EMC® Documentum®
Documentum Compliance Manager

Version 6.5

Administration Guide
P/N 300-007-342-A01

EMC Corporation
Corporate Headquarters:
Hopkinton, MA 01748-9103
1-508-435-1000
www.EMC.com

Published July 2008

EMC believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED AS IS. EMC CORPORATION MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Use, copying, and distribution of any EMC software described in this publication requires an applicable software license.

For the most up-to-date listing of EMC product names, see EMC Corporation Trademarks on EMC.com.

Table of Contents

List of Figures

List of Tables

Preface

This manual describes how to configure DCM and how to perform DCM-specific administration tasks.

Intended Audience

This manual is primarily intended for system administrators and Documentum administrators.

Revision History

The following changes have been made to this document.

Revision History

Revision Date	Description
July 2008	Initial Publication.

Related Documentation

The information in this guide refers to concepts and procedures described in *Documentum Compliance Manager User Guide*. This guide contains information, instructions, and procedures for the normal system administration tasks of a Documentum Compliance Manager application. It gives you an overview of the DCM's configuration and some guidelines for making your own configuration decisions.

How to Use the DCM Administration Guide

Documentum Compliance Manager integrates with Webtop, and because DCM extends Webtop functionality all available functionality in the DCM user interface (UI) is accounted for in this user guide. To become better acquainted with the DCM user interface you should also consider reading Appendix D, Icons.

Chapter 1

About Documentum Compliance Manager

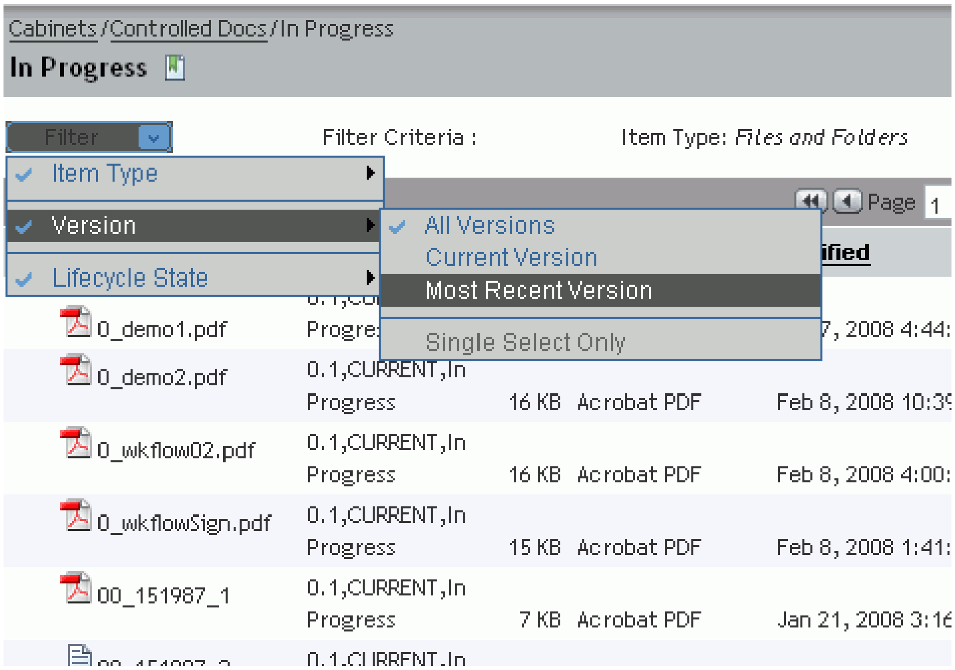
This chapter introduces you to Documentum Compliance Manager (DCM), describes what DCM is, and how it works. Topics discussed in this chapter include:

New in DCM 6.5

- Native DCM code for e-signatures and watermarks/overlays are now supported using PDF Stamping Services (PSS). This built in functionality replaces dependencies on Trusted Content Services (TCS), third party software.
- Certification with Branch Office Caching Services (BOCS). BOCS enables customers to support a distributed DCM environment which provides local response time when accessing content at remote locations.
- Certification with Retention Policy Services (RPS). DCM and RPS can be configured so that DCM controls content from creation through approval and use while RPS controls the content after it is no longer in use until disposition. The "DCM Obsolete" state can be configured such that content is moved to an RPS folder configured with a retention policy. The DCM Obsolete folder for example can be renamed as DCM Records where content remains until final disposition is performed to either transfer or destroy the content.

- Introduction of the "Most-Recent" version label. The Most-Recent version label is intended primarily to prevent confusing two versions of the same content in a DCM lifecycle with the same "CURRENT" version label. Version 1.0 of a controlled document in the Effective state for example is labelled CURRENT and could later be superseded by a newer more recent version 1.1 of the same document now in the In-Progress state. The Most-Recent version label is now used to avoid duplicate CURRENT labels in the same version tree. The dcm.properties file includes settings to support the Most-Recent version label for DCM system configurations. An additional filter option "Most-Recent Version" is also included to give DCM D6.5 users the option to choose the "Most Recent Version" as a convenient way to find the latest object in the version tree.

Figure 1. Most Recent Version



Cabinets / Controlled Docs / In Progress

In Progress

Filter Criteria : Item Type: *Files and Folders*

0_demo1.pdf Progre 7, 2008 4:44

0_demo2.pdf 0.1,CURRENT,In Progress 16 KB Acrobat PDF Feb 8, 2008 10:39

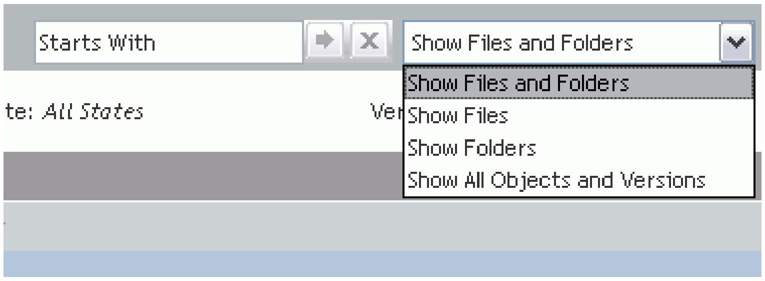
0_wkflow02.pdf 0.1,CURRENT,In Progress 16 KB Acrobat PDF Feb 8, 2008 4:00

0_wkflowSign.pdf 0.1,CURRENT,In Progress 15 KB Acrobat PDF Feb 8, 2008 1:41

00_151987_1 0.1,CURRENT,In Progress 7 KB Acrobat PDF Jan 21, 2008 3:16

- "*Show All Objects and Versions*" is a new filter option provided for extended filter control when displaying results for various item types in the content pane. It is listed at the bottom of the filter list box displayed on the user interface just above the right-hand corner of the content pane.

Figure 2. Show All Objects and Versions



- Enhanced Notifications user interface. Notifications in the Administration node now provides more options to handle notifications so that administrators can specify when to send emails and to whom.
- Improved selection capabilities for configuring DCM relationships. Enhanced filtering capability now reduces the selection list returned and the time taken for the response. The list returned, on the Attachments page for a controlled document as displayed from its Properties, can be narrowed to a specific document class to avoid a list of all document classes.

How Does DCM Work?

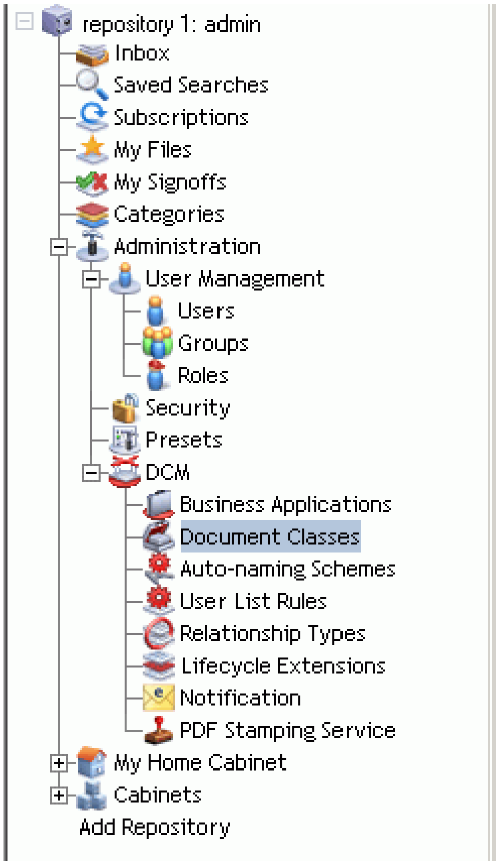
Documentum Compliance Manager (DCM) extends Webtop capabilities to provide enhanced document management capabilities. Extended capabilities add features to manage controlled documents using workflows and signoffs.

Although the DCM and the Webtop user interface (UI) appear similar, DCM may add features or modify existing features, for example: checkin of a controlled document, My Signoffs node, and reports. Unlike Webtop, DCM includes enhanced workflow management to enforce compliance of controlled documents for legal purposes.

Note: If you can see My Signoffs and Compliance, you are in the DCM user interface. The presence of the DCM folder in the Administration node is another indication that you are in the DCM user interface. These visual indicators are specific to DCM. Organization of the cabinets, Change Notices, Change Requests, Controlled Documents, etc. is also characteristic of DCM out-of-the-box (OOTB) but not limited to this structure once OOTB and customized. The Home Cabinet also consists of two folders typical to DCM: Workspace Customizations and Startup Items.

Figure 3. DCM menu bar

Figure 4. DCM administration navigation pane



DCM lets you manage large volumes of documents that must comply with regulatory and quality standards. DCM defines document properties, access permissions, and change management processes for each document type during each stage of its lifecycle from initial draft to publication through retirement.

DCM uses a defined processes for the creation, review, and signoff of controlled documents. DCM lets users read documents and issue change requests. DCM lets coordinators create and edit documents and route them for review and signoff.

DCM allows you to establish relationships between documents to help automate review cycles and document lifecycle changes, enhancing workflow functionality.

DCM also includes auditing and electronic sign off. DCM's auditing and electronic sign-off features conform to the regulatory requirements mandated in the *Code of Federal*

Regulations, Title 21, Food and Drugs, part 11, "Electronic Records; Electronic Signatures" (21CFR11).

Once your administrator configures DCM, DCM automates many processes for you, conforming to your company's business rules.

DCM assists you in the following ways:

- Distributes the most recent documents that have the "**Most-Recent**" version label, to the right people for you in a timely and consistent manner.

  Note: The "Most-Recent" version label differentiates the "CURRENT" version of an effective document, 1.0 for example, in the Effective state of a DCM lifecycle from a newer more recent version, 1.1 for example, of the same document now in the In-Progress state. Since documents regardless of there version number in the Effective state are labelled CURRENT, the Most-Recent version label is used against a newer version of the same document in the same version tree.
- Helps you to manage document changes.
- Uses a defined process that incorporates controlled documents, change requests, change notices, and supporting documents to record and track changes.
- Assures that documents are reviewed and signed by the appropriate people in a consistent, controlled process.
- Allows consumers to read documents and issue change requests.
- Allows coordinators to create and edit documents and issue change notices, as well as route documents for review and signoff.
- Provides automated control over who may view, print, or change electronic documents.
- Provides built-in reports to help you track your documents.
- Automatically generates audit trails for events such as the creation, modification and signoff or sign-off of controlled documents.
- Allows you to add custom headers, footers and watermarks to published documents when they are viewed or printed. (Optional module)
- Restricts printing, tracks printed copies and recalls printed copies for controlled documents. (Optional module)

Using the various configuration options included with DCM, you can configure a business application that defines automatic properties and processes for specified document types.

