attributes** XML attribute mapping provided with the **administration data**.

As a best practice, map IDs and XML numbers to the **item_id** attribute when defining XML attribute mappings. This way topics can be referenced by a unique ID within the XML document. This enables version management, import, compose/decompose, and referencing to work between topics. The DC Base Attributes XML attribute mapping supports the use of XML numbers specifically as the ID.

Caution:

- To create attribute mappings, you must have an understanding of the schemas.
- Because Content Management uses attribute mappings to map values between database properties and XML attributes in topics, it is recommended that you do not localize any values for properties that are included in attribute mappings. You should use Content Management translation objects to manage the translation of content and all mapped attributes.

DC Base Attributes XML attribute mapping overview

The **DC Base Attributes** XML attribute mapping is provided with the Content Management Base **administration data**. It is related to each Base topic type included in the administration data. You can use the attributes exchanged in this attribute mapping as the minimum exchange setting, modifying it to meet the differences in schemas and to add additional attributes for exchange. XML attribute mappings for DITA topic types are also included in the administration data.

Attribute Name	Function	Path	Constant Value	Fie
DocumentTitle	Decompose	/title	--- NO TITLE ---	
DocumentTitle	First Compose	/title		
DocumentTitle	Translation Receive	/title	--- NO TRANSLATED TITLE ---	
contentVersion	Translation Submit	/@contentversion		
contentVersionReference	Translation Receive	/@contentversion		
contentVersionReference	Translated Compose From Translation	/@contentversion		
ctm0LanguageTagref	Translated Compose From Translation	/@languagekey		
ctm0MasterLanguageTagref	Decompose	/@languagekey		
ctm0MasterLanguageTagref	Non Translated Compose	/@languagekey	English UK	
ctm0MasterLanguageTagref	Translation Receive	/@languagekey		
ctm0VersionNumber	Bidirectional	/@version		
/ctm0destAnchorFragId	Reference	/@id		
graphicPath	Processing Data	/processing-data/graphic-path		
graphicPathTarget	Processing Data	/processing-data/graphic-path/@target		
item_id	Bidirectional	/@xmlnumber		
item_id	Translation Receive	/@xmlnumber		
item_id	Clone	/@xmlnumber		
protected	Special	/@protected		
release_statuses	Compose	/@lifecyclestate		

This XML attribute mapping contains the following entries:

Attribute Name and Function	Purpose of the entry
DocumentTitle/Decompose	During decompose, the content from the **title** element in the XML is transferred to the **DocumentTitle** property in the database. If the **title** is empty, the constant value **NO TITLE** is used.
DocumentTitle/First Compose	The first time a topic is composed, the content from the **DocumentTitle** property in the database is transferred to the **title** element in the XML.
DocumentTitle/Translation Receive	When a translation delivery is imported, the **title** element in the XML of the translation is transferred to the **DocumentTitle** property of the translation in the database. If the **title** is empty, the constant value **NO TRANSLATED TITLE** is used.
contentVersion/Translation Submit	When a translation delivery is created, the **contentversion** attribute from the topic's XML is transferred to the **contentVersion** of the translation.
contentVersionReference/Translation Receive	When a translation delivery is imported, the **contentversion** attribute in the XML of the translation is transferred to the

Attribute Name and Function	Purpose of the entry
	contentVersionReference property of the translation in the database.
contentVersionReference/Translated Compose From Translation	When a translation is composed, the **contentversion** attribute in the XML of the translation is transferred to the **contentVersionReference** property of the translation in the database.
ctm0LanguageTagref/Translated Compose From Translation	When a translation is composed, the **languagekey** attribute in the XML of the translation is transferred to the **ctm0LanguageTagref** property of the translation in the database, so the content is composed in the translated language.
ctm0MasterLanguageTagref/Decompose	During decompose, the language is updated. For example, if a user changes the **languagekey** attribute from **English US** to **English UK** in the XML editor, when the content is saved, the master language of the topic is changed.
ctm0MasterLanguageTagref/Non Translated Compose	When content is composed for publishing, exporting or editing, the language is added to the XML attribute **languagekey**.
ctm0MasterLanguageTagref/Translation Receive	*Not currently used*
ctm0VersionNumber/Bidirectional	The **ctm0VersionNumber** attribute is a runtime property that is a combination of the revision and sequence number. During compose and translated compose, the content of the **ctm0VersionNumber** database property from the topic is transferred to the **version** attribute in the topic. During decompose, the value is transferred from the XML to the database.
/ctm0destAnchorFragId/Reference	**ctm0destAnchorFragID** is the attribute name on the relation between one topic and another topic it references. This entry specifies that this attribute uses the ID attribute from the topic that is being referenced. This entry enables **link validation**.

Attribute Name and Function	Purpose of the entry
graphicPath/Processing Data	Copies the value defined for **graphicPath** in the **ctm0_processing_data** preference to the composed content during compose, and removes the same content during decompose. This happens only for the root element of the XML data that is composed.
graphicPathTarget/Processing Data	Copies the value defined for **graphicPathTarget** in the **ctm0_processing_data** preference to the composed content during compose, and removes the same content during decompose. This happens only for the root element of the XML data that is composed.
item_id/Bidirectional	During compose and translated compose, the content of the **item_id** database property from the topic is transferred to the **xmlnumber** attribute in the topic. During decompose, the value is transferred from the XML to the database.
item_id/Translation Receive	When a translation delivery is imported, the **xmlnumber** attribute in the XML of the translation is transferred to the **item_id** property of the translation in the database.
item_id/Clone	The content of the **item_id** attribute from the topic being referenced appears in the reference when content is cloned using the **Copy For Editor→Composed Content as Clone** option.
protected/Special	Attribute used by editing tools to mark content as protected, or read-only, so it cannot be edited.
release_statuses/Compose	Adds the first release status to the XML attribute **lifecyclestate**.

Create an XML attribute mapping

1. Select the folder in which you want to create the new XML attribute mapping, and then either click **New Administrative Item...** , or choose **File→New→New Administrative Item**.

2. In the **New Administrative Item** dialog box, expand **Complete List**, click **XML Attribute Mapping**, and then click **Next**.

3. In the dialog box, type a name for the XML attribute mapping in the **Name** box.

4. (Optional) In the **Admin Comment** box, type a comment to be saved with the XML attribute mapping.

5. Click **Finish**.

6. Open the **Summary** view and select the XML attribute mapping.

7. If namespaces are used, under **Namespaces Table**, click **Add**.

   If namespaces are not used, go to step 13.

   > Tip:
   >
   > Namespaces allow different XML vocabularies to be used in a single XML document and provide an XML parser a means to categorize each vocabulary.

8. In the dialog box, in the **Namespaces Prefix** box, type the prefix that represents the namespace URI.

   > Tip:
   >
   > Prefixes are added to the beginning of element names to place them in the namespace.

9. In the **Namespace URI** box, type the URI that uniquely identifies the namespace.

   > Tip:
   >
   > Although this is a URI, it is not used to identify and retrieve a web address.

10. Click **Finish**.

11. For each namespace you want to add to the mapping, repeat steps 8 - 10.

    > Tip:
    >
    > To remove a namespace, click it, and select **Delete**.

12. Close the **Namespaces Table Entry** dialog box.

13. In the **Summary** view, under **XML Attribute Map Table**, click **Add**.

14. In the dialog box, enter the following for an attribute you want to add to the mapping:

In this column	Do this
Attribute Name	Type the name of the attribute as it is defined in the database. You can use the following in the attribute name for additional functionality: • **.** (period) Gets the attribute name on a reference property; for example, **ctm0MasterLanguageTagref.isoLanguageCode**. • **/** (front slash) Gets the property on a relationship; for example, **/ ctm0KeyName**. • **Ctm0BOMLine!** Gets the property on an occurrence note; for example **Ctm0BOMLine!Ctm0versionNo** .
Constant Value	Type the default value to be used when the attribute value is blank.
Field Separator	Type the string you want to be used as a field separator. The field separator enables you to combine multiple values from the XML/SGML content into a single object attribute during attribute mapping. If you add multiple rows with the same **Function** and **Attribute Name** in your attribute mapping object, and enter the **Field Separator String** on all but one of the rows, the values from the XML/SGML content are appended together into a single value during decompose operations. Also, a single value can be split at the separators and included in the XML/SGML content as multiple values. It is recommended that you use a field separator that does not typically appear in the XML/SGML content values. If the attribute does not allow separators, use the **Fixed Field Length** below.
Fixed Field Length	For attributes that do not allow separators, type the number of characters to which you want to truncate values when they are transferred from the database to XML/SGML content, as an alternative to the **Field Separator String**.
Function	Select the function for the attribute, from the list of **XML attribute mapping functions**.

In this column	Do this
Omit empty attribute	Select either **True** or **False**, to specify whether empty attributes are omitted when content is transferred.
Path	Type the path to express the element and attribute in the XML/SGML tag. Tip: The element and attribute must be defined in the schema related to the topic type; otherwise it is not allowed in the content being composed. A path consists of any number of element tag names separated by slashes (/), and it may contain no more than one attribute name or processing instruction (PI) target, although one is not required. An attribute is separated from the element tag names with the at symbol (@), and a PI is separated with a question mark (?). Guidelines for the syntax are: • A path always starts with a slash. • A path may not end with a slash, at symbol, or question mark. • A path may not have both an at symbol and a question mark. • All slashes must appear before any at symbols or questions marks. • A path may use two slashes to mark that it should be ignored. **Examples:** `/note/para@version` = Attribute **version** of the element **para** inside the element **note** `/note/para?PItarget` = PI with target **PItarget** inside element **para** inside element **note** `/@version` = Attribute **version** `/?PItarget` = PI with target **PItarget** `/note` = Element **note** `/` = Current element

In this column	Do this
	// = Ignore this path
XML Procedure	For more complicated mappings, type the name of the **procedure** script used to copy the values for an attribute. If a procedure is used, all other columns except **Attribute Name** and **Function** are not used, and the procedure replaces the standard mapping for this row.

15. Click **Finish**.

16. For each attribute you want to add to the mapping, repeat steps 14 - 15.

> Tip:
>
> To remove an attribute, click it, and select **Delete**.

17. Close the **XML Attribute Map Table Entry** dialog box.

18. (Optional) Add a **callback function** to an attribute to further configure or customize XML compose or decompose results.

    a. In the line for the attribute, click the **Callback Name** column, and choose **View Properties**.

    b. In the **Properties** dialog box, enter the callback name in the **Callback Name** box, and click **OK**.

Callback functions in XML attribute mappings

Application administrators can add callback functions to attributes in XML attribute mappings to further configure or customize XML compose or decompose results. A callback adds an interrupt in the attribute mapping process that allows authors to insert processing to modify the XML using C++ code.

1. Create the callback function, and add it to a library.

> Example:
>
> ```
> int example_compose_xam_callback_fn( tag_t itemRev,
> tag_t language,
> tag_t xamAttrMapRow,
> xercesc::DOMDocument & pDoc,
> NameValueMap &keyValueArgs)
> ```

```
int example_decompose_xam_callback_fn( tag_t itemRev,
tag_t topicType,
tag_t xamTblRow,
xercesc::DOMNode* target,
NameValueMap & keyValueArgs,
std::string& returnValue )
```

2. Install the callback code in your environment using the **install_callback** utility. For example:

Example:

```
install_callback -u=dba -p=dba -g=dba -mode=create
-type=ContmgmtComposeXMLAttrMapping   -library=libCtm0Callbacks
-function=example_compose_xam_callback_fn
-name=example_compose_xmlattrmap_callback_fn


install_callback -u=dba -p=dba -g=dba -mode=create
-type=ContmgmtDecomposeXMLAttrMapping   -library=libCtm0Callbacks
-function=example_decompose_xam_callback_fn
-name=example_decompose_xmlattrmap_callback_fn
```

3. When you create an **XML attribute mapping**, add the **Callback Name** to the appropriate attribute.

Example:

In the first example above, the **Callback Name** is **example_compose_xam_callback_fn**.

XML attribute mapping functions

The following are the functions that may be assigned to an attribute when you are **creating an XML attribute mapping**:

Function	Description
Bidirectional	Copies values from the topic object to the composed content during compose and translated compose, and copies values from the composed content to the topic object during decompose.
Clone	Copies values from source topics to target topics when content is cloned using the **Copy For Editor→Composed Content as Clone** option.
Compose	Copies values from the topic object to the composed content during compose and translated compose.

Function	Description
Decompose	Copies values from the composed content to the topic object during decompose.
Export File Name	Specifies a separate export file name for exports with **composable and content reference topic types**.
First Compose	Copies values from the topic object to the composed content during compose and translated compose, but only if the topic does not have any content yet. Typically, this applies to newly created topics.
Fixed link	Copies values from source topics to the XML reference when fixed links are used.
Floating link	Copies values from source topics to the XML reference when floating links are used.
Get	Retrieves values from the composed content during decompose.
Graphic Reference	Identifies graphic references during compose and decompose.
Non Translated Compose	Copies values from the topic object to the composed content during compose, but not during translated compose.
Processing Data	Copies the values stored in the **ctm0_processing_data** preference to the composed content during compose, and removes the same content during decompose. This happens only for the root element of the XML data that is composed.

Example:

If the XML attribute mapping has these entries:

Attribute Name	Function	Path
graphicPath	Processing Data	/graphic-path
graphicPathTarget	Processing Data	/graphic-path/@target

And the **ctm0_processing_data** preference contains these values:

graphicPath=[root-dir]graphics

graphicPathTarget=NONE

The content contains this in the composed content for the root element:

Function	Description
	<graphic-path target="None">C:\Users\username\Teamcenter\contmgmt\9000.0.0\graphics</graphic-path>
Reference	Copies values when a reference is created or decomposed.
Remove	When the XML attribute mapping uses a **Compose** or **Decompose** function for removing a value from the xmlNumber, versionNumber, or protected attributes, and the function uses a procedure triggered by another attribute, use the **Remove** function to remove the values from those attributes. XML attribute mapping functions are checked in this order for removing a value from the xmlNumber or versionNumber attributes during compose: 1. **Remove** 2. **Bidirectional** 3. **Compose** XML attribute mapping functions are checked in this order for removing a value from the xmlNumber, versionNumber, contentVersion, or protected attributes during decompose: 1. **Remove** 2. **Bidirectional** 3. **Decompose**
Set	*For future use.*
Special	Identifies special attributes used by Content Management such as **protected**.
Translate Attribute	During the translation receive process, retrieves the value of the translate attribute the XML document. For example, in the DITA standard, this attribute is **translate**. If the attribute's value is **no** or **false**, then the translated topic is skipped.

Function	Description
Translated Compose From Topic	Copies values from the topic object to the composed content during translated compose, but not during standard compose.
Translated Compose From Translation	Copies values from the translation object (not from the topic object) to the composed content during translated compose.
Translation Receive	Copies values when receiving translation deliveries from the translation office.
Translation Submit	Copies values when building translation submittals for the translation office.

19. Creating style types and stylesheets

Overview of style types and stylesheets

You use stylesheets to customize output. A stylesheet is represented by a source file used for transforming XML data. Stylesheets are applied for transformations during import or export, for data presentations in viewing and editing tools, and for the creation of renditions, such as PDF or HTML files.

If you publish product documentation that uses multiple stylesheets, you can combine and manage those stylesheets within a style type. The style type establishes defaults for stylesheet options on preview and publish functions.

> Note:
>
> You must create a stylesheet to enable the viewing of content in the **Preview** view using the **Preview** option.

Stylesheet items have stylesheet revisions. A stylesheet revision must be related to a topic type, and it may be related to a style type and an editing or publishing tool before it can be used.

The DITA administration data includes several DITA stylesheets. To locate them, you can use the system defined searches for Content Management.

Create a style type

1. Select the folder in which you want to create the new style type, and then either click **New Administrative Item...**  , or choose **File→New→New Administrative Item**.

2. In the **New Administrative Item** dialog box, expand **Complete List**, click **Style Type**, and then click **Next**.

3. In the dialog box, do the following:

For this option	Do this
Name	Type the name for the style type.
System Usage	Select one of the following: • **USER** The style type will appear in selection lists in the appropriate dialog boxes.

For this option	Do this
	• **SYSTEM** The style type will *not* appear in selection lists in dialog boxes. Note: This is used for special purposes only, such as for adding new objects that may not be ready for general use or for objects to be used only by system processes, such as workflows.

4. Click **Finish**.

Using style types

Style types are used to manage multiple stylesheets. Therefore, to associate a style type and stylesheet, relate a style type to stylesheets that are used for publishing.

When you publish, depending on the stylesheets you want to execute, you can select the appropriate style type from the **Publish** dialog box.

Example:

You want to publish to PDF, and your admin data is as follows:

```
Style Type for Company A
|_ Stylesheets for Company A
     |_ PDF Publish Tool


Style Type for Company B
|_ Stylesheets for Company B
      |_ PDF Publish Tool
```

In this case, if you want to use the stylesheets for company B, then in the **Publish** dialog box, first select PDF as the **Tool** and then select Style Type for Company B as the **Style Type**.

Create a stylesheet

1. (Optional) Either select a folder in which you want to create the stylesheet, or select the style type to which you want to relate the stylesheet.

2. Either click **New Administrative Item...**  , or choose **File→New→New Administrative Item**.

3. In the **New Administrative Item** dialog box, expand **Complete List**, click **Stylesheet**, and then click **Next**.

4. In the dialog box, do the following:

For this option	Do this
ID	Either type an ID for the stylesheet, or leave the box blank so that the next available item ID is automatically assigned
Revision	Either type a revision for the stylesheet, or leave the box blank so that the next available revision is automatically assigned.
Name	Type the name for the stylesheet.
Public ID	Type a public ID for the stylesheet. Note: The public ID cannot be modified after the stylesheet has been created because it is unique.
Stylesheet Type	Select the stylesheet type from the **list of stylesheet types**.
Stylesheet Main File (for Zipped)	If the stylesheet is contained in a combined file format, such as a .zip file, type the name of the main stylesheet file that includes or refers to all other files.
Stylesheet Resulting Content Type	Select the resulting content type of the stylesheet, which specifies the kind of content produced by the stylesheet. Tip: When creating a stylesheet to enable the viewing of content in the **Preview** view using the **Preview** option, select **xml**.
Ant Build Target	Type the reference to the ANT control process target for formatting.
Select Content	Browse to and select the content file of the stylesheet. Tip: In the **Browse** dialog box, the file type you browse for depends on the tools and types of stylesheets your company uses. For example, the .xac file type is used

For this option	Do this
	with XMetaL Author. The .xsl and .xml files types are used with many types of tools.