Each DCM business application includes the following features:

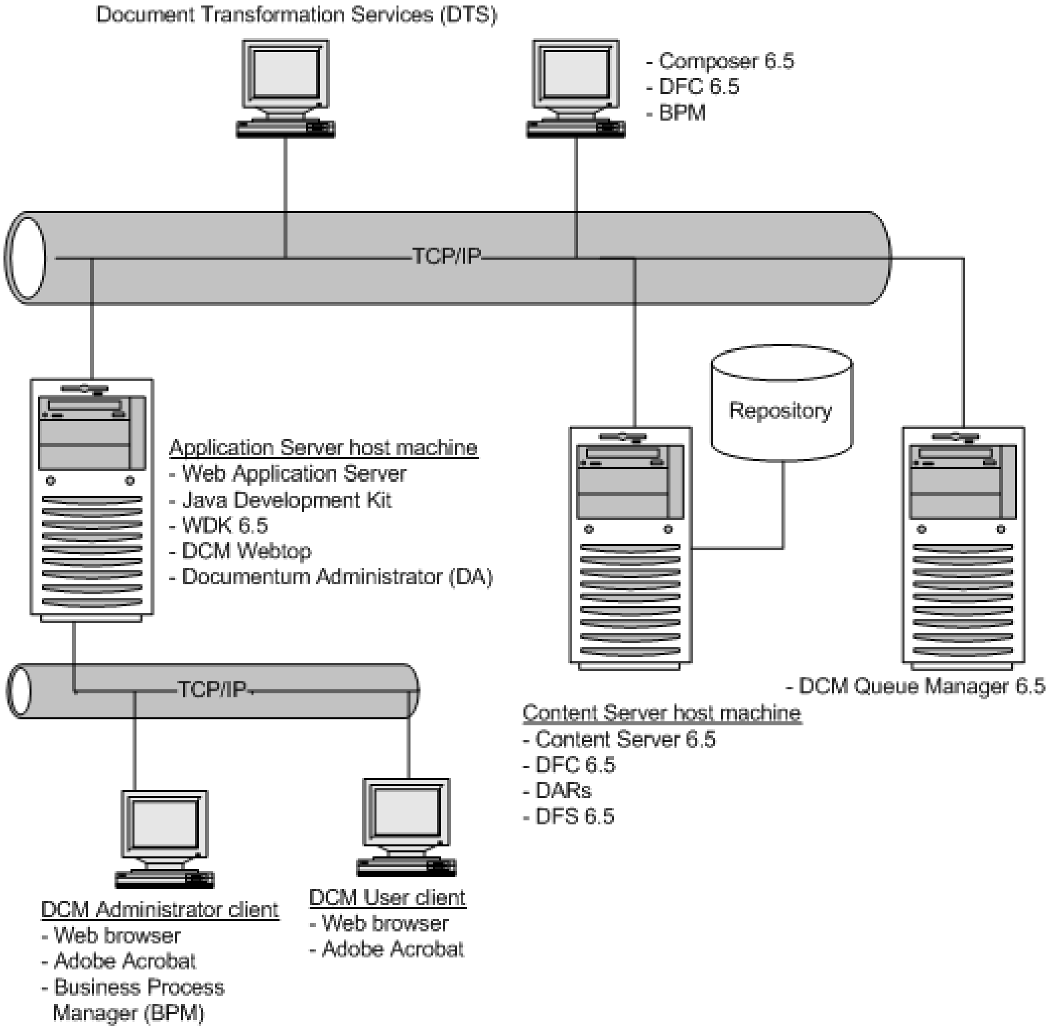
- Controlled document creation, editing, viewing, and printing
- Auditing, authentication, and electronic sign-off
- Automatic document lifecycle management
- Automatic document naming, versioning, and other property settings

- Workflow management and automation
- Automated distribution and notification functionality
- Controlled central printing services

Typical Installation Configuration

The following figure show the basic installation configuration for a DCM system:

Figure 5. Typical Installation Configuration



How Auditing Works with DCM

Auditing is a security feature that allows you to monitor events that occur in a repository or application. Events are operations performed on objects in a repository or something that happens in an application. Auditing an event creates an audit trail, which is a history in the repository of the occurrence of the event. Creating an audit trail is a useful way to prove compliance with a business rule.

DCM’s auditing and electronic sign-off features conform to the regulatory requirements mandated in the *Code of Federal Regulations, Title 21, Food and Drugs, part 11*, "Electronic Records; Electronic Signatures" (21CFR11).

DCM works with Content Server to create audit trails using Trusted Content Server, an optional Content Server feature. Trusted Content Server is a set of features that you unlock by providing a license key during Content Server installation. These features are:

- Encrypted filestores
- Secure (SSL) DMCL or LDAP communications
- Electronic signatures

Trusted Content Server runs behind the scenes for most operations.

To audit a DCM event, the event must be registered with Documentum Administrator by a user who connects with the following extended privileges: Config, View, and Purge Audit.

If you enabled Trusted Content Server, the electronic signature feature is added to the existing auditing feature. The process of electronic sign-off automatically creates the audit records.

For an overview of how Trusted Content Server and Content Server’s auditing features work, the default functionality, and an overview of what can be customized, see the chapter named *Security Services* in the *Documentum Content Server Fundamentals*. For information and step-by-step instructions to implement a custom signature page, see the *Documentum Content Server Administrator’s Guide*.

DCM, since 5.3 SP1, also supports PDFsign services for e-signatures on the Trusted Content Server. Refer to the DCM Installation Guide to install and configure PDFsign and also integrate it with the DCM Application Server.

What Events Are Audited in DCM

The following table describes which events are audited in DCM.

Table 1. Events audited in DCM

Objects	Event name	Event meaning	Event Attributes Use	Event specific info
Common for Controlled Documents, CN and CR	dcm_acquire_ signoff	Signoff initiated for controlled document	string_2: user name string_4: group name string_5: return code, “successful" or "failure"	Two entries are recorded, one for the document which is the parent and one for the child which is the signatory (user or group)
Common for Controlled Documents, CN and CR	dcm_add_ signatory	Signatory added to controlled document	string_2: user or controlled document name string_3: Lifecycle stage string_5: return code, “successful" or "failure"	Two entries are recorded, one for the document which is the parent and one for the child which is the signatory (user or group)
Common for Controlled Documents, CN and CR	dcm_attach_ lifecycle	Lifecycle attached to document class	string_1: Lifecycle name string_5: return code, “successful" or "failure"	
Common for Controlled Documents	dcm_authentication_failed	Signatory authenticated before signing off	string_1: Signatory name string_2: signoff type string_3: triggered event name string_4: Description of event string_5: return code, always "failure"	

Objects	Event name	Event meaning	Event Attributes Use	Event specific info
Common for Controlled Documents, CN and CR	dcm_cancel_ checkout_cd	Cancel checkout of controlled document	string_5: return code, always "failure"	
Common for Controlled Documents, CN and CR	dcm_cancel_ reject	Cancel of rejecting controlled document	string2: user who cancel the rejection of document string3: event description string_5: return code, always "failure"	Two entries are recorded, one for the document which is the parent and one for the child which is the signatory (user or group)
Auto-name Schema	dcm_change_ autoname_ scheme	Autoname scheme has changed	string_1: attribute name string_2: old value string_3: new value	
Business Application	dcm_change_ business_ application	Business application has changed	string_1: attribute name string_2: old value string_3: new value	
Documents, CN and CR	dcm_change_ cd_properties	Controlled document properties has changed	string_1: attribute name string_2: old value string_3: new value string_4: justification text (null for DCM 5.3) string_5: return code, “successful" or "failure"	

Objects	Event name	Event meaning	Event Attributes Use	Event specific info
Document class	dcm_change_ document_ class	Document class has changed	string_1: attribute name string_2: old value string_3: new value	
Relationship type	dcm_change_ relation_type	Relation type has changed	string_1: attribute name string_2: old value string_3: new value	
User List Rule	dcm_change_ user_list_rule	User List rule has changed	string_1: attribute name string_2: old value string_3: new value	
Common for Controlled Documents, CN and CR	dcm_ checkin_cd	Controlled document checked in	string_2: object ID string_5: return code, “successful" or "failure"	
Common for Controlled Documents, CN and CR	dcm_ checkout_cd	Controlled document checked out	string_5: return code, “successful" or "failure"	
Common for Controlled Documents, CN and CR	dcm_convert	Make document controlled	string_2: object name string_3: Document class name string_5: return code, “successful" or "failure"	

Objects	Event name	Event meaning	Event Attributes Use	Event specific info
Common for Controlled Documents	dcm_copy_cd	Controlled document copied	string_1: business application string_2: copied document object ID string_3: document class string_5: return code, “successful" or "failure"	
Controlled Document	dcm_create_cd	Controlled document created	String1: Business process name String3: Document class name string_5: return code, “successful" or "failure"	
CN	dcm_create_cn	Change notice creation	String1: Business process name String3: Document class name string_5: return code, “successful" or "failure"	

Objects	Event name	Event meaning	Event Attributes Use	Event specific info
CR	dcm_create_cr	Change request creation	String1: Business process name String3: Document class name string_5: return code, “successful" or "failure"	
Common for Controlled Documents, CN and CR	dcm_create_ relationship	Relationship between two documents was created	String2: child document name String3: relation type name string_5: return code, “successful" or "failure"	
Common for Controlled Documents, CN and CR	dcm_delegate_ signoff	Signoff task delegated to another user	string_2: delegated user name string_5: return code, “successful" or "failure"	
Common for Controlled Documents, CN and CR	dcm_delete_ all_tbrs	All TBRs related to the controlled document are deleted. Runs automatically when a controlled document is deleted.	string_2: number of deleted TBR notices string_5: return code, “successful" or "failure"	
Autoname Schema	dcm_delete_ autoname_ scheme	Autoname scheme deleted	string_1: autoname scheme name string_2: object type string_3: owner name	

Objects	Event name	Event meaning	Event Attributes Use	Event specific info
Business Application	dcm_delete_ business_ application	Business Application deleted	string_1: object name string_2: object type string_3: business application or document class string_4: owner name	
Common for Controlled Documents, CN and CR	dcm_delete_cd	Controlled document deleted	String1: business application name String3: document class name string_5: return code, “successful" or "failure"	
Document Class	dcm_delete_ document_ class	Document class deleted	String_1: object name String_2: object type String_3: business application or document class String_4: owner name	
Relationship Type	dcm_delete_ relation_type	Relation type deleted	String _4: relation type name	
User List Rule	dcm_delete_ user_list_rule	Deletion of Dynamic Auto Process (also known as User List Rule) object(s)	string _1: auto process object name string _2: auto process type string _3: Lifecycle state type string _4: owner name	

Objects	Event name	Event meaning	Event Attributes Use	Event specific info
Common for Controlled Documents, CN and CR	dcm_demote	Controlled document demoted to previous state	string_1, lifecycle name string_2, State to which object was demoted string_5: return code, “successful" or "failure"	
Common for Controlled Documents, CN and CR	dcm_edit_cd	Controlled document edited	N/A	
Common for Controlled Documents, CN and CR	dcm_ export_cd	Controlled document exported	N/A	
Controlled Document	dcm_ import_cd	Controlled document imported	string_2: original document name string_3: document class name string_4: justification text (null for DCM 5.3) string_5: return code, “successful" or "failure"	
Common for Controlled Documents, CN and CR	dcm_initiate_ signoff	Signoff task initiated	String_2: signoff initiator name string_5: return code, “successful" or "failure"	Two entries are recorded, one for the document which is the parent and one for the child which is the signatory (user or group)