5. Click **Finish**.

6. Create the following relationships between the stylesheet revision and other objects:

   - **Relate** the stylesheet revision to the style type if you did not choose the style type in step 1.
   - **Relate** the stylesheet revision to **topic types** as necessary.
   - **Relate** the **editing tool** or **publishing tool** that applies to the stylesheet revision.

Edit a stylesheet

You can edit a stylesheet for Base or DITA Content Management using an editing tool, similarly to how you edit topics. To **edit S1000D stylesheets**, you must edit the stylesheet files on Teamcenter Dispatcher.

1. In the **Home** component view, right-click the stylesheet and choose **Edit→Edit Stylesheet**.

2. In the **Check-Out** dialog box, click **Yes**.

3. The stylesheet opens with the editing tool and is saved in a local folder. In the **Home** component view, the stylesheet appears as checked out and in edit mode .

4. In the editing tool, edit the stylesheet as necessary.

5. When you are done editing the stylesheet, do one of the following:

To do this	Do the following	For this result
Save your changes and close the file.	a. Use the editing tool's options for saving files, and then in the **Home** component view, right-click the stylesheet, and choose **Edit→Save and Close Edit**. b. Right-click the stylesheet, and choose	The editing tool closes the file, and the stylesheet is updated in the database.

To do this	Do the following	For this result
	Check-In/Out→Check In.	
Close the file without saving your changes.	a. In the **Home** component view, right-click the stylesheet, and choose **Edit→Cancel Edit**.	The editing tool closes the file, the stylesheet remains unchanged in the database.
	b. Right-click the stylesheet, and choose **Check-In/Out→Cancel Checkout**.	

Stylesheet types

The following are the available stylesheet types:

Stylesheet Type	Description
ARCHIVE_ TRANSFORMATION	A stylesheet used in transformation policies. If the stylesheet is applied to a set of related objects, it is applied only to the main object.
CONTENT_ TRANSFORMATION	A stylesheet used in transformation policies. If the stylesheet is applied to a set of related objects, it is applied to all objects in the set.
COMPOSED_ERROR_LOG	A stylesheet for compose error information. This displays the error log created for a composed document.
COMPOSED_VERSION_ LOG	A stylesheet for compose version information. This displays the compose version log created for a composed document.
EDITOR_VIEW	A stylesheet used by an editing tool to display composed topics for an editing session.
FOSI_VIEW	A stylesheet used with the FOSI (Formatting Output Specification Instance) language.
IMPORT	A stylesheet that transforms XML data by Extensible Style Sheet Transformations (XSLT) during an import into Content Management.
INTERNET_EXPLORER _PREVIEW	A stylesheet that transforms data to HTML while displaying it in the **Preview** view.

Stylesheet Type	Description
PREDECOMPOSE	A stylesheet that transforms data by XSLT before starting the decompose of edited topics, which is done when you save and close the topic.
PUBLISH_FOR_PRINT	A stylesheet that transforms data to a format suitable for printing on paper.
PUBLISH_TO_HTML	A stylesheet that transforms data to HTML.
PUBLISH_TO_PDF	A stylesheet that transforms data to PDF. This type of stylesheet is used by any publishing tool that supports the creation of PDF documents.
TOPIC_DEPENDENT	The default stylesheet type for publishing and previewing. The stylesheet that is applied when a topic is published is dependent on the topic type, tool, and style type.
TRANSLATION_ORDER	*This stylesheet type is no longer used.*

20. Enabling graphics features

Required tasks to enable authors to use graphics

To enable authors to use graphics in publications, you must do the following:

- **Create reference topic types** to enable the functioning of graphics within content
- **Define default graphic usages**
- **Create graphic priority lists**
- **Create graphic attribute mappings**
- **Maintain graphic tagging options**

Enabling references between topics and graphics

Authors create references between topics and graphics by pasting a link generated in the Teamcenter client into the content in the editing tool. These links are generated in the Teamcenter client by using the **Copy For Editor** option on graphic option revisions.

For these references to function, you must create a reference topic type defined as a **Content Graphic Reference** reference type, and this reference topic type must be related to each topic type used by authors. A graphic reference type is included in the Base **administration data**; however you may need to modify it to work with your organization's schemas depending on the element, attribute, or reference type that you use to create those links.

Create a graphic reference topic type

1. Do one of the following:
    - Modify the reference topic type **graphic-reference**, included in the **administration data**, to meet the needs of your organization.
    - **Create a reference topic type**, entering the following:

For this option	Do this
Name	Type a name to assign to the reference type.
Local Tag Name	Type the root tag name to be used for the reference. The tag name must be valid for the topic type(s) in which you insert the reference.
Reference Type	Choose **Content Graphic Reference**.

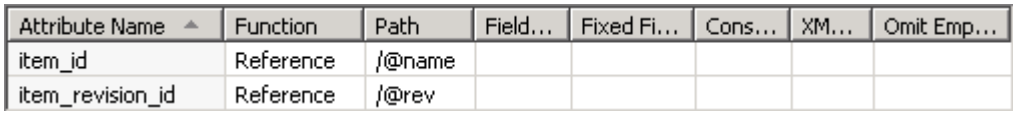
2. Relate the reference topic type to each topic type in your system, ensuring that you have a reference type between each pair of related topic types.

3. Do one of the following:

   - Modify the XML attribute mapping **graphic-data** included in the administration data.
   - Create an XML attribute mapping, with the required attributes.

   Example:

   The **graphic-data** XML attribute mapping contains entries that copy the **item_id** and **item_revision_id** attributes from the graphic to the **name** and **revision** attributes in the topic when the reference is created.



Attribute Name	Function	Path	Field...	Fixed Fi...	Cons...	XM...	Omit Emp...
item_id	Reference	/@name					
item_revision_id	Reference	/@rev					

4. If you did not use the objects in the provided administration data, relate the XML attribute mapping to the reference topic type.

Defining default graphic usages

Graphic usages are assigned to a graphic to specify the types of deliverables you want to use the graphic for. System-defined graphic usages include: **ICON**, **PDF**, **PRINT**, **SOURCE**, **THUMBNAIL**, **VIEW**, and **WEB**.

When graphics are imported, they may be assigned unique graphic usages, or they may be assigned the usages defined in the **ctm0GraphicUsagePref** preference, which assigns usages based on graphics' file extensions. To make defined usages available to users, administrators must edit the **ctm0GraphicUsagePref** preference to establish them. When values are set for this preference they apply to the site.

Your publishing and editing tools, as well as the output format and display, must support the graphic file format you assign. For example, users may need JT2Go or a similar application to view a JT graphic if included in the published output.

Working with graphic priority lists

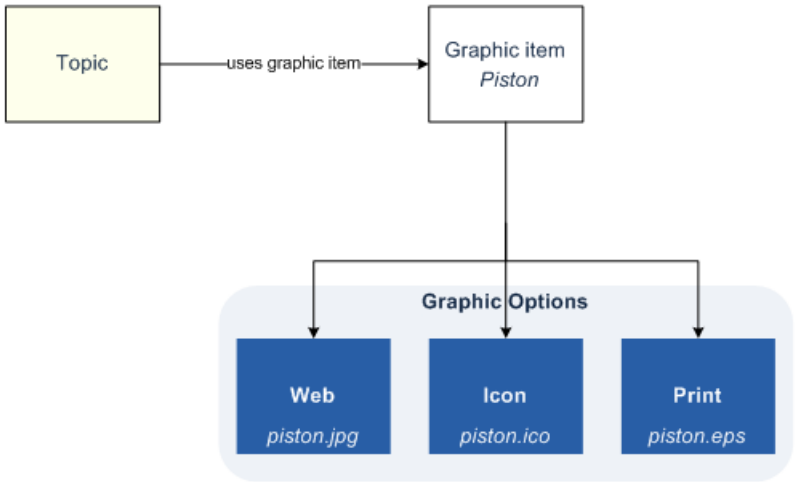
A *graphic* item is the main object representing a set of different file types for the same graphic. A *graphic option* is a specific file of a specific file type. For example, a graphic item may be named *piston*, and it may have several associated graphic options named piston.jpg, piston.gif, and piston.bmp.

When a graphic option is imported to the database, one or more usages are assigned to it, such as VIEW for use in previewing and web publishing, or PDF for use in high-resolution printing. Each graphic item may be assigned unique usages, or they may be assigned the usages specified in the **ctm0GraphicUsagePref** preference.

You must create graphic priority lists in the Organization application and associate one with each **editing**, **publishing**, and **comparing** tool when you create the tool. A graphic priority list determines the priority with which graphic options are selected for use with XML content when it is opened or published with the tool. Graphic options are selected according to the order of usages in the graphic priority list. The tool selects the graphic option for the first usage listed; however, if no file with this usage is found, the graphic option for the usage in the next position is selected, and so on.

> Example:
>
> An XML document with graphics is opened in an editing tool. The editing tool's graphic priority list has usages in this order: WEB, ICON, and PRINT. All graphics in the XML document are first searched for graphic options with the WEB usage. If a graphic has a WEB graphic option, that graphic option is downloaded with the content. If a graphic does not have a WEB graphic option, a search for an ICON graphic option is done. If found, the ICON option is used. If not found, the graphic is searched for the PRINT option, and so on.

This example is illustrated here:



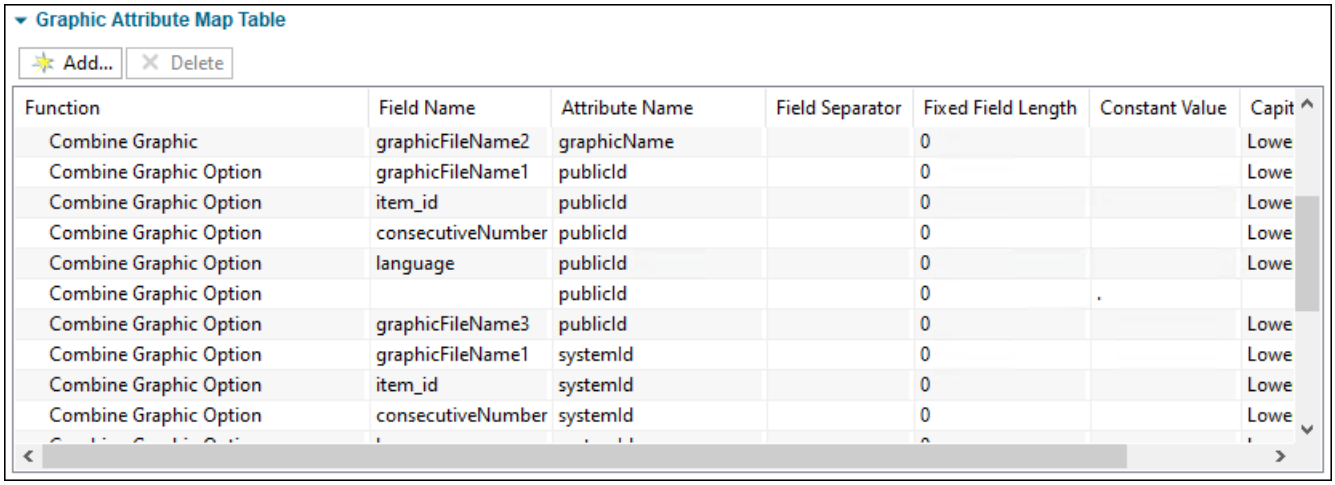
Note:

Reference types are required for the functioning of graphic options and graphic items within content. You must have graphic reference related to each topic type.

Working with graphic attribute mappings

A graphic attribute mapping controls how a file name or attributes on a graphic file being imported are mapped to properties in Content Management. For example, a graphic attribute mapping can define how a graphic option's file name is transferred to its Public Identifier property in Content Management. A graphic attribute mapping can map file names or attributes to both graphic options and graphic items.

You must have at least one graphic attribute mapping defined in your system. You can import sample graphic attribute mappings included with Base and S1000D Content Management **administration data**. A Base sample graphic attribute mapping follows:



Function	Field Name	Attribute Name	Field Separator	Fixed Field Length	Constant Value	Capit
Combine Graphic	graphicFileName2	graphicName		0		Lowe
Combine Graphic Option	graphicFileName1	publicId		0		Lowe
Combine Graphic Option	item_id	publicId		0		Lowe
Combine Graphic Option	consecutiveNumber	publicId		0		Lowe
Combine Graphic Option	language	publicId		0		Lowe
Combine Graphic Option		publicId		0	.	
Combine Graphic Option	graphicFileName3	publicId		0		Lowe
Combine Graphic Option	graphicFileName1	systemId		0		Lowe
Combine Graphic Option	item_id	systemId		0		Lowe
Combine Graphic Option	consecutiveNumber	systemId		0		Lowe

Warning:

If you change the default graphic attribute mapping, Teamcenter might function incorrectly and you could get an error message. Therefore, it is recommended that you change it only after understanding its consequences.

Example

You modify the default graphic attribute mapping for the *publicId* attribute by removing the *language* field. Since there is no language field in publicId, the publicId attribute for different graphic option revisions would be the same if the same graphic is imported with different languages. This would cause the import graphic operation to fail, as publicId must be unique, for example. Therefore, if you want to make a change, you must make sure to keep the publicId attribute unique for all graphic option revisions.

Create a graphic attribute mapping

1. Select the folder in which you want to create the new graphic attribute mapping, and then either click **New Administrative Item...** , or choose **File→New→New Administrative Item**.

2. In the **New Administrative Item** dialog box, expand **Complete List**, click **Graphic Attribute Mapping**, and then click **Next**.

3. In the dialog box, in the **Name** box, type the name for the mapping.

4. In the **Admin Comment** box, type a comment that you would like to store with the graphic attribute mapping.

5. Click **Finish**.

6. Close the **New Administrative Item** dialog box.

7. Open the **Summary** view and select the graphic attribute mapping.

8. In the **Summary** view, click **Add**.

9. In the **Graphic Attribute Mapping Table** dialog box, from the **Function** list, select the following for the file name, portion of a file name, or attribute to map *from* the files you are importing:

   > Tip:
   >
   > You may map multiple fields to be combined in one graphic item or option attribute, by adding functions for each field in the order in which you want them to be mapped.

To map this	Select this
The suffix of the importing graphic file name. For example, **.eps** is the suffix of **piston.eps**.	**ASSUME_SUFFIX_FROM_IMPORTGRAPHICFILENAME**
Map a part of the importing graphic file name, not including the suffix Note: If file names contain more than one separator,	**SPLIT_IMPORTGRAPHICFILENAME_WITHOUT_SUFFIX**

To map this	Select this
for example: wheel.right.jpg, use this function.	
Map a portion of the importing graphic file name, including the suffix	**SPLIT_IMPORTGRAPHICFILENAME**
Map an attribute from an importing graphic item	**SPLIT_ATTRIBUTE_FROM_GRAPHICOBJECT**
Map an attribute from a language object used for the import	**SPLIT_ATTRIBUTE_FROM_LANGUAGEOBJECT**
Map the suffix of the importing graphic option's original file name. Use this to define the capitalization on the graphic option's resulting suffix, using the **Capitalization Behavior** option in another function, **COMBINE_GRAPHICOPTION**.	**SET_SUFFIX_IMPORTORIGINALFILENAME_GRAPHICOPTION**
Map the consecutive numbering of the importing graphic option	**SET_CONSECUTIVEID_GRAPHICOPTION**

10. Complete the remaining options according to the function you selected:

If this function is selected	Do this
ASSUME_SUFFIX_FROM_IMPORTGRAPHICFILENAME	In the **Field Name** box, type any name to assign the suffix to, for example, **graphicFilenameSuffix**.
SPLIT_IMPORTGRAPHICFILENAME_WITHOUT_SUFFIX	a. In the **Field Name** box, type any name to assign to this part of the file name. b. Do one of the following:

If this function is selected	Do this
	Note: When combining multiple fields to map to an attribute in Content Management, for the last field, neither of these options are required—all remaining characters of the file name are mapped. • In the **Field Separator** box, enter the type of separator used in the graphic file name at which point you want to split the file name into multiple parts and assign this part to the **Field Name** you specified. • In the **Fixed Field Length** box, type the number of characters in the file name that you want to assign to this part as the **Field Name** you specified. You must assign a **Fixed Field Length** for the part of the file name that is between two separators; however, you can use an asterisk (*) to assign a variable length to *one* field when combining multiple fields. Example: You add one function with a **Fixed Field Length** of **8** and another function with a period (.) for the **Field Separator**. When you import **wheel.right.jpg**, the first function maps **wheel.ri** and the second function maps **ight**. You use another function, either **COMBINE_GRAPHIC** or **COMBINE_GRAPHICOPTION**, to combine these fields to map to an attribute in Content Management.
SPLIT_IMPORTGRAPHICFILENAME	a. In the **Field Name** box, type any name to assign to the portion of the file name. b. Do one of the following: • In the **Field Separator** box, enter the type of separator used in the graphic file name at which point you want to split the file name into multiple parts and assign this part to the **Field Name** you specified.

If this function is selected	Do this
	• In the **Fixed Field Length** box, type the number of characters in the file name that you want to assign to this part as the **Field Name** you specified.
SPLIT_ATTRIBUTE_FROM_GRAPHICOBJECT	a. In the **Field Name** box, type any name to assign to the attribute you are mapping. b. In the **Attribute Name** box, type the name of the attribute from the graphic files you are importing.
SPLIT_ATTRIBUTE_FROM_LANGUAGEOBJECT	a. In the **Field Name** box, type any name to assign to the attribute you are mapping. b. In the **Attribute Name** box, type the name of the attribute from the graphic translation object you are importing.
SET_SUFFIX_IMPORTORIGINALFILENAME_GRAPHICOPTION	In the **Field Name** box, type any name to assign to the attribute you are mapping.
SET_CONSECUTIVEID_GRAPHICOPTION	In the **Field Name** box, type any name to assign to the numbering you are mapping. Note: Use the **COMBINE_GRAPHICOPTION** function below to specify the attribute in Content Management you are mapping the numbering to.