Objects	Event name	Event meaning	Event Attributes Use	Event specific info
Common for Controlled Documents, CN and CR	dcm_promote	Controlled document promoted (also makes it temporarily effective)	string_1, lifecycle name string_2, State from which object was promoted string_5: return code, “successful" or "failure"	
Common for Controlled Documents, CN and CR	dcm_reject	Controlled document rejected to signoff	string_1, justification text string_2, rejected person name string_5: return code, “successful" or "failure"	Two entries are recorded, one for the document which is the parent and one for the child which is the signatory (user or group)
Common for Controlled Documents, CN and CR	dcm_remove_ relationship	Relationship between two documents is removed	string_2, parent or child object name string_3, relationship type string_5: return code, “successful" or "failure"	Two entries are recorded, one for the document which is the parent and one for document which is child
Common for Controlled Documents, CN and CR	dcm_remove_ signatory	Signatory removal event	string_2: removed signatory name string_3: document state at which an signatory is removed string_5: return code,	Two entries are recorded, one for the document which is the parent and one for the child which is the signatory (user or group)

Objects	Event name	Event meaning	Event Attributes Use	Event specific info
			“successful" or "failure"	
Common for Controlled Documents, CN and CR	dcm_resume	Resumes a controlled document from state it was suspended	string_1, controlled document type string_2, State from which object was suspended/ resumed string_5: return code, “successful" or "failure"	
Common for Controlled Documents, CN and CR	dcm_send_tbr	To Be Reviewed notice was sent	String_2: sender name String_4: notification text string_5: return code, “successful" or "failure"	
Common for Controlled Documents, CN and CR	dcm_set_ effective_date	Controlled document effective date is set	string_1: attribute name string_2: old value string_3: new value string_4: justification text (null for DCM 5.3) string_5: return code, “successful" or "failure"	

Objects	Event name	Event meaning	Event Attributes Use	Event specific info
Common for Controlled Documents, CN and CR	dcm_set_ expiration_ date	Controlled document expiration date is set	string_1: attribute name string_2: old value string_3: new value string_4: justification text (null for DCM 5.3) string_5: return code, “successful" or "failure"	
Common for Controlled Documents, CN and CR	dcm_signoff	Controlled document signed off	string_1: Justification text string_2: signatory/ document name string_5: return code, “successful" or "failure"	Two entries are recorded, one for the document which is the parent and one for the child which is the signatory (user or group)
Common for Controlled Documents, CN and CR	dcm_signoff_ tbr	TBR notice confirmed	string_1: signatory name string_2: signoff date string_3: document name string_4: justification text string_5: return code, “successful" or "failure"	

Objects	Event name	Event meaning	Event Attributes Use	Event specific info
Common for Controlled Documents, CN and CR	dcm_suspend	Controlled document suspended	string_1, lifecycle name string_2, State to which object was suspended/ resumed string_5: return code, "successful" or "failure"	
Common for Controlled Documents, CN and CR	dcm_tbr_ change_ permission	Grant a TBR performer READ permission	Not implemented yet	
Common for Controlled Documents, CN and CR	dcm_update_ review_date	Review date updated	string_1: attribute name string_2: old value string_3: new value string_4: justification text (null for DCM 5.3) string_5: return code, "successful" or "failure"	
Documents, CN and CR	dcm_view_cd	Controlled document viewed	N/A	

Objects	**Event name**	**Event meaning**	**Event Attributes Use**	**Event specific info**
Lifecycle Extension (dm_policy)	dcm_change_ lifecycle_ extension	Lifecycle extension changed	string_1: attribute name string_2: old value string_3: new value string_4:object name string_5: return code, “successful” or “failure”	
Workflow (dm_process)	dcm_start_ controlled_ workflow	Start controlled workflow for controlled document	string_2: document name string_5: workflow description	
Workflow (dm_process)	dcm_abort_ controlled_ workflow	Abort controlled workflow for controlled document	string_2: document name string_5: workflow description	
Workflow (dm_process)	dcm_pause_ controlled_ workflow	Pause controlled workflow for controlled document	string_2: document name string_5: workflow description	
Workflow (dm_process)	dcm_resume_ controlled_ workflow	Resume controlled workflow for controlled document	string_2: document name string_5: workflow description	
Workflow (dm_process)	dcm_delete_ controlled_ workflow	Delete controlled workflow for controlled document	string_2: document name string_5: workflow description	

About DCM Configuration Parameters

DCM configuration parameters define document classes and business rules, in the form of user list rules, for specific or multiple business applications. Predefined document classes are provided with DCM such as Change Notices and Change Requests. You can however choose to create your own.

You can associate a user list rule with an existing business application, document class, and lifecycle state.

The DCM node consists of the following nodes used to create DCM configuration parameters:

- Business Applications
- Document Classes
- Auto-naming Schemes
- User List Rules
- Relationship Types
- Lifecycle Extensions

The **Business Applications** node gives DCM administrators the ability to create or modify business applications.

The **Document Classes** node: gives DCM administrators the ability to create or modify document classes.

The **Auto-naming Schemes** node: gives DCM administrators the ability to create file naming schemes.

The **User List Rules** node: gives DCM administrators the ability to create user list rules.

The **Relationship Types** node: gives DCM administrators the ability to create relationship types, and to delete existing relationship types.

The **Lifecycle Extensions** node: gives DCM administrators the ability to create signatories for a document.

Additional details and procedures are provided throughout this guide for each of the DCM configuration parameters listed.

Note: Configuration parameters are associated such that document classes can be set to inherit parameter settings from the business application or to override parameter settings of the business application.

Configuration parameters, such as whether manual promotion is allowed, signoff/rejection codes, and justifications can be specified at each lifecycle state, or you can specify these parameters when creating users, groups, or roles. You can even specify a configuration that combines parameters set in a combination of lifecycle state, user, group, or role.

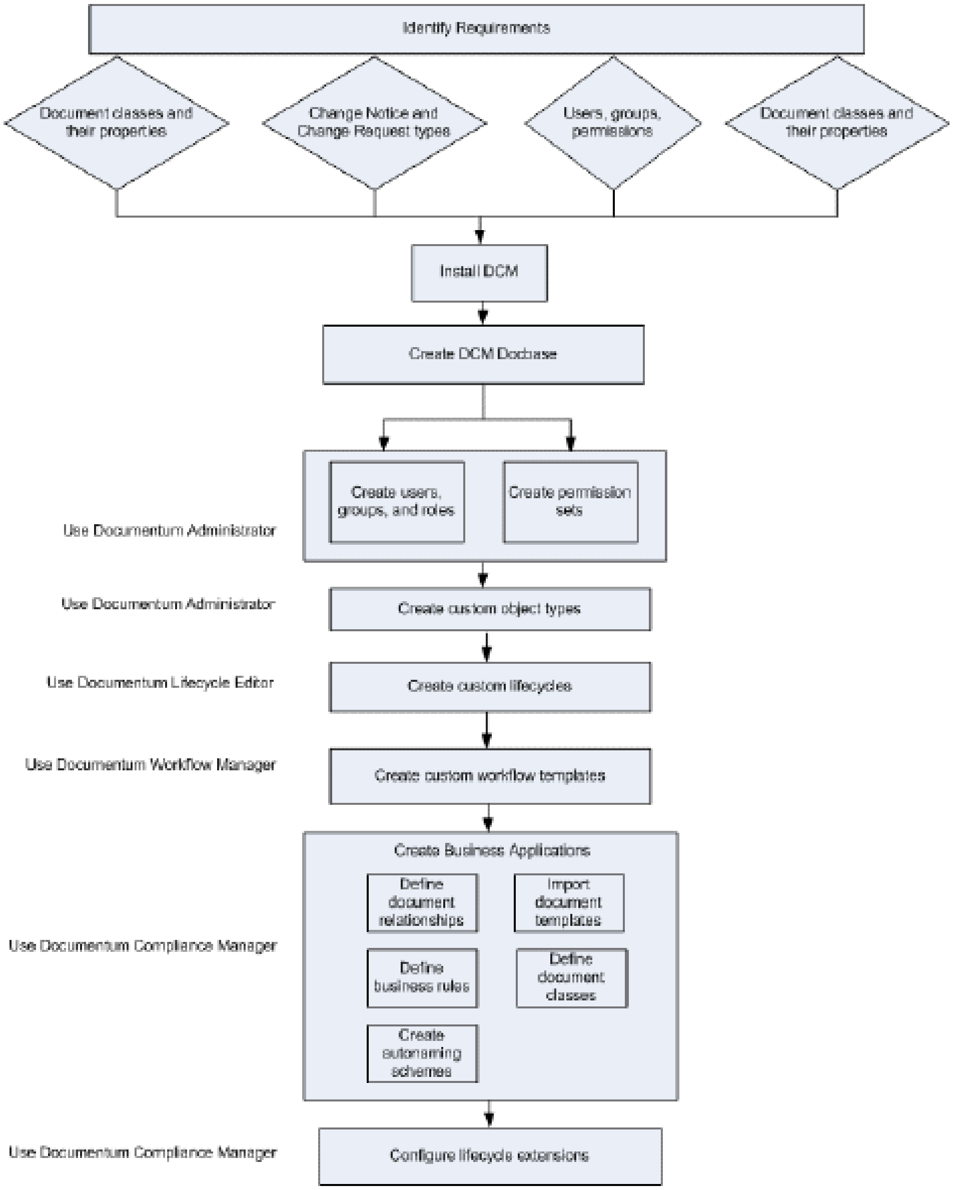
DCM allows overrides of the system-level configuration at the user, group or role level, and also at the lifecycle state level. The order of precedence for the configuration information is:

1. Look for configuration for a particular user and lifecycle state combination.
2. Look for configuration for a particular user (that is, a configuration tied directly to the user and not to a lifecycle state).
3. Look for a configuration that is tied directly to the lifecycle state.

Overview of DCM Configuration Tasks

The following figure describes the task flow for setting up a DCM system. The tasks described in this section are explained in detail in following chapters.

Figure 6. Task Sequence for Setting Up a DCM System


Identify Requirements
Document classes and their properties
Change Notice and Change Request types
Users, groups, permissions
Document classes and their properties
Install DCM
Create DCM Docbase
Use Documentum Administrator
Create users, groups, and roles
Create permission sets
Use Documentum Administrator
Create custom object types
Use Documentum Lifecycle Editor
Create custom lifecycles
Use Documentum Workflow Manager
Create custom workflow templates
Create Business Applications
Define document relationships
Import document templates
Use Documentum Compliance Manager
Define business rules
Define document classes
Create autonaming schemes
Use Documentum Compliance Manager
Configure lifecycle extensions

To configure DCM so that it enforces custom business processes, you typically perform the following tasks:

1. Perform planning and requirements tasks.

    a. Identify your business application requirements.

       Before creating and configuring DCM, determine what tasks you want DCM to manage, and map out your processes. What sorts of documents do you need to control? Who will be performing creation and signoff tasks? How are documents approved? What requirements must document fulfil before they can be approved for general use?

b. Identify document classes and their desired properties.

   Each document class is assigned a document owner/coordinator and certain parameters

c. Identify change notice (CN) and change request (CR) types and their desired properties

   If you choose not to use the predefined Change Notice and Change Request document classes provided with DCM, you must create document classes for the change request and change notice types you want DCM to manage.

d. Identify users, groups, permission sets, lifecycles, and workflows.

   You can create and use your own subgroups for each document type and lifecycle state, or for workflow reviews. For example, you may want to set up a signatories group for controlled documents when they reach the In Approval lifecycle state.

2. Set up the DCM repository:

   a. Install DCM.

      For instructions on how to how to install DCM, see the *Web Development Kit and Applications Installation Guide*.

   b. Create a DCM repository.

      For instructions on how to add a repository to an existing Documentum system, see the *Documentum Content Server Administrator's Guide*.

   c. Create users, roles, and groups.