11. Click **Finish**.

12. Do the following to map the file name, portion of a file name, or attribute *to* a graphic item or graphic option in Content Management:

    a. From the **Function** list, choose one of the following:

    - **COMBINE_GRAPHICOPTION**, to map an attribute to a graphic option
    - **COMBINE_GRAPHIC**, to map an attribute to a graphic item

    Note:

    When mapping attributes to a graphic item, you cannot use the results from these functions:

- **SPLIT_ATTRIBUTE_FROM_LANGUAGEOBJECT**
- **SET_CONSECUTIVEID_GRAPHICOPTION**
- **SET_SUFFIX_IMPORTORIGINALFILENAME_GRAPHICOPTION**

b. Do one of the following:

- In the **Field Name**, type the field name you defined in step 10.
- Type a value in the **Constant Value** box, which is a value that is always mapped to the attribute you specify.

c. In the **Attribute Name** box, type the name of the attribute in the graphic item or option that you want to fill in with the mapped value.

d. (Optional) In the **Field Separator** box, type a string, for example, a hyphen (-), to use as a separator between fields in the attribute. The field separator is added after the value of the **Field Name** or **Constant Value**.

e. (Optional) In the **Capitalization Behavior** box, choose either **LOWERCASE**, **UPPERCASE**, or **MIXEDCASE** to indicate how the mapped value appears in the attribute. **MIXEDCASE** is the default, which leaves the capitalization unchanged from the original file.

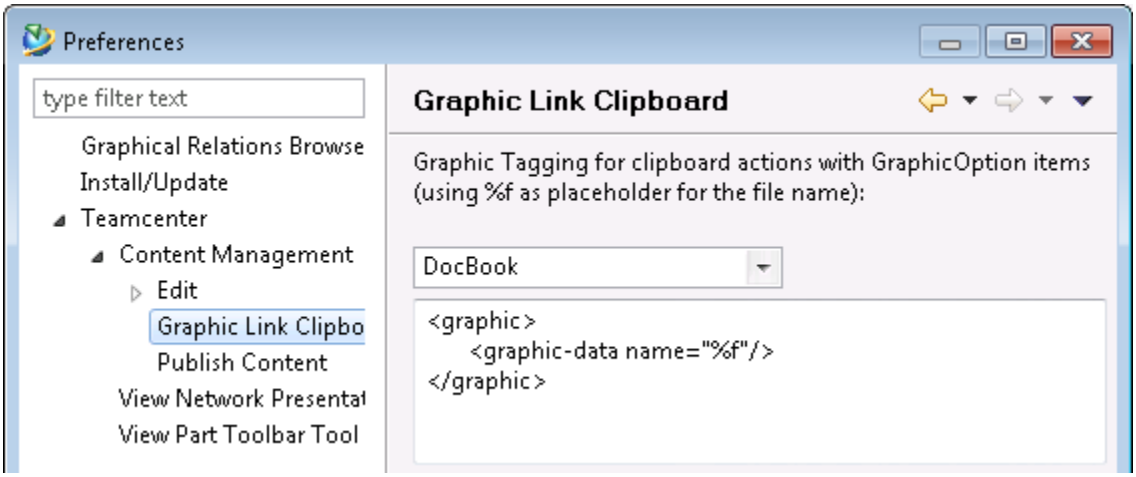
f. Click **Finish**.

13. Repeat steps 9–12 for each function to add to the graphic attribute mapping.

14. When you are done adding functions to the graphic attribute mapping, close the **Graphic Attribute Mapping Table** dialog box.

Maintaining graphic tagging options

Authors use the **Graphic Link Clipboard** Window preference to set the tagging used in the editing tool when a graphic is copied and pasted from Content Management to the editing tool. Administrators use Teamcenter Content Management preferences to specify the options available in this preference and how each option functions.



Upon installation, seven preferences exist to define the options for the **Graphic Link Clipboard** Window preference.

The **ctm0GraphicPref.values** preference defines the values listed in the **Graphic Link Clipboard** preference list box. The values at installation are **docBook**, **dita**, and **s1000d**. The names and graphic tagging are defined in the following preferences:

- **ctm0GraphicPref.dita.name**

  Defines the name of the option to choose when working with the DITA standard. The default is **DITA**.

- **ctm0GraphicPref.dita.text**

  Defines the default XML tagging that you want to use in the editing tool when a graphic is copied from Content Management and pasted to content in the editing tool, when working with the DITA standard for publications.

- **ctm0GraphicPref.docBook.name**

  Defines the name of the option to choose when working with the DocBook or similar XML standard. The default is **DocBook**.

- **ctm0GraphicPref.docBook.text**

  Defines the default XML tagging that you want to use in the editing tool when a graphic is copied from Content Management and pasted to content in the editing tool, when working with the DocBook or similar XML standard.

- **ctm0GraphicPref.s1000d.name**

  Defines the name of the option to choose when working with the S1000D 4.0, 4.1, 4.2, or 5.0 standard. The default is **S1000D 4.x**.

- **ctm0GraphicPref.s1000d.text**

  Defines the default XML tagging that you want to use in the editing tool when a graphic is copied from Content Management and pasted to content in the editing tool, when working with the S1000D 4.0, 4.1, 4.2, or 5.0 standard.

You can add a new option to the list box by doing the following:

1. Add a new value to the **ctm0GraphicPref.values** preference.

2. Create a new preference to specify how the new option appears in the list box, for example, **ctm0GraphicPref.***newstandard***.name**.

3. Create a new preference to specify the graphic tagging to use for the new option, for example, **ctm0GraphicPref.***newstandard***.text**.

21. Enabling content editing

Setting up tools for editing content

An editing tool launches when you edit a topic and is used to author and edit XML and SGML content. Editing tools are installed on each client following the supplier's instructions. To set up an editing tool, complete the actions below.

> Note:
>
> Editing tools are supported only on Windows clients.

The following are examples of editing tools that work with Content Management.

- Oxygen XML Editor
- JustSystems XMetaL Author

Additional tools may be defined to work with Content Management.

Set up actions

To set up an editing tool, complete the actions below as applicable.

- Create an editing tool object for each editing tool installed on the client.
- **Install** any additional plug-ins required for the editing tool, unless you are using **Teamcenter XMetaL Client**. If you are using Teamcenter XMetaL Client, do not install the XMetaL plugin in the Teamcenter interface.
- Relate a **stylesheet revision** to an editing tool if you want a specific stylesheet to be used when topics are edited.
- Implement any desired customizations or macros provided by the editing tool.
- If you are using the Oxygen XML Editor integration, activate the **Open in Oxygen** command in Active Workspace. To activate, update the preference named **ctm0EnableOxygenIntegration** and set its value to **TRUE**.
- If you import Base Content Management **administration data**, you also import the **XMetal 9.0** editing tool object. Therefore, if you install XMetal 9.0 and you import this tool object, verify that the assigned tool path matches the path for your installation of the tool.

Create an editing tool object

1. Select the folder in which you want to create the new editing tool object, and then either click **New Administrative Item...**  , or choose **File→New→New Administrative Item**.

2. In the **New Administrative Item** dialog box, expand **Complete List**, click **Editing Tool**, and then click **Next**.

3. In the dialog box, do the following:

For this option	Do this
Name	Type a name for the editing tool. Tip: If you are using Teamcenter XMetaL Client (released with Teamcenter 10.1.0.1 and later), locate the XMetaL installation folder ...\SoftQuad\XMetaL to obtain the version number, and type the name as **XMetaL**, space, and *version number*. For example, `XMetaL 9.0`.
Tool Activation	Select the tool activation that corresponds to the editing tool object you are creating: Tip: If no tool activation is available for the editing tool object you are creating, leave the value blank or select **SIMPLE_TEXT_EDITOR**. The tool is launched externally using the **Tool Command**.

If the editing tool is	Select this
Altova XML-Spy	**EDITOR_XMLSPY**
FrameMaker	**EDITOR_FRAMEMAKER**
Altova Authentic	**EDITOR_AUTHENTIC**
XMetaL Author	**EDITOR_XMETAL**
A text editor, such as Microsoft Notepad, Microsoft	**SIMPLE_TEXT_EDITOR**

For this option	Do this
	WordPad, or Textpad Note: The **EDITOR_EMBEDDED_XMETAL** tool is for future use.

Tool Command	Type the command used by Content Management to start the tool. This includes the start command (for example, an **.exe** file) and any required or optional parameters for the tool you are using. The valid command line parameters for the tool are typically documented by the tool supplier.
	
	Here are some example tool commands:
	
	`XMLSpy.exe "%s"`
	
	`xmetal90.exe "%s"`
	
	`FrameMaker.exe "%s"`
	
	`TextPad.exe "%s"`
	
	`wordpad.exe "%s"`
	
	Tip:
	The valid parameters depend on the tool you use.

Tool Path	Type the path where the tool's start file, such as an **.exe** file, is stored. For example: C:\Program Files\XMetaL 9.0\Author
Graphic Priority Table Reference	Select the **graphic priority list** to associate with the tool. The graphic priority list is used to select the **graphic options** to be downloaded to the content when it is opened in the editing tool.
Download graphics	Select either **True** or **False**, to specify whether graphics that are referenced by the content are downloaded from Teamcenter when the editing tool is launched. This allows for the most recent graphics to be distributed to users from a central location.
Download schema	Select either **True** or **False**, to specify whether schemas on which content is based are downloaded from Teamcenter when the editing tool is launched. This allows for the most

For this option	Do this
	recent schemas to be distributed to users from a central location.
Download stylesheet	Select either **True** or **False**, to specify whether stylesheets are downloaded from Teamcenter when the editing tool is launched. This allows for the most recent stylesheets to be distributed to users from a central location.
Processing instruction	Type any necessary processing instruction for the editing tool. The processing instruction is written into the content files when opened in the editing tool. XMLSpy requires a processing instruction inside the content to select the stylesheet to use during editing. The processing instruction for this tool is: `<?altova_sps%s?>` where *%s* is the stylesheet main file name.
System Usage	Select one of the following: • **SYSTEM** The editing tool will *not* appear in selection lists in dialog boxes. Tip: This is used for special purposes only, like adding new objects that may not be ready for general use or for objects to be used only by system processes; such as workflows. • **USER** The editing tool will appear in selection lists in the appropriate dialog boxes.

4. Click **Finish**.

Install editing tool plug-ins

Some editing tools require you to install additional plug-ins within Content Management. An editing tool plug-ins adds a **Teamcenter CMS** menu to the editing tool. This menu allows the author to perform actions to Teamcenter Content Management within the editing tool.

Note:

If you use **Teamcenter XMetaL Client**, do not install the XMetaL plugin in the Teamcenter interface.

1. Choose **Window→Preferences**.
2. In the **Preferences** dialog box, expand **Content Management**, and expand **Edit**.
3. Select the editing tool you want to install, and enter the appropriate information.

   Example:

   If you are installing XMetaL, enter the path to the tool's install location, for example: C:\Program Files\XMetaL 9.0\Author.

4. (For most tools) Click **Install**.
5. Click **Apply**.
6. Click **OK** to close the **Preferences** dialog box.

Tip:

If you are using XML Spy and the **Teamcenter CMS** menu does not appear on the toolbar, for the XML Spy tool, click **Uninstall** and then click **Install** to reinstall the required plug-in. Then in XML Spy, choose **Tools→Customize→Toolbars**, and then select **Reset All**.

22. Enabling content publishing

Tasks for enabling publishing content

Publishing tools are used to render content to PDF, HTML, and other formats, and are launched when you publish content. A publishing tool is also used to view content.

Publishing tools can be installed on the client or a server, following the supplier's instructions. Create a publishing tool object for each publishing tool installed.

To enable authors to publish content you must do the following:

- Create a **stylesheet** for each type of output.
- Create a publishing tool for each type of output and relate it to the stylesheet revision.
- Relate the stylesheet revision to topic types and publication types.
- (Optional) To enable authors to save output to a file system where your Web server is deployed, you must **enable publishing to a file system**.
- If you are upgrading from a version of Content Management prior to 9.0, you must reselect the **Tool Activation** option for all the publishing tools you previously created.
- To enable the publishing of S1000D content on Linux, you must ensure that the correct directories exist on the computer running the Teamcenter server if the Java version used by the Teamcenter server is updated.

  Example:

  If the system is updated to Java 1.8.x, these directories must exist:

  /usr/lib64/java-1.8.0

  /usr/share/java-1.8.0

Publishing tool examples

The following tools are examples of publishing tools that work with Content Management. Additional tools may be defined to work with Content Management.

- Antenna House XSL Formatter

- Apache FOP (Formatting Objects Processor)
- DITA Open Toolkit

Pre-configured publishing tool objects

If you import Content Management **administration data**, you import the following pre-configured publishing tool objects:

> Note:
>
> Verify that the tool paths assigned to imported publishing tools match the paths for your installation of the tools.

- Base Content Management tools:
  - **FOP**

    Supports Formatting Objects Processor, a Java application that converts XSL Formatting Objects files to PDF.

    > Note:
    >
    > If you notice that certain fonts in a particular language do not appear in your published content, check if the processor you are using supports all the language characters.
    >
    > For example, if you are using the Apache FOP processor and notice that certain Chinese characters are displayed by the character #, check the Apache FOP documentation at **https://xmlgraphics.apache.org/fop/2.6/fonts.html**.

  - **InternalViewer**

    An internal tool that publishes XML content to the **Preview** view. This tool is configured to use Internet Explorer and HTML styled content.

  - **PDF serverside**

    Supports Antenna House XSL Formatter, one of many available publishing applications that can be purchased from a third-party supplier, installed on a server, and used to publish content to PDF.

  - **serverside HTML publish**

    An internal tool that publishes content to HTML. Processing occurs on the Teamcenter server, and the tool is configured to use XSLT stylesheets.

- DITA Content Management tools:

Note:

- If you are using DITA Open Toolkit publishing tools on a Linux system, the tool command must be **ant** and not **ant.bat** as it is for Windows systems.
- Set the path for `DITA_ANT_HOME` in the *Dispatcher_Root***Module\Translators\-contmgmtpublish\config\contmgmtpublish_config.properties** file. The path must be set to the Ant directory in the DITA OT.
- Content Management supports version 2.4.6 of the **DITA Open Toolkit (OT) publishing tools**.

  To use an older version of DITA OT, copy the older version of DITA OT to: the *Dispatcher_Root***Module\Translators\contmgmtpublish\lib** folder.

  - **DITA Open Toolkit XHTML**

    A tool for publishing DITA content to XHTML.

  - **DITA Open Toolkit PDF**

    A tool for publishing DITA content to PDF.

  - **DITA Open Toolkit RTF**

    A tool for publishing DITA content to RTF.

- S1000D 4.0/4.1 Content Management tools:

  - **S1000D Publish XHTML**

    A tool for publishing S1000D content to XHTML.

  - **S1000D Publish PDF**

    A tool for publishing S1000D content to PDF.

Create a publishing tool object

1. Select the folder in which you want to create the new publishing tool object, and then either click **New Administrative Item...**  , or choose **File→New→New Administrative Item**.

2. In the **New Administrative Item** dialog box, expand **Complete List**, click **Publishing Tool**, and then click **Next**.

3. In the dialog box, do the following:

For this option	Do this
Name	Type a name for the publishing tool.
Tool Activation	Select the **publishing tool activation** that corresponds to the publishing tool object you are creating. Note: When you upgrade from a version of Content Management prior to 9.0, you must reselect the **Tool Activation** option for all the publishing tools you previously created.
Tool Command	Type the command used by Content Management to start the tool. This includes the start command (for example, an **.exe** file) and any required or optional parameters for the tool you are using. The valid command line parameters for the tool are typically documented by the tool supplier. If you use command line parameters for one or more of the following, they must be entered in this order. If your tool cannot accept the parameters in this order, you must use a custom launch program to publish files. a. XML file b. Stylesheet file c. Output file Tip: If you are using DITA Open Toolkit publishing tools on a Linux system, the tool command must be **ant** and not **ant.bat** as it is for Windows systems.
Tool Path	Type the path where the tool's start file (for example, an **.exe** file) is stored. For example: C:\Program Files\Antenna House\AHFormatterV62
Graphic Priority Table Reference	Select the **graphic priority list** to associate with the tool. The graphic priority list is used to select the **graphic options** to be downloaded to the content when it is launched with the publishing tool.
Download graphics	Select either **True** or **False**, to specify whether graphics that are referenced by the content are downloaded from Teamcenter when the publishing tool is launched. This allows

For this option	Do this
	for the most recent graphics to be distributed to users from a central location.
Download schema	Select either **True** or **False**, to specify whether schemas on which content is based are downloaded from Teamcenter when the publishing tool is launched. This allows for the most recent schemas to be distributed to users from a central location.
Download stylesheet	Select either **True** or **False**, to specify whether stylesheets are downloaded from Teamcenter when the publishing tool is launched. This allows for the most recent stylesheets to be distributed to users from a central location.
Local Indicator	Select either **True** or **False**, to specify whether the publishing tool is run from the client instead of being run from a publishing server.
Number of Instances	Type the number of licenses you have for this tool, if you want to store that value here; however it is not enforced by the system.
Processing instruction	Type any necessary processing instruction for the publishing tool. The processing instruction is written into the content files when opened with the publishing tool.
System Usage	Select one of the following: • **SYSTEM** The publishing tool will *not* appear in selection lists in dialog boxes. Note: This is used for special purposes only, such as for adding new objects that may not be ready for general use or for objects to be used only by system processes, such as workflows. • **USER** The publishing tool will appear in selection lists in the appropriate dialog boxes.

4. Click **Finish**.

Publishing tool activations

The following are the provided tool activations you may select when **creating a new publishing tool object**:

> Note:
>
> Customizers may add additional activations.

If the publishing tool is	Select this tool activation
Installed on the client and can act as a command line tool or has a command line interface.	**com.teamcenter.rac.contmgmtbase.publish.publisher .ClientExternalCmdPublisher**
A tool that publishes content to the **Preview** view.	**com.teamcenter.rac.contmgmtbase.publish.publisher .ClientInternalPublisher**
Formatting Objects Processor (FOP)	**com.teamcenter.contmgmtpublish.prepare .FOPPublisher**
A plain text publishing tool.	**com.teamcenter.contmgmtpublish.prepare .PlainTextPublisher**
Installed on the publishing server and can act as a command line tool or has a command line interface.	**com.teamcenter.contmgmtpublish.prepare .ServerExternalCmdPublisher**
An internal publishing tool that is installed on the publishing server.	**com.teamcenter.contmgmtpublish.prepare .ServerHtmlPublisher**
Server side XSL Formatter tool	**com.teamcenter.contmgmtpublish.prepare .ServerXslFormatterWithStyleSheetParams**
DITA Open Toolkit publishing tool installed on the publishing server, for DITA content.	**com.teamcenter.ditapublish.prepare .DitaMapExternalCmdPublisher**
S1000D publishing tool installed on the publishing server, for S1000D content.	**com.teamcenter.s1000dpublish.prepare .S1000DExternalCmdPublisher**

Add a custom publishing tool activation

Customizers may add publishing tool activations for purposes not met by the **provided publishing tool activations**.