      For more information on how to create users, roles, groups, and permissions sets, see the *Documentum Content Server Administrator's Guide*.

   d. Create DCM-specific permissions sets.

      DCM uses permissions sets and user/group permission levels to determine user access to DCM files and folders in the repository. The permissions sets associated with the documents and folders in the DCM cabinet/folder hierarchy determine which DCM documents a user can view and apply DCM functionality to. DCM permissions sets can be created or modified using the Documentum Administrator utility

   e. Create lifecycles.

      Determine the lifecycle states you want to use for each of your document types.

      For example, do you want a document to go through Draft, Review, Published, and Retired states?

You create lifecycles using Documentum Application Builder. For detailed instructions, see the *Documentum Application Builder User Guide*.

f. Create custom workflow templates.

   You create workflows using Documentum Workflow Manager or Business Process Manager (BPM). For detailed instructions, see the *Workflow Manager User Guide* or the *Business Process Manager User Guide*.

3. Configure DCM:

   a. Create or import custom document templates

      The procedure for importing a template is the same as importing any other object into your repository. It is important to note, however, that when importing templates into your repository, you must import them into the Templates cabinet and you must import one template for each object type that you want to make available in your repository.

   b. Optionally, define an autonaming scheme.

      For more information about autonaming schemes, see Chapter 10, Setting Up Autonaming Schemes.

   c. Define a business application.

      Document classes are grouped and managed in a business application. You can set the defaults and common properties for all the document classes included in a business application. For more information about business classes, see Chapter 5, Creating and Administering Business Applications

   d. Define document classes.

      For more information on defining document classes, see Chapter 6, Setting Up Document Classes and Document Relationships .

   e. Optionally, create change notice and change request document classes.

      DCM provides default change notice and change request document classes. You can modify these existing document classes, or create new document classes for your change notices and change requests. For information about setting up new change notice and change request document types, see .

   f. Edit lifecycle states to add DCM-specific extensions.

      For information specific to DCM's lifecycle extensions, see Chapter 8, Creating and Editing Controlled Document Lifecycles.

   g. Define document relationships.

      For more information on defining document relationships, see Chapter 6, Setting Up Document Classes and Document Relationships .

4. Customize DCM for your own needs.

   a. Create custom object types for your DCM application.

For more information on how to create types, see the *Content Server Administrator's Guide*.

b. Define business rules.

   For information about business rules, see Business Rules, page 101.

c. Put your company's logo and other images on the customized DCM screens.

   For more information about changing the appearance of DCM using branding and themes, see the *Web Development Kit and Client Applications Development Guide*.

Obtaining Access to the DCM Administration Node

Not all DCM users can see the DCM Administration node and its associated pages. After installing DCM, the installation owner can view the Administration pages.

A user that is assigned to the group "business application owner" and assigned with "none" for privileges and extended privileges and assigned with "Coordinator" client capability is able to access the User Management, Security, and DCM nodes under the Administration node.

Note: A user's membership in a group does not affect their ability to see the DCM Administration node. It is simply a best practice recommendation for classifying users in your repository.

Using Other Software Tools with DCM

To set up a DCM system, you will need to work with several other tools. Some of these are Documentum products and some are third-party products. Typically, these products include:

- Documentum Administrator

  a Web-based interface that lets you monitor, administer, configure, and maintain Documentum repositories throughout the enterprise (both local and federated) from any system running a Web browser. Documentum Administrator also provides easy access to the Documentum System Administration Tool suite and the administration methods.

- Headless Composer

  the principal tool for installing server applications using DAR files.

- Composer

  Documentum Composer provides tools to create and customize applications for Documentum Content Server.
- Workflow Manager and Buusiness Process Manager

  tools for developing workflow templates that can be accessed through Documentum Desktop, Documentum Developer Studio, and Webtop clients.
- Documentum Content Rendition Services

  help you manage PDF renditions of controlled documents.

To set up a custom DCM business application, the DCM administrator typically creates one or more repositories. For instructions on how to create a repository in an existing Documentum system, see the *Documentum Content Server Administrator's Guide*.

To create any required custom object types, methods, or jobs, the DCM administrator uses Documentum Administrator. For detailed information, see the *Documentum Administrator User's Guide*.

To create and maintain users, groups, and permission sets, the DCM administrator uses the DCM administration screens.

To create and maintain DARs, workflows and lifecycles, the DCM administrator must use Composer and Workflow Manager. DDS and Workflow Manager both require sys_admin privileges to use.

Related Documents

The information in this guide refers to concepts and procedures described in other manuals. These manuals are:

- *Documentum Content Server Administrator's Guide*
- *Documentum Content Server Fundamentals*
- *Documentum Content Server API Reference Manual*
- *Documentum Content Server Object Reference Manual*
- *Documentum Application Builder User Guide*
- *Documentum Administrator User Guide*
- *Web Development Kit and Client Applications Development Guide*
- *Workflow Manager User Guide*
- *Composer User Guide*

Using the Additional Accessibility Options

The accessibility mode provides an easier-to-read interface by replacing menus with full-page lists of available actions and by providing additional descriptive text on many options. You can select the accessibility mode when logging in. You can set the accessibility mode as your default mode in your **Preferences**.

The accessibility mode provides the following:

- Linear navigation. Links replace menus.
- All UI elements are tab enabled.
- All UI elements have ALT tags with the names, descriptions, and in some cases states of the elements. If your screen reader has the option to set which tags it reads, include title and ALT tags.
- Accessible HTML Help documentation
- Java-script & Java UI elements, such as the actions menu, display filters. Import screens, are replaced with HTML alternatives.

For more information on accessibility features, see Accessibility Features, page 43.

To use the accessibility mode:

1. In your Web browser, type in the DCM URL. You type the URL in the **Address** field for Internet Explorer or in the **Location** field for Netscape.
2. In the **Login Name** field, type your case-sensitive user name.
3. In the **Password** field, type your case-sensitive password.
4. In the **Repositories** field, select the repository.
5. Click **More Options**.
6. Select the **Additional Accessibility Options** checkbox.
7. Click **Login**.

Accessibility Features

The WDK (Web Development Kit) platform is a Web-based application that is accessed via an Internet browser. The following are known issues in the accessibility mode:

- Large numbers of frames. Screen readers with frame navigation features help in dealing with this problem.
- Drop-list form control causes immediate navigation. JAWS users can use ALT+up/down arrow to avoid immediate navigation.

- Refreshing the page in the browser always returns you to the top level of your home node.
- All standard components are accessible, but components developed or modified by a third party might not be accessible.

Keep in mind WDK is an application not a Web site.

Components of the DCM user interface:

- **Title bar**

  This is the first frame. The first element in this frame is a quick link to the work area. Because WDK is a Web-based screen, refreshing can cause the focus to jump out of the work area back to the top of the page. This link allows you to quickly return to the work area. The frame also contains a Documentum-specific search box, links to advanced searches, preferences, logout, and help. Documentum Help launches in a new browser window. The Documentum logo provides the build version. The toolbar and menu bar frames are minimized when accessible mode is on, as they don't facilitate linear navigation. When navigating frames using assistive technology, you might encounter the hidden frames as an extra navigational step between the browser tree and work area, but the frames will appear blank. If you encounter text in these frames, you might not have accessibility settings on.

- **Browser tree**

  The browser tree is hierarchical tree navigation, but because it appears in HTML it behaves different from a regular Windows or Java tree control. The frame includes a link to the work area, then a link to the selected node and a link to the bottom of the frame. The tree then contains a list of available repositories. Clicking the expand icon expands the node in the tree. Clicking the item name itself displays the repository login page, if you are not currently logged in to that repository. If you are within a repository, on a node and activate it, the contents of that node will load in the content frame. Currently selected nodes also have an adjacent link to the next node.

- **Work area**

  The work area has two main views: object list and action form. Examples of object lists include the Inbox, Subscriptions, and My Files. Examples of action forms include properties, preference, checkin and import pages. Both views include page titles and in most cases breadcrumbs.

- **Object lists**

  Objects lists are paginated to improve performance. The default number of items per page is 10, but this can be changed by using the **Show Items** field. Most object lists are sorted by name but can be sorted by other column headers, which appear as links with ALT tags. When a column is sorted, a tagged arrow appears next to the column.

  The next link after the column headers is **Global Actions**. This link loads a page from which you can invoke all object-independent actions (for example, create new folder, new user etc.). Within the list the first link on a row is the actions link. The actions link loads a page with all available actions for that object. In order to check out or edit the file, you must use the actions page. Clicking the name of the object opens its contents—for a folder, the list of objects within the folder; for a file, the file's contents in View mode only. The rows for a given object contain the most pertinent metadata for that object and can be navigated using your screen reader table navigation. More information can be found on the object's **Properties** page. Most object lists have a link to the **Properties** page.

- **Actions page**

  An actions page loads when you invoke the actions link. All the available actions appear as links. Setting focus on the link and invoking it causes the action to load. In certain cases the action is a navigation to modal page—for example, viewing a list of translations. If the navigation option is not available, use the breadcrumb link at the top of the page.

- **Action forms**

  The properties page is the most prevalent example of an actions form. Most contain a series of tabbed pages, shown as vertical tabs. These are described as pages in the tags, so that users don't confuse them with Tab controls that they are already familiar with. The currently selected tab has a selected tag.

  Most properties pages contain a combination of edit boxes, check boxes and radio buttons. All of these are explicitly labeled. At the bottom of all action forms are buttons with ALT tags that allow you to perform the action, cancel, or move forward or backward. Cancel typically brings you back to the object list you were on. In some cases it might be necessary to refresh the page, which will return you to the node stated in your preferences.

- **Choosers**

  Choosers are used throughout the application to select objects, such as files, or users. The chooser requires objects in the global list to be selected and added to the selected list.

- **Message bar**

  This contains system messages for your current session. If an action you are trying is not working, it might be necessary to set focus to the message bar, or to navigate to the adjacent Status Bar frame. The first element lets you view all messages for a given session.

Chapter 2

Most-Recent Label

In order to identify the most recent version in a version tree, whether an Effective version exists or not, DCM 6.5 introduces the "Most Recent" symbolic label. Using the "Most Recent" version label, contributors and coordinators can determine which version they should work on. DCM administrators can specify "Most Recent" as the filter for the packages associated with controlled workflow or uncontrolled workflow activities. This alleviates the limitation imposed by using "CURRENT" on an Effective document.

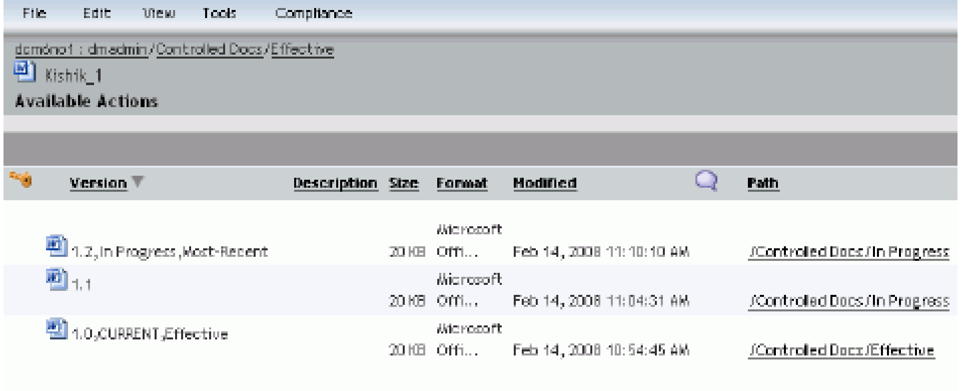
By default, a WebTop based application, including DCM, always displays the CURRENT version of a version tree. In the first lifecycle iteration, users always see the most recent version. Once the Effective version is introduced, users always see the Effective version even if the lifecycle is re-started by re-versioning.

Most-Recent Label Top Level Functionality

Currently, all DCM 5.3.x and earlier controlled documents receive three version labels upon creation: numeric version label like "0.1", current version label "CURRENT", and symbolic version which reflects the document lifecycle states such as "In Progress" and "Review" for example.

All controlled documents created by DCM 6.5 and later in addition to the three labels mentioned above should automatically receive the fourth "Most-Recent" label upon creation. Moving through document lifecycle, a controlled document labelled the "Most-Recent" version will remain with the document until it is re-versioned. The "Most-Recent" version is moved/reassigned to the new version of controlled document upon check-in while the previous "Most-Recent" label is nullified.

Figure 7. Checked in effective version



Version	Description	Size	Format	Modified	Path
1.2,In Progress,Most-Recent		20 KB	Microsoft Offi...	Feb 14, 2008 11:10:10 AM	/Controlled Docs/In Progress
1.1		20 KB	Microsoft Offi...	Feb 14, 2008 11:04:31 AM	/Controlled Docs/In Progress
1.0,CURRENT,Effective		20 KB	Microsoft Offi...	Feb 14, 2008 10:54:45 AM	/Controlled Docs/Effective

By having the Most-Recent label, a user can simply determine the latest work-in-progress (if it has the Most-Recent label) and the effective version (with the CURRENT label). Users have a choice when adding an attachment to a DCM controlled workflow to select the CURRENT version label, Optional version label, or specify their own. The CURRENT label cannot be used as a unique identifier for the Most-Recent version when an Effective version exists within the version tree. Specifying the Optional label may also cause problems when a user is authoring a controlled document within a regular workflow. Problems with workflows used to support various lifecycles are now resolved with the introduction of the Most-Recent version label. The Most-Recent version label provides unique identification for the work in-progress version.

Chapter 3

Setting Up Automated Auditing and Electronic Signature Processes

Note: It is recommended you avoid having Document Transformation Services (DTS) configured for more than one lifecycle state. The Signature Page of an existing PDF rendition is lost as a new rendition is created when it enters another state that also has DTS configured. Default lifecycles provided with DCM, are configured to provide a PDF rendition in the Review state. No PDF renditions are generated if DTS is not configured.

DCM uses PDF Stamping Services (PSS) electronic signature and audit. This chapter explains how auditing and electronic signatures work and how to set up auditing and electronic signatures for specific DCM events. The following topics are discussed:

- How Electronic Signatures Work with DCM, page 49
- Adding DCM Audit Events, page 56
- Enabling Electronic Signature for DCM Signoff and Reject Events, page 59

How Electronic Signatures Work with DCM

Many business processes have signature requirements for one or more steps in the process. Similarly, some lifecycle states may require a signature before an object can move to the next state. For example, a budget request may need signoff before the money is disbursed. Users may be required to sign SOPs (standard operating procedures) to indicate that they have read the procedures. Or a document may require signoff before the document is published on a Web site.

Content Server supports signature requirements with three options:

- Electronic signature
- Digital signature
- Simple signoffs

Electronic signatures are generated and managed by PDF Stamping Services (PSS). The feature is supported by two API methods: Addesignature and Verifyesignature. Use this option if you require a rigorous signature implementation to meet regulatory requirements.

Digital signatures are electronic signatures in formats such as PDKS #7, XML signature, or PDF signature. Digital signatures are generated by PSS when an Adddigsignature method is executed. Use this option if you want to implement strict signature support in a client application. For more information, refer to Digital Signatures, page 55.

Simple signoffs are the least rigorous way to supply an electronic signature. Simple signoffs are implemented using the Signoff method. This method authenticates a user signing off a document and creates an audit trail entry for the dm_signoff event. Signoff Method Usage, page 56, describes this option in detail.

Electronic Signatures

Electronic signatures are generated by PSS when an application or user issues an Addesignature method. Using Addesignature is a rigorous way to fulfill a signature requirement. Signatures generated by Addesignature are recorded in a formal signature page and added to the content of the signed object. The Addesignature method is audited automatically, and the resulting audit trail entry is itself signed by PSS. The auditing feature cannot be turned off. If an object requires multiple signatures, before allowing the addition of a signature, PSS verifies the preceding signature. PSS also authenticates the user signing the object.

All the work of generating the signature page and handling the content is performed by PSS. The client application is only responsible for recognizing the signature event and issuing the Addesignature method. A typical sequence of operations in an application using the feature is:

1. A signature event occurs and is recognized by the application as a signature event.

   A signature event is an event that requires an electronic signature on the object that participated in the event. For example, a document checkin or lifecycle promotion might be a signature event.

2. In response, the application asks the user to enter a password and, optionally, choose or enter a justification for the signature.

3. After the user enters a justification, the application can call the Createaudit method to create an audit trail entry for the event.

   This step is optional, but auditing the event that triggered the signature is common.

4. The application calls Addesignature to generate the electronic signature.

After Addesignature is called, PSS performs all the operations required to generate the signature page, create the audit trail entries, and store the signature page in the repository with the object. You can add multiple signatures to any particular version of a document. The maximum number of allowed signatures on a document version is configurable.

Electronic signatures require a template signature page and a method (stored in a dm_method object) to generate signature pages using the template. Documentum provides a default signature page template and signature generation method that can be used on documents in PDF format or documents that have a PDF rendition. (The default template and method are described in detail in The Default Signature Page Template and Signature Method, page 52.) You can customize the electronic signature support in a variety of ways. For example, you can customize the default template signature page, create your own template signature page, or provide a custom signature creation method for use with a custom template.

What Addesignature Does

When an application or user issues an Addesignature method, PSS performs the following operations:

1. Authenticates the user and verifies that the user has at least Relate permission on the document to be signed.

   If a user name is passed in the Addesignature method arguments, that user must be the same as the session user issuing the Addesignature method.

2. Verifies that the document is not checked out.

   A checked out document cannot be signed by Addesignature.

3. Verifies that the pre_signature hash argument, if any, in the method, matches a hash of the content in the repository.

4. If the content has been previously signed, the server

   5. Retrieves all the audit trail entries for the previous dm_addesignature events on this content.

   6. Verifies that the most recent audit trail entry is signed (by PSS) and that the signature is valid

   7. Verifies that the entries have consecutive signature numbers

   8. Verifies that the hash in the audit trail entry matches the hash of the document content

9. Copies the content to be signed to a temporary directory location and calls the signature creation method. The signature creation method:

   10. Generates the signature page using the signature page template and adds the page to the content.

   11. Replaces the content in the temporary location with the signed content.

12. If the signature creation method returns successfully, the server replaces the original content in the repository with the signed copy.

If the signature is the first signature applied to that particular version of the document, PSS appends the original, unsigned content to the document as a rendition with the page modifier set to dm_sig_source.

13. Creates the audit trail entry recording the dm_addesignature event.

    The entry also includes a hash of the newly signed content.

You can trace the operations of Addesignature and the called signature creation method.

The Default Signature Page Template and Signature Method

Documentum provides a default signature page template and a default signature creation method with PSS so that you can use the electronic signature feature with no additional configuration. The only requirement to use the default functionality is that documents to be signed must be in PDF format or have a PDF rendition associated with their first primary content page.

Default Signature Page Template

The default signature page template is a PDF document generated from a Word document. Both the PDF template and the source Word document are installed for PSS. They are installed in %DM_HOME%\bin ($DM_HOME/bin). The PDF file is named sigpage.pdf and the Word file is named sigpage.doc.

In the repository, the Word document that is the source of the PDF template is an object of type dm_esign_template. It is named Default Signature Page Template and is stored in

```
System/Applications/pss
```

The PDF template document is stored as a rendition of the word document. The page modifier for the PDF rendition is dm_sig_template.

The default template allows up to six signatures on each version of a document signed using that template.

How Content is Handled by Default

If you are using the default signature creation method, the content to be signed must be in PDF format. The content can be the first primary content page of the document or it can be a rendition of the first content page.

When the method creates the signature page, it appends or prepends the signature page to the PDF content. (Whether the signature page is added at the front or back of the content to be signed is configurable.) After the method completes successfully, PSS adds the content to the document:

- If the signature is the first signature on that document version, the server replaces the original PDF content with the signed content and appends the original PDF content to the document as a rendition with the page modifier dm_sig_source.
- If the signature is a subsequent addition, the server simply replaces the previously signed PDF content with the newly signed content.

The Audit Trail Entries

PSS automatically creates an audit trail entry each time an Addesignature method is successfully executed. The entry records information about the object being signed, including its name, object ID, version label, and object type. The ID of the session in which it was signed is also recorded. (This can be used in connection with the information in the dm_connect event for the session to determine what machine was used when the object was signed.)

PSS uses the generic string attributes in the audit trail entry to record information about the signature. The following table lists the use of those attributes for a dm_addesignature event.

Table 2. Generic String Attribute Use for Dm_addesignature Events

Attribute	Stores
string_1	Name of the user who signed the object
string_2	The justification for the signature
string_3	The signature's number, the name of the method used to generate the signature, and a hash of the content prior to signing. The hash value is the value provided in the pre_signatureHash argument of the Addesignature method. The information is formatted in the following manner: `sig_number/method_name/pre_signature hash argument`

Attribute	Stores
string_4	Hash of the primary content page 0. The information also records the hash algorithm and the format of the content. The information is formatted in the following manner: `hash_algorithm/format_name/hash`
string_5	Hash of the signed content. The information also records the hash algorithm and the format of the content. The information is formatted in the following manner: `hash_algorithm/format_name/hash` If the signed content was added to the document as primary content, then value in string_5 is the same as the string_4 value.

What You Can Customize

If you are using the default electronic signature functionality, signing content in PDF format, you can customize the signature page template. You can add information to the signature page, remove information, or just change its look by changing the arrangement, size, and font of the elements on the page. You can also change whether the signature creation method adds the signature page at the front or back of the content to be signed.

If you want to embed a signature in content that is not in PDF format, you must use a custom signature creation method. You may also create a custom signature page template for use by the custom signature creation method; however, using a template is not required.

Verifying Signatures

Electronic signatures added by Addesignature are verified by the Verifyesignature method. The method finds the audit trail entry that records the latest dm_addesignature event for the document and performs the following checks:

- Calls the Verifyaudit method to verify the PSS signature on the audit trail entry
- Checks that the hash values of the source content and signed content stored in the audit trail entry match those of the source and signed content in the repository
- Checks that the signatures on the document are consecutively numbered.

Only the current signature is verified. If the current signature is valid, previous signatures are guaranteed to be valid.

General Usage Notes

Here are some general notes about working with electronically signed documents:

- Users can modify a signed document's attributes without invalidating the signatures.
- If the signed document was created on a Macintosh machine, modifying the resource fork does not invalidate the signatures.
- Addesignature cannot be executed against an object that is checked out of the repository.
- Checking out a signed document and then checking it in as the same version invalidates the signatures on that version and prohibits subsequent signings.
- If you dump and load a signed document, the signatures are not valid in the target repository.
- If you replicate a signed document, executions of Addesignature or Verifyesignature against the replica will act on the source document.
- Using Addesignature to sign a document requires at least Relate permission on the document.
- Using Verifyesignature to verify a signature requires at least Browse permission on the signed document.

Digital Signatures

Digital signatures are electronic signatures in formats such as PKCS #7, XML Signature, or PDF Signature. Signatures in these formats are implemented and managed by the client application. The application is responsible for ensuring that users provide the signature and for storing the signature in the repository. The signature can be stored as primary content or renditions. For example, if the application is implementing digital signatures based on Microsoft Office XP, the signatures are typically embedded in the content files and the files are stored in the repository as a primary content files for the documents. If Adobe PDF signatures are used, the signature is also embedded in the content file, but the file is typically stored as a rendition of the document, rather than primary content.