1. Create a jar file with customized java class(es), and save it in the *TC_ROOT***portal\plugins** folder.

2. In the Business Modeler IDE, add the package name of the jar file to the values for the **Ctm0PubToolActivation** list of values (LOV).

3. Create a **new publishing tool** with the newly created tool activation.

   Note:

   For server side publishing tools, select **False** for the **Local Indicator**. For client side publishing tools, select **True**.

Enable publishing through Teamcenter Dispatcher

When content is published to HTML, PDF, or another format, it can be published at the client or through Teamcenter Dispatcher, if it is available.

- To enable publishing through Dispatcher, do the following:
  - Install Dispatcher.
  - When installing Dispatcher, enable the **ContMgmtPublish** translator.
  - Configure Dispatcher.
  - Configure Teamcenter Dispatcher for asynchronous processing. With asynchronous processing enabled, when you publish content through Dispatcher, the **Publish** dialog box closes when you click **Finish**.
  - (Optional) If your organization publishes very large publications, increase the value of the **MaximumProgress** property in the *DISPATCHER_ROOT\Module\conf\transmodule.properties* file.

Enable publishing to a file system

When content is published to HTML or PDF format, the output can be saved to a file system where a distribution web server is deployed. When new versions of output are published, the system creates a folder for each new version.

Note:

To access files published to a file system for access by a distribution web server, you must configure a web server that can access those files.

- To enable publishing to a file system, create the following preferences:

  You can set these preferences by choosing **Edit→Options→Content Management→Publish** tab.

  - **publish_to_file_system**

    Enables publishing to a file system.

  - **publish_to_file_system_loc**

    Specifies the name of the parent folder to store published content.

  - **deploy_absolute_path**

    Specifies the file path to the folder specified in the **publish_to_file_system_loc** preference for published content. This path is platform-specific.

  - **deploy_link_prefix**

    Specifies the URL to connect to the folder storing published content.

Enable schema validation during publishing

When you publish a topic or document, you can configure whether validation is required or not.

Validating what your authors writes is a requirement for XML content. You can choose when and how you want the schema validation to occur.

Procedure

1. Update the **ctm0Schema_Validation** preference with the following values, based on how strict your requirements for validation are:

Schema validation action	Preference value
Publish even though there are schema validation errors. Publish the content even if schema validation errors exist. A log is provided with errors to support correcting content.	**0**
Do not publish if there are schema validation errors.	**1**

Schema validation action	Preference value
If validation errors exist, publish will not complete. A log is provided with errors to support correcting content.	
Skip schema validation. Publish without checking schema validation on content. Publish will complete unless other errors prevent compilation and publish.	**2**
Allow user to choose how to validate the schema. The user will be shown a list of schema validation options.	**-1**

23. Change the tool for automatically publishing the topic edited by a user

By default, Teamcenter provides the **DITA Open Toolkit PDF** and **S1000D Publish PDF** publishing tools for automatically publishing the DITA topics and S1000D and S1000D 4.x/5.x data modules that a business user has edited and checked in.

However, as per your business requirement, you can change the default publishing tools by using the **Ctm0PublishingToolPair** preference. In the preference, if you specify a PDF publishing tool, the published document is in PDF format. If you specify an HTML publishing tool, the published document is in HTML format.

Procedure

1. Search for and open the **Ctm0PublishingToolPair** preference.

2. Add the publishing tool of your choice as a value to the preference.

   You can specify multiple values in the preference. Each value is a class and publishing tool pair in the format **<class name:publishing tool name>**. If the class of a topic is the same as the class name or the subclass of the class name, the corresponding publishing tool is used for publishing the edited topic.

24. Create a comparing tool object

Comparing tools are used to compare differences in topics. A comparing tool is installed on each client, following the supplier's instructions.

> Note:
>
> You may have only one comparing tool object in Content Management.

1. Select the folder in which you want to create the comparing tool object, and then either click **New Administrative Item...**  , or choose **File→New→New Administrative Item**.

2. In the **New Administrative Item** dialog box, expand **Complete List**, click **Comparing Tool**, and then click **Next**.

3. In the dialog box, do the following:

For this option	Do this
Description	Type a description for the comparing tool.
Graphic Priority Table Reference	Select the **graphic priority list** to associate with the tool. The graphic priority list is used to select the **graphic options** to be downloaded to the content when it is launched with the comparing tool.
Name	Type a name for the comparing tool.
Processing instruction	Type any necessary processing instruction for the comparing tool. The processing instruction is written into the content files when opened with the comparing tool.
System Usage	One of the following: • **SYSTEM** The comparing tool will *not* appear in selection lists in dialog boxes. > Tip: > This is used for special purposes only, such as for adding new objects that may not be ready for general use or for objects to be used only by system processes such as workflows. • **USER**

For this option	Do this
	The comparing tool will appear in selection lists in the appropriate dialog boxes.
Tool Command	Type the command used by Content Management to start the tool. This includes the start command (for example, an **.exe** file) and any required or optional parameters for the tool you are using. The valid command line parameters for the tool are typically documented by the tool supplier. Some tools require the use of at least two parameters, such as for placeholders for the left and right file names. Example: `wincmp3.exe "%1$s" "/=%4$s" "%2$s" "/=%5$s" /1 /R2` In this command line, the following parameters are used: `"%1$s"` and `"%2$s"` are placeholders for the left file name and the right file name. Quotation marks are often used with parameters to avoid issues with folder names that include space characters. `"/=%4$s"` and `"/=%5$s"` are captions for the left and right files. `/1` allows only one instance of the tool. `/R2` specifies that the right pane is read-only. **The valid parameters depend on the tool you use.**
Tool Output State	*Not currently used*
Tool Path	Type the path where the tool's start file (for example, an **.exe** file) is stored. For example: C:\Program Files\CompareIt

4. Click **Finish**.

25. Create a viewing tool object

Viewing tools are used to run an external viewing tool to view topics. This is useful when a quick or remote preview session is needed, without the processing time required to generate a PDF document. Viewing tools are installed on each client, following the supplier's instructions. Create a viewing tool object for each viewing tool installed.

In order for users to access a viewing tool, you must write the rich client java class to launch the tool, and then define the tool to use that class.

> Note:
>
> A viewing tool is an object different from that used to view topics in the **Preview** view. Previewing content is enabled with a **publishing tool**.

1. Select the folder in which you want to create the new viewing tool object, and then either click **New Administrative Item...**  , or choose **File→New→New Administrative Item**.

2. In the **New Administrative Item** dialog box, expand **Complete List**, click **Viewing Tool**, and then click **Next**.

3. In the dialog box, do the following:

For this option	Do this
Description	Type a description for the viewing tool.
Graphic Priority Table Reference	Select the **graphic priority list** to associate with the tool. The graphic priority list is used to select the **graphic options** to be downloaded to the content when it is launched with the viewing tool.
Name	Type a name for the viewing tool.
System Usage	One of the following: • **SYSTEM** The viewing tool will *not* appear in selection lists in dialog boxes. Tip: This is used for special purposes only, such as for adding new objects that may not be ready for general

For this option	Do this
	use or for objects to be used only by system processes such as workflows. • **USER** The viewing tool will appear in selection lists in the appropriate dialog boxes.
Tool Activation	Type a tool activation if one is required for the tool you are using. If no tool activation is entered, the tool is launched externally using the **Tool Command**.
Tool Command	Type the command used by Content Management to start the tool. This includes the start command (for example, an **.exe** file) and any required or optional parameters for the tool you are using. The valid command line parameters for the tool are typically documented by the tool supplier.
Tool Path	Type the path where the tool's start file (for example, an **.exe** file) is stored.
Viewable Content Type	Select type of file to which you want to assign a viewing tool.

4. Click **Finish**.

26. Exchange Content Management data between multiple sites

How multisite data exchange for Content Management works

- **Extract Content Management data from Teamcenter using the multisite_automation.pl perl script**

  By default, this script extracts Content Management data from Teamcenter that has changed in the last 24 hours and passes it to the **data_share** utility to transfer the data to the target sites. You can customize what data is exported and when by modifying the arguments in the script file.

- **Transfer data to the target site using the data_share utility**

  This utility is used for multisite data transfers. It is invoked from the *multisite_automation.pl* perl script. The data is imported into the target site and the relationships between the data is maintained using the default Content Management closure rules.

You also have an option of downloading the data as TC XML files and then importing that data into the target site.

How administrators can set up and exchange Content Management data?

- Whenever you, as the administrators want to synchronize data between sites, you must run the *multisite_automation.pl* perl script. To automate the synchronization, run this script as a Windows service or as a Linux cron job.

  This script file is located in the *TC_ROOT\install\contmgmtbase* directory.

- You can customize the data, workspace objects, sites, output directory, and age of data to synchronize in the script file. You can also specify whether the data in exported as a TC XML file.

- In the script file, you can customize how the **data_share** utility behaves by specifying the options you want. For example, you can specify custom check-in and check-out behavior.

- To manually download data as TC XML files, use the **-offline** argument in the **data_share** utility.

- If you have downloaded data as TC XML files, you can import the data into the remote site using the **-optionset** argument of the **data_share** utility.