Note: If you wish assistance in creating, implementing, or debugging a digital signature implementation in an application, you must contact Documentum Professional Services or Documentum Developer Support.

PSS supports digital signatures with an attribute on SysObjects and the Adddigsignature method. The attribute is a Boolean attribute called a_is_signed. The Adddigsignature method generates an audit trail entry recording the signing. The event name for the audit trail entry is dm_adddigsignature. The information in the entry records who signed the document, when it was signed, and a reason for signing, if one was provided.

An application using digital signatures typically implements the following steps for the signatures:

1. Obtain the user's signature.
2. Extract the signature from the document and verity it.
3. If the verification succeeds, set the a_is_signed attribute to T.
4. Check the document in to the repository.
5. Issue the Adddigsignature method to generate the audit trail entry.

It is possible to require PSS to sign the generated audit trail entries. Because the Adddigsignature method is audited by default, there is no explicit registry object for the event. However, if you want PSS to sign audit trail entries for dm_adddigsignature events, you can issue an explicit Audit method for the event, setting the sign_event argument to TRUE in the Audit method.

Signoff Method Usage

Simple signoffs are the least rigorous way to enforce a signature requirement. A simple signoff is useful in situations in which the signoff requirement is not rigorous. For example, you may want to use a simple signoff when team members are required to sign a proposal before the proposal is sent to upper management.

Simple signoffs are implemented using a Signoff method. The method accepts a user authentication name and password as arguments. When the method is executed, PSS calls a signature validation program to authenticate the user. If authentication succeeds, PSS generates an audit trail entry recording the signoff. The entry records what was signed, who signed it, and some information about the context of the signing. Using Signoff does not generate an actual electronic signature. The audit trail entry is the only record of the signoff.

You can use a simple signoff on any SysObject or sysobject subtype. A user must have at least Read permission on an object to perform a simple signoff on the object.

You can customize a simple signoff by creating a custom signature validation program.

Adding DCM Audit Events

DCM uses the TCS audit trail, which stores entries in the repository where the event occurred. This audit trail contains the following information:

- the name of the application you were using to make changes

- document name
- repository version of the audited document (for example, 1.4)
- the object ID

  You can derive the object creation date and the modification date from the object ID, if necessary
- the application event resulting in the audit trail entry being created
- A GMT timestamp for when the action occurred
- the user who performed the action
- a justification for the action
- location where the action occurred (for example, the name of the host computer where PSS and the repository are installed)

The DCM administrator must perform the following tasks to enable auditing, authentication and electronic signature:

- Register DCM application events to the data dictionary

  Application events are user-defined events that are recognized and audited by client applications. For example, a user opening a particular dialog can be an audited application event. Audited application events are configured using the Documentum Server Manager.
- Register DCM audit events

  Audit events are those actions performed on documents that are recorded in the Documentum audit trail object (dm_audittrail).
- Enable electronic signature for DCM Signoff and Reject events

  Electronic signatures are generated and managed by PSS.

The following sections describe how to perform each of these tasks.

Note: You may configure your system to use the PDFsign server for e-signatures.

Registering DCM Application Events to the Data Dictionary

You use Documentum Server Manager to register DCM application events to the Documentum data dictionary. To perform the following procedure, you must connect to the repository as the an install owner.

To register DCM application events:

1. Login to the Content Server.
2. Launch Documentum Server Manager.

To do this, choose **Start > Programs > Documentum > Documentum Server Manager** from the Windows task bar.

3. On the repository tab, select the required repository and click the **IDQL** button.
4. Enter the Content Server install owner name and password.
5. Type the following DQL statement: in the console window:

```
alter type dm_sysobject
   append auditable_appevents=’dcm_export_cd’,
   append auditable_appevents=’dcm_cancel_reject’,
   append auditable_appevents=’dcm_cancel_checkout_cd’,
   append auditable_appevents=’dcm_remove_relationship’,
   append auditable_appevents=’dcm_create_relationship’,
   append auditable_appevents=’dcm_change_cd_properties’,
   append auditable_appevents=’dcm_change_review_period’,
   append auditable_appevents=’dcm_signoff’,
   append auditable_appevents=’dcm_reject’,
   append auditable_appevents=’dcm_convert’,
   append auditable_appevents=’dcm_import_cd’,
   append auditable_appevents=’dcm_create_cd’,
   append auditable_appevents=’dcm_add_approver’,
   append auditable_appevents=’dcm_remove_approver’,
   append auditable_appevents=’dcm_attach_lifecycle’,
   append auditable_appevents=’dcm_promote’,
   append auditable_appevents=’dcm_demote’,
   append auditable_appevents=’dcm_checkin_cd’,
   append auditable_appevents=’dcm_checkout_cd’,
   append auditable_appevents=’dcm_change_policy’,
   append auditable_appevents=’dcm_change_business_application’,
   append auditable_appevents=’dcm_change_document_class’,
   append auditable_appevents=’dcm_change_autoname_scheme’,
   append auditable_appevents=’dcm_change_relation_type’,
   append auditable_appevents=’dcm_change_auto_process’,
   append auditable_appevents=’dcm_start_control_workflow’,
   append auditable_appevents=’dcm_delete_cd’,
   append auditable_appevents=’dcm_create_cr’,
   append auditable_appevents=’dcm_send_tbr’,
   append auditable_appevents=’dcm_create_cn’
   append auditable_appevents=’dcm_set_effective_date’
   append auditable_appevents=’dcm_add_approver’
   append auditable_appevents=’dcm_remove_approver’
   append auditable_appevents=’dcm_delegate_approver’
   append auditable_appevents=’dcm_cancel_checkout_cd’
   append auditable_appevents=’dcm_view_cd’
   append auditable_appevents=’dcm_edit_cd’
   append auditable_appevents=’dcm_update_review_date’
   append auditable_appevents=’dcm_signoff_tbr’
   append auditable_appevents=’dcm_delete_business_application’
   append auditable_appevents=’dcm_delete_document_class’
   append auditable_appevents=’dcm_delete_autoname_scheme’
   append auditable_appevents=’dcm_change_user_list_rule’
   append auditable_appevents=’dcm_delete_relation_type’
   append auditable_appevents=’dcm_set_expiry_date’

   PUBLISH
   Go
```

6. If the "object_altered" returns a value of '1', this indicates that the DCM events were added successfully.

Note: The application events are installed when the DAR files are installed using Composer.

Registering DCM Audit Events

You use Documentum Administrator to select which DCM events are audited. To register audit events, you must connect to Documentum Administrator as a superuser with the Config, View, and Purge Audit extended privileges.

To register DCM audit events:

1. Login to Documentum Administrator (DA) as a super user.
2. Ensure the user has Config, View, and Purge Audit extended privileges.
3. Click on the **Manage Auditing By Object Type** link.
4. Click on the **Audit Management** node in the Navigation column.

   Note: If this link is not available, then the user's extended privilege has not been set up correctly. To fix it, login as another super user, and modify the initial super user's extended privilege. Then reconnect to DA as the initial super user.
5. Choose type dm_sysobject then click **OK**.
6. Click **Add Audit.**
7. Type `dm_dcm` in the **Application Code** field.
8. Click **Add Event**.
9. Choose the DCM application events that you want to audit, and click OK.
10. Click **OK** on all pages until you get back to the **Audit Management** Web page.

Enabling Electronic Signature for DCM Signoff and Reject Events

Electronic signatures are generated and managed by PDF Stamping Services (PSS), native DCM software. Third party software for electronic signatures and watermarks using Trusted Content Services (TCS) is no longer needed with DCM 6.5. Procedures specific to TCS contained in this section in previous releases used to verify whether the electronic signature functionality was already enabled for signoff and reject events, and the procedure provided to enable, if not already enabled, are no longer required.

Chapter 4

Setting Up Users, Groups, and Roles

This chapter describes how to configure users, groups, and roles for DCM. It provides information and instructions for creating users and groups, defining user roles, and dynamically creating user lists by defining user list rules. This chapter also provides information on modifying users, groups, and roles once created.

Topics discussed in this chapter include:

Users and Roles in DCM

In addition to the system administrator, also known as the installation owner, DCM installation automatically creates the following three roles:

- business application owner
- document class owner
- functional area supervisor

These three predefined roles are all DCM administrators. Users with these roles can see the relevant DCM administration screens when they connect to DCM. These predefined role names cannot be changed. However, you can add or remove users from these roles. The DCM administrator roles are fully described in the following sections.

You can also define additional roles. You create or modify roles by editing the DCM XML configuration files. For example, you can change the scope of a role to expand or restrict capabilities. You can also group functionality in a single role. For more information about customizing DCM, see the *DCM Development Guide*.

There are several typical roles included with the sample DCM application installed with the product:

- signatory
- author
- consumer
- print supervisor

For more information about how roles are handled in the Documentum 5 system, see the *Security Services* chapter in *Content Server Fundamentals*.

About the Installation Owner

The installation owner is the person in charge of installing, configuring, and maintaining the DCM system. These activities require technical knowledge as well as security access to the network, server, database, Documentum Administrator, Documentum Developer Studio, and user information.

The installation owner typically performs the following tasks:

- installing or upgrading DCM
- creating DCM repositories
- migrating from earlier versions of DCM
- creating and modifying lifecycles
- creating and modifying workflows
- granting Audit permissions to view, configure, or purge, create and modify object types and properties
- creating and modifying permissions sets
- creating and modifying users, groups, and roles; assigning users to business application roles
- creating and modifying cabinets and folders
- creating and modifying server methods
- creating and modifying server jobs

- maintaining the system
- enabling or disabling audit, signature, and authentication events
- running DMCL, WDK, and Webtop traces
- generating reports

Do not perform actions on DCM-controlled repository objects if you are connected to DCM as an installation owner or other system superuser. These actions include:

- promoting a document to the next lifecycle state
- suspending or resuming workflows
- checkin or checkout

To perform these actions, log out from the superuser account and log in again as a document class owner, business application owner, or coordinator with the correct privileges.

About the Business Application Owner Role

The business application owner is in charge of one or more business applications, such as Master Batch Records, Change Management System, Manufacturing Procedures, New Drug Submission, Safety Policies, Government Policy Creation, or Contracts Management System.

The business application owner creates document classes that apply to a particular business application.

The business application owner also typically performs the following tasks:

- creating lifecycle templates
- creating workflow templates
- bulk-loading documents and properties into the repository
- enabling or disabling audit, signature, and authentication events
- creating and modifying permissions sets
- creating and modifying users, roles, and groups
- creating and modifying cabinets and folders
- creating and maintaining business applications
- creating and maintaining document classes
- assigning a document class owner to a document class
- assigning and removing signatories and distribution lists to and from business applications
- run DMCL, WDK, and Webtop traces
- qualifying business application configurations

- generating reports
- performing the role of Document Class Owner

Users added to this role should have the following user privileges and client capability:

- **Privileges**: Superuser or System Administrator
- **Client Capability**: System Administrator

About the Document Class Owner Role

The document class owner is in charge of one or more document classes, such as SOPs or QA Manuals.

The document class owner is responsible for configuring one or more document classes that have been assigned by a business application owner. For example, a manufacturing change management system may require QA, QC, Manufacturing, Labeling, Design, Purchasing, Inventory, and Change Notice document classes.

The document class owner also typically performs the following tasks:

- configuring document classes;
- assigning and removing signatories and distribution lists to and from document classes;
- creating and modifying cabinets and folders;
- performing any author tasks

Users added to this role should have the following user privileges and client capability:

- **Privileges**: Superuser or System Administrator
- **Client Capability**: System Administrator

About the Functional Area Supervisor Role

Functional area supervisors are concerned with an area of the business or a particular operation. They organize lists of users around functional areas (that is, what users are working on) rather than the documents they must use. Functional area supervisors typically maintain lists of users for a functional area of the organization, as well as maintaining business rules. Documents are assigned to members of the business rules by matching the attributes of the functional area to attributes on the documents themselves.

Users added to this role should have the following user privileges and client capability:

- **Privileges**: Superuser or System Administrator
- **Client Capability**: System Administrator

About Authors and Contributors in the DCM System

Authors and contributors are responsible for creating, editing, signing documents electronically, and responding to notification on changes.

Each document class or business application defines the authors for documents created using that document class. Defining a user as an author for a document class does *not* override the permission sets that control a user's access to documents, folders, and cabinets in a Documentum system.

For example, to edit a document, a user must have at least Write permission. However, if you are not defined as an author on the Authors/Contributors tab of the document class properties page, then you cannot create or import a controlled document regardless of your permissions. You can optionally select users, groups, or roles as authors for a document class.

Authors typically perform the following tasks:

- creating a controlled document
- importing a controlled document class
- checking out, checking in, or canceling checkout of a controlled document
- exporting controlled documents

  Note: Exporting configurations will fail if the DFC_DATA path is not available to DCM.

- attaching or detaching children documents to or from a parent document
- assigning or removing required signatories based on document class rules
- submitting To-Be-Read notices
- halting and restarting a workflow
- reviewing, signing off, rejecting, or re-initiating signoff for document class instances
- performing any consumer tasks

A contributor is any other user collaborating with the document owner. Additionally, DCM requires contributors to be defined on the Authors/Contributors tab of the document class properties page.

Caution: When you create users for an author or contributor role, ensure that you give the selected users and groups sufficient user privileges in the repository. Otherwise, these users will not have sufficient privileges to create or delete DCM relationship types.

About the Coordinator Role

Document class coordinators have the same responsibilities as document authors in demoting, suspending, or re-initiating signoff on documents.

Each document class or business application defines the coordinators for documents created using that document class. Document class coordinators can be defined individually when the document class is created, or the list of coordinators specified in the business application can be inherited.

Coordinators can perform the following tasks:

- Initiate Signoff
- Respond to promotion problem notifications
- Conduct periodic reviews and update last review date
- Demote a document
- Suspend a document
- Re-initiate signoff of a document

About the Signatory Role

A signatory is an optional way to designate who is responsible for signing off or rejecting a change, a request, or purchase. The signatory role is intended for managers responsible for business operations. The type and/or cost for the change, request, or purchase usually determine the level of signoff required.

Signatories for a document are assigned on the **Lifecycle Extensions** screen, on the **Signatories** tab.

Users defined as signatories must have at least Read permission to read and signoff the documents for which they are signatories. The Documentum Content Server requires that users have at least Relate permission to electronically sign off documents.

A signatory can perform the following tasks:

- reviewing controlled documents
- signing off or rejecting controlled documents

About the Consumer Role

The consumer is an optional way to designate who is the content consumer. Consumers typically view rather create content. For example, QA, QC, shop operators, trainers, and supervisors are examples of the types of users who are typically assigned a consumer role for a manufacturing SOP document.

DCM consumers are any user who has Read permission on a document. Consumers are not formally defined in the DCM system, nor are they assigned in any of the DCM screens. The permissions set assigned to a document determines whether a user can view it.

The consumer can perform the following tasks: viewing Effective documents; searching for Effective documents; generating reports; acting on To Be Read notifications.

About the Print Coordinator Role

The print Coordinator Role is responsible for generating controlled copy prints of controlled documents through the PDFAqua controlled printing interface. The role is not optional. Only members of this role are given access to the controlled printing functions in the Aqua menu in DCM.

Default Permissions Assigned to Users and Roles

This section describes how permission sets and security work in a repository, then describes the default permission levels assigned to DCM roles and objects.

What is a Permission Set?

A *permission set* determines who can access a particular item in the repository. Your access to the folders, documents and other items in a repository is determined by the permission sets that are assigned to those folders, documents and other items. Each item in the repository has an associated permission set, determining who can access the item and what actions each user with access can perform.

A permission set lists the users and user groups who have access and lists the level of access each user and user group has.

A user or group listed in a permission set is assigned one of seven access levels. Each access level includes all the permissions of the preceding levels:

- None: No access to the item.
- Browse: Users can view the item's properties (but not its content).

- Read: Users can view the item's content.
- Relate: Users can annotate the item.
- Version: Users can modify and check in new versions of the item.
- Write: Users can modify and check in the item as the same version.
- Delete: Users can delete items.

DCM Permissions

Editing a Permission set from the Security node, **Administration>Security**, will result in expected behavior. Attempting to change the permission set otherwise may result in unexpected behavior changing the specified ACL inadvertently. For example, editing Permissions from the folder level, **Cabinets>any_cabinet>any_folder**, results in unpredictable behavior replacing the specified ACL unexpectedly with one that is different.

The permission levels are:

- None: 1
- Browse: 2
- Read: 3
- Relate: 4
- Version: 5
- Write: 6
- Delete: 7

Table 3. DCM Approved Documents Permission Set

Role or User Name	Default Permission Level
administrator	7
signatory	4
author	7
business application owner	6
consumer	4
document class owner	6
functional area supervisor	6
dm_owner	7
dm_world	1

Table 4. DCM Approved Folder Permission Set

Role or User Name	Default Permission Level
administrator	3
signatory	3
author	6
business application owner	6
consumer	3
document class owner	6
functional area supervisor	6
dm_owner	6
dm_world	1

Table 5. DCM Suspended Folder Permission Set

Role or User Name	Default Permission Level
administrator	3
signatory	3
author	3
business application owner	3
consumer	1
document class owner	3
functional area supervisor	3
dm_owner	6
dm_world	1

Table 6. DCM Change Management Folders Permission Set

Role or User Name	Default Permission Level
administrator	6
signatory	3
author	6
business application owner	3
consumer	3
document class owner	3

Role or User Name	Default Permission Level
functional area supervisor	3
dm_owner	6
dm_world	1

Table 7. DCM Change Notice Closes Permission Set

Role or User Name	Default Permission Level
administrator	6
signatory	5
author	6
business application owner	5
consumer	3
document class owner	5
functional area supervisor	5
dm_owner	6
dm_world	1

Table 8. DCM Change Notice Documents Permission Set

Role or User Name	Default Permission Level
administrator	6
signatory	5
author	6
business application owner	5
consumer	5
document class owner	5
functional area supervisor	5
dm_owner	6
dm_world	1

Table 9. DCM Change Request Closes Permission Set

Role or User Name	Default Permission Level
administrator	6

Role or User Name	Default Permission Level
signatory	5
author	6
business application owner	5
consumer	3
document class owner	5
functional area supervisor	5
dm_owner	6
dm_world	1

Table 10. DCM Change Request Documents Permission Set

Role or User Name	Default Permission Level
administrator	6
signatory	5
author	6
business application owner	5
consumer	5
document class owner	5
functional area supervisor	5
dm_owner	6
dm_world	1

Table 11. DCM Effective Documents Permission Set

Role or User Name	Default Permission Level
administrator	6
signatory	5
author	5
business application owner	5
consumer	3
document class owner	5
functional area supervisor	5

Role or User Name	Default Permission Level
dm_owner	6
dm_world	1

Table 12. DCM Effective Folder Permission Set

Role or User Name	Default Permission Level
administrator	6
signatory	6
author	6
business application owner	6
consumer	3
document class owner	6
functional area supervisor	6
dm_owner	6
dm_world	1

Table 13. DCM General Folder Permission Set

Role or User Name	Default Permission Level
administrator	6
signatory	3
author	6
business application owner	6
consumer	3
document class owner	6
functional area supervisor	6
dm_owner	6
dm_world	1

Table 14. DCM In Progress Documents Permission Set

Role or User Name	Default Permission Level
administrator	7
signatory	7

Role or User Name	Default Permission Level
author	7
business application owner	7
consumer	1
document class owner	7
functional area supervisor	7
dm_owner	7
dm_world	1

Table 15. DCM In Progress Folder Permission Set

Role or User Name	Default Permission Level
administrator	6
signatory	6
author	6
business application owner	6
consumer	1
document class owner	6
functional area supervisor	6
dm_owner	6
dm_world	1

Table 16. DCM Obsolete Documents Permission Set

Role or User Name	Default Permission Level
administrator	3
signatory	1
author	1
business application owner	3
consumer	1
document class owner	3
functional area supervisor	3
dm_owner	6
dm_world	1

Table 17. DCM Obsolete Folder Permission Set

Role or User Name	Default Permission Level
administrator	3
signatory	1
author	1
business application owner	3
consumer	1
document class owner	3
functional area supervisor	3
dm_owner	6
dm_world	1

Table 18. DCM Retired Documents Permission Set

Role or User Name	Default Permission Level
administrator	6
signatory	3
author	3
business application owner	3
consumer	1
document class owner	3
functional area supervisor	3
dm_owner	6
dm_world	1

Table 19. DCM Retired Folder Permission Set

Role or User Name	Default Permission Level
administrator	3
signatory	3
author	3
business application owner	3
consumer	1
document class owner	3

Role or User Name	Default Permission Level
functional area supervisor	3
dm_owner	6
dm_world	1

Table 20. DCM In Progress Change Notice Documents Permission Set

Role or User Name	Default Permission Level
administrator	6
signatory	5
author	6
business application owner	5
consumer	5
document class owner	5
functional area supervisor	5
dm_owner	7
dm_world	1

Table 21. DCM In Progress Change Request Documents Permission Set

Role or User Name	Default Permission Level
administrator	6
signatory	5
author	6
business application owner	5
consumer	5
document class owner	5
functional area supervisor	5
dm_owner	7
dm_world	1

Table 22. DCM Suspend Folder Permission Set

Role or User Name	Default Permission Level
administrator	3

Role or User Name	Default Permission Level
signatory	3
author	3
business application owner	3
consumer	1
document class owner	3
functional area supervisor	3
dm_owner	6
dm_world	1