- When data is imported at the target site, the object and relationships are set per the default Content Management closure rules. You can specify additional closure rules by modifying the **MultiSiteDefaultCR** closure rule in PLM XML/TC XML Export Import Administration.

For more information, see the *multisite_automation.pl* file located in the *TC_ROOT\install\contmgmtbase* directory.

Example:

Synchronizing data between sites:

```
perl multisite_automation.pl -u=dba -p=dba -g=dba
-dt="21-Jan-2019 17:00:00" -sites=site1,site2,site3
-o="C:\temp\multisite_output\"
```

Download data to a directory:

```
perl multisite_automation.pl -u=dba -p=dba -g=dba
-dt="21-Jan-2019 17:00:00" -sites=site1,site2,site3
-offline -o="C:\temp\export_dir"
```

Import data that was downloaded to a directory:

```
data_share -optionset -u=dba -p=dba -g=dba -site=site1
 -filename=C:\temp\test_cm_multisite\DC_TopicRevision.txt
-include_bom -all_revisions -f=send
```

27. Setting up translation features

Tasks required to set up translation features

Teamcenter provides support for the translation of content from a source language to additional languages and the publication of documents in different languages. To manage the translation process, authors create translation orders and send them to a translation office along with topics to be translated to other languages. Before authors can work with translations, the application administrator must do the following:

- **Establish the languages that are available for translations**, to ensure that each language needed for translation is represented by a language item in the database.
- **Create translation offices**, ensuring that each translation office used for translations is created in the database.
- **Setting the graphic priority for translation orders**.
- **Configure your Teamcenter installation to support the UTF-8 character set**.

Establishing languages available for translations

Each language used for translations in Content Management must be represented by a language item in the database and managed in the Organization application. Each language item is identified by a unique language name and stores such attributes as the ISO language code and ISO country code. You can import language items from the **administration data** included with Content Management.

Create a translation office

A translation office is an organization that translates content into other languages. A translation office object contains the necessary information to manage the translation process in Content Management for a specific translation office, including which languages can be selected for translations. Translation office items have translation office revisions.

1. Select the folder in which you want to create the new translation office, and then either click **New Administrative Item...**  , or choose **File→New→New Administrative Item**.

2. In the **New Administrative Item** dialog box, expand **Complete List**, click **Translation Office**, and then click **Next**.

3. In the dialog box, do the following:

For this option	Do this
ID	Either type an ID for the translation office, or leave the box blank so that the next available item ID is automatically assigned.
Revision	Either type a revision for the translation office, or leave the box blank so that the next available revision is automatically assigned.
Name	Type the name that is assigned to the translation office object.
Translation Office Title	Type the title of the translation office.
Address	Type the address of the translation office.
Contact Name	Type the contact name for the translation office.
Phone	Type the phone number of the translation office.
Website	Type the website address of the translation office.
Email Inbox	Type the e-mail address of the translation office to which a translation order can be sent.
Deliver Composed Content	Select either **True** or **False**, to specify whether content is typically delivered to this translation office as composed content. This value is for informational purposes only; it does not affect the processing of translation deliveries.
Deliver Decomposed Content	Select either **True** or **False**, to specify whether content is typically delivered to this translation office as decomposed content. This value is for informational purposes only; it does not affect the processing of translation deliveries.
Include Supporting Data	Select either **True** or **False**, to specify whether the content delivered includes supporting data. This value is for informational purposes only; it does not affect the processing of translation deliveries.

4. Click **Finish**.

5. Select the translation office revision in the **Home** component view, and open the **Summary** view.

6. Under **Language Table**, click **Add**.

7. In the **Languages Table** dialog box, from the **Language Reference** list, select a language that is supported by the translation office.

8. For the **Review Ordered** option, select whether the translation office typically offers a review of translations in this language. This value is for informational purposes only; it does not affect the processing of translation deliveries.

9. Click **Finish**.

10. Add additional languages as necessary.

Setting the graphic priority for translation orders

When translation orders are created, a graphic priority list is assigned. To specify which graphic priority list is assigned, you must edit the **ctm0_translation_graphicpriority** preference.

If this preference is not set, the **EDIT** graphic priority list is used.

Configure your Teamcenter installation to support the UTF-8 character set

If you import translated content in languages that require multibyte characters, such as Russian and Chinese Simplified, you must configure your Teamcenter installation to support the UTF-8 character set to ensure that titles and other properties on this content display correctly in your environment.

28. Transforming XML content

Using XSLT scripts to transform XML

Transformation policies can be used to make specific modifications to XML or SGML content during the loading or storing of content. This can be done using XSLT scripts that transform XML from one structure to another. To use an XSLT script in a transformation policy, you create a **procedure** object that contains the script, and then add the procedure to the transformation policy. Transformation policies contain a table of entries, where each entry contains a user action and a procedure name. Procedures are applied to the content when the action is used.

> Caution:
>
> Configuration of procedures and transformation policies should be performed by experienced administrators only. Incorrect application of these procedures and policies can cause errors in your data.

You can also use procedures in **XML attribute mappings**. XSLT scripts in attribute mappings enable you to perform more complex tasks than are possible with a basic attribute mapping. For example, a script may be used to change the values of attributes before they are extracted from the XML.

Creating XML transformations for publishing and exporting

You can create transformation policies that authors select when they publish or export content. For example, you can create a transformation policy that converts units of measure in the author's document from US standard to metric. You create these transformation policies the same way as other transformation policies: Create a **procedure** that contains the XSLT script, and then add the procedure to the **transformation policy**. When you create the transformation policy, you must select **Publish**, **Preview**, or **Export** for the **User Action**.

This example of an XSLT stylesheet contains the scripts to convert inches to centimeters and feet to meters. The schema you use must include the elements used by the script.

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

      <xsl:template match="node()|@*">
            <xsl:copy>
                  <xsl:apply-templates select="node()|@*"/>
            </xsl:copy>
      </xsl:template>

      <!-- Convert inches to cm -->
      <xsl:template match="measure[@units='inches']">
            <measure units="cm">
                  <xsl:value-of select="format-number(. * 2.54,'#.00')"/>
            </measure>
      </xsl:template>

      <!-- Convert feet to meters -->
      <xsl:template match="measure[@units='feet']">
            <measure units="meters">
                  <xsl:value-of select="format-number(. div 3.28084,'#.00')"/>
            </measure>
      </xsl:template>

</xsl:stylesheet>
```

The author's document would include the elements in the XML, for example:

```
<measure unit="feet">5</measure>
<measure unit="inches">62</measure>
```

Create a procedure for XML transformation

1. Select the folder in which you want to create the new procedure, and then either click **New Administrative Item...**, or choose **File→New→New Administrative Item**.

2. In the **New Administrative Item** dialog box, expand **Complete List**, click **Procedure**, and then click **Next**.

3. In the dialog box, type the following:

In this box	Do this
ID	Either type an ID for the procedure, or leave the box blank so that the next available item ID is automatically assigned.
Revision	Either type a revision for the procedure, or leave the box blank so that the next available revision is automatically assigned.
Name	Type a name for the procedure.

In this box	Do this

Procedure Usage	Select one of the following to define what the procedure is used for:
	• For procedures in XML attribute mappings:
	  • **DC_ATTRIBUTEMAPPING_COPY_FROM_OBJECT_TO_TAG**
	  Copies an attribute value from a Teamcenter object to the specified XML tag in the content. This type does not support table attributes.
	  • **DC_ATTRIBUTEMAPPING_COPY_FROM_TAG_TO_OBJECT**
	  Copies data from an XML tag in the content to a specified attribute of the Teamcenter object. This type does not support table attributes.
	  • **DC_ATTRIBUTEMAPPING_GET_FROM_TAG**
	  Gets a value from an XML tag in the content. This type does not support table attributes.
	  • **DC_ATTRIBUTEMAPPING_REMOVE_FROM_TAG**
	  Removes data from an XML tag in the content.
	  • **DC_ATTRIBUTEMAPPING_SET_TO_TAG**
	  Sets a value in an XML tag in the content. This type does not support table attributes.
	• For procedures in transformation policies:
	  • **DC_CONTENT_TRANSFORMATION**
	  For XML transformations, transforms content during an import or export or during publishing.
	• Reserved for future use:
	  • **DC_OPERATOR_PROCEDURE**

In this box	Do this
	• **DC_OBJECT_MODIFICATION** • **DC_PRE_CREATE**
Select Content	Browse to and select the file containing the XSLT script.

4. Click **Finish**.

Create an XML transformation policy

1. Select the folder in which you want to create the new transformation policy, and then either click **New Administrative Item...** , or choose **File→New→New Administrative Item**.

2. In the **New Administrative Item** dialog box, expand **Complete List**, click **Transformation Policy**, and then click **Next**.

3. In the dialog box, in the **Name** box, type the name for the policy.

4. (Optional) In the **Admin Comment** box, type a comment that you would like to store with the transformation policy.

5. Click **Finish**.

6. Open the **Summary** view and select the transformation policy.

7. In the **Summary** view, click **Add**.

8. In the **Transformation Policy Table Entry** dialog box, from the **User Action** list, select the user action for which the transformation will run.

9. From the **XML Procedure** list, select the procedure that runs when the user actions occurs.

10. Click **Finish**.

11. For each user action you want to add to the policy, repeat steps 7 - 10.

> Note:
>
> To remove an item, click it, and click **Delete**.