About DCM Overrides

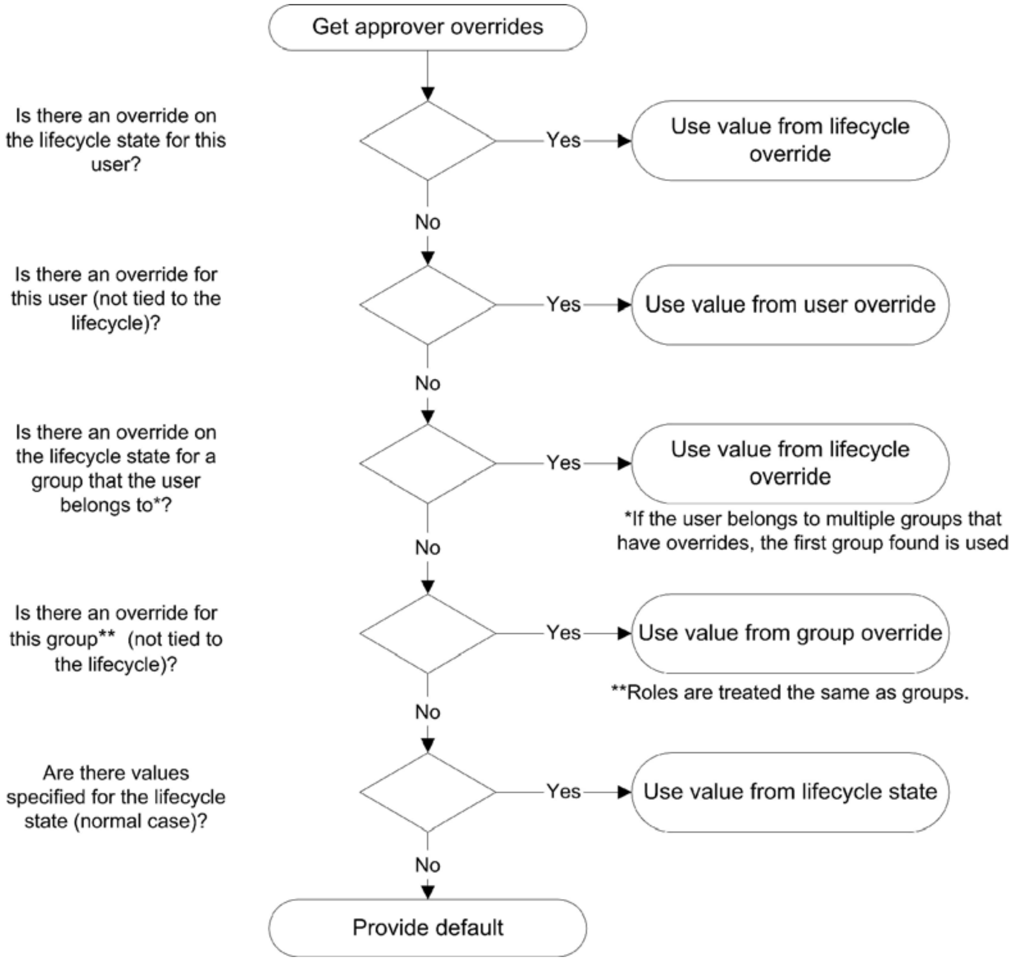
When creating or modifying users or groups in DCM, you have the option of overriding the default signoff codes, rejection codes, justification text, and notification text already defined for particular lifecycle states. If you choose to override the predefined codes and text with custom codes or text for a particular user or group, you will be prompted to choose those alternative codes and text when you create the user or group.

The override information will be used when the following actions occur:

- signing off a document
- rejecting a document
- sending a notification, such as a to-be-read notification

DCM processes overrides as shown in the following figure.

Figure 8. Override Processing in DCM


Get approver overrides
Is there an override on the lifecycle state for this user?
Yes
Use value from lifecycle override
No
Is there an override for this user (not tied to the lifecycle)?
Yes
Use value from user override
No
Is there an override on the lifecycle state for a group that the user belongs to*?
Yes
Use value from lifecycle override
*If the user belongs to multiple groups that have overrides, the first group found is used
No
Is there an override for this group** (not tied to the lifecycle)?
Yes
Use value from group override
**Roles are treated the same as groups.
No
Are there values specified for the lifecycle state (normal case)?
Yes
Use value from lifecycle state
No
Provide default

If you have multiple overrides of signoff/rejection codes or confirmation text strings specified on different DCM configuration pages, the overrides are processed in the following order:

1. DCM looks for configuration information specified for the particular user, for the current lifecycle state of the document.
2. DCM looks for configuration information specified for the particular user, but tied directly to the user and not to a lifecycle state.
3. DCM looks for configuration information specified for the first group that it can find that the user belongs to for the current lifecycle state.
4. DCM looks for configuration information specified for the first group or role that it can find that the user belongs to (independent of lifecycle state).
5. DCM looks for configuration information specified for the lifecycle state.

6. DCM uses the default values set for signoff/rejection codes and confirmation text strings.

The default values for the confirmation codes are:

- MEETS_21CFR11_REQ (“Meets 21FR11 requirements”)
- MEETS_INTERNAL_REQ (“Meets internal requirements”)
- MEETS_FDA_REQ (“Meets FDA requirements”)

The default values for the rejection codes are:

- NOT_MEET_21CFR11_REQ (“Does not meet 21CFR11 requirements”)
- NOT_MEET_INTERNAL_REQ (“Does not meet internal requirements”)
- NOT_MEET_FDA_REQ (“Does not meet FDA requirements”)

You can change these default values to match those used by your organization. For more information, see Appendix C, Adding or Modifying Reason Codes, Confirmation Codes, or Locale-Specific Value Assistance.

Creating Users

If your role is business application owner, then you can create, modify, or remove existing users.

Before you create users, determine what type of authentication your Content Server uses:

- If the server authenticates users against the operating system, each user must have an account on the server host.
- If the server uses an LDAP directory server for user authentication, the users do not need to have operating system accounts.

To create a DCM user:

1. Navigate **Administration > User Management > Users**.
2. In the left pane, click on **Users**.
3. To create a user, choose **File > New > User**.
4. Enter values in the following fields:

Table 23. User Properties Fields

Field Name	Value
State	Indicate whether the user is Active or Inactive: • An active user can connect to a repository. • An inactive user cannot connect to a repository. This option is useful if, for example, you are creating user accounts for new hires in advance of their start dates or if a user goes on a leave of absence.
User Source	Specifies how to authenticate a given repository user's user name and password. Valid values are: • **null** • **LDAP**: The user is authenticated through an LDAP directory server. • **UNIX only**: The user is authenticated using the default UNIX mechanism, dm_check_password or other external password checking program. • **Domain only**: The user is authenticated against a Windows domain. • **UNIX first**: This is used for UNIX repositories where Windows domain authentication is in use. The user is authenticated first by the default UNIX mechanism; if that fails, the user is authenticated against a Windows domain. • **Domain first** : This is used for UNIX repositories where Windows domain authentication is in use. The user is authenticated first against a Windows domain; if that fails, the user is authenticated by the default UNIX mechanism.

Field Name	Value
Description	A description of the user.
Email Address	The user's email address for receiving notifications from the repository.
User OS Name	The new user's operating system user name.
Windows Domain	On Windows, the domain name associated with the new user's Windows account. On UNIX, the domain on which the user is authenticated if Windows domain authentication is in use.
Home repository	The repository where the user receives notifications and tasks.
Default Folder	The default storage place for any object the user creates.
Default Group	To assign a default group to the new user, click **Select Group**. When the user creates an object in the repository, it belongs to the group name associated with the user's default permission set.
Default Permission Set	A permission set used to assign the default permissions to objects created by the user.
DB Name	The new user's username in the underlying RDBMS.
Privileges	Choose a user privilege from the drop-down list. User privileges authorize certain users to perform activities that are required to administer and maintain the system. The privilege levels are: • None • Create Type • Create Cabinet • Create Cabinet and Type • Create Group

Field Name	Value
	• Create Group and Type • Create Group and Cabinet • Create Group, Cabinet, and Type • System Administrator • Superuser
Extended Privileges	(Version 5.2 and later repositories only.) Sets the level of extended privileges for auditing. Superusers and Sysadmins cannot modify their own extended privileges. • **None**: The user cannot configure auditing, view audit trails, or purge audit trails. • **Config audit**: The user can configure auditing. • **Purge audit**: The user can purge existing audit trails. • **Config and Purge Audit**: The user can configure auditing and purge existing audit trails. • **View Audit**: The user can view audit trails. • **Config and View Audit**: The user can configure auditing and view existing audit trails. • **View and Purge Audit**: The user can view existing audit trails and purge them. • **Config, View, and Purge Audit**: The user can configure auditing and view and purge existing audit trails.
Alias Set	The default alias set for the user.
Workflow Disabled	Indicates whether a user can receive workflow tasks.
Turn off authentication failure checking	If checked, user may exceed the number of failed logins specified in the **Maximum Authentication Attempts** field of the repository config object

5. Click **Next** to continue.

   The **DCM Overrides** page appears.

6. To override any signoff codes, rejection codes, confirmation text, or notification text strings already defined on the system with custom options for this user, click the **Use Override** checkbox.

7. If you checked the **Use Override** checkbox, the tab refreshes to display new options.

8. If you selected **DCM Override** , modify the values for one or more of the fields.

   The DCM configuration values listed on the **DCM Override** page can be specified at each lifecycle state, or they can be specified per user, group, or role.

Table 24. DCM Override Page Fields

Field Name	Value
Allow manual promotion	If checked, this option specifies if the system should display a Promote button for a particular signatory in a specific lifecycle state.
Signoff Code	The justification, or reason code text, that a user must use when performing a signoff or promote process.
Signoff Confirmation Text	The text for the signoff confirmation. Click **Edit** to choose from a list of pre-existing signoff confirmation text strings. You can also type a new signoff confirmation. For example, "`Meets FDA requirements.`" This confirmation text applies to DCM workflow task processes which require signature (such as Finish, Reject, Delegate, and Close). When a controlled document is routed via a workflow, then the justification and confirmation defined for this document type will be enforced and the justification selected will be applied to its signature page.

Field Name	Value
Reject Code	The rejection text list that a user selects from when performing a reject operation.
Reject Confirmation Text	The text for the rejection confirmation.
Notification Text	Text used in the notification process.
TBR Notification Text	Text used in the to-be-read notification process.

9. Click **OK** to save your changes and to create the user.

Searching for Users

You can search for existing users in the repository by specifying either repository user name, operating system name, or group membership.

To search for a DCM user:

1. Navigate to **Administration > User Management > Users**.
2. To search for a specific user, enter values in one or more of the displayed fields, then click **Go**.
3. To show a list of all defined DCM users in the repository, choose **Show All Users** from the pull-down menu in the upper right-hand corner of the page.

Viewing or Modifying User Properties

You can edit properties, such as permissions, group membership or email address for existing users in the repository by modifying the Properties page associated with a particular user.

To edit properties for an existing DCM user:

1. Navigate to **Administration > User Management > Users**.
2. To search for a specific user, enter values in one or more of the displayed fields, then click **Go**.
3. To show a list of all defined DCM users in the repository, choose **Show All Users** from the pull-down menu in the upper right-hand corner of the page.

4. To modify an existing user, right-click on a user and select **Properties**.
5. Change any or all of the values for the following fields:

Table 25. User Properties Fields

Field Name	Value
State	Indicate whether the user is Active or Inactive: • An active user can connect to a repository. • An inactive user cannot connect to a repository. This option is useful if, for example, you are creating user accounts for new hires in advance of their start dates or if a user goes on a leave of absence.
User Source	Specifies how to authenticate a given repository user's user name and password. Valid values are: • **null** • **LDAP**: The user is authenticated through an LDAP directory server. • **UNIX only**: The user is authenticated using the default UNIX mechanism, dm_check_password or other external password checking program. • **Domain only**: The user is authenticated against a Windows domain. • **UNIX first**: This is used for UNIX repositories where Windows domain authentication is in use. The user is authenticated first by the default UNIX mechanism; if that fails, the user is authenticated against a Windows domain. • **Domain first** : This is used for UNIX repositories where Windows domain authentication is in use. The user is authenticated first against a

Field Name	Value
	Windows domain; if that fails, the user is authenticated by the default UNIX mechanism.
Description	A description of the user.
Email Address	The user's email address for receiving notifications from the repository.
User OS Name	The new user's operating system user name.
Windows Domain	On Windows, the domain name associated with the new user's Windows account. On UNIX, the domain on which the user is authenticated if Windows domain authentication is in use.
Home repository	The repository where the user receives notifications and tasks.
Default Folder	The default storage place for any object the user creates.
Default Group	To assign a default group to the new user, click **Select Group**. When the user creates an object in the repository, it belongs to the group name associated with the user's default permission set.
Default Permission Set	A permission set used to assign the default permissions to objects created by the user.
DB Name	The new user's username in the underlying RDBMS.

Field Name	Value
Privileges	Choose a user privilege from the drop-down list. User privileges authorize certain users to perform activities that are required to administer and maintain the system. The privilege levels are: • None • Create Type • Create Cabinet • Create Cabinet and Type • Create Group • Create Group and Type • Create Group and Cabinet • Create Group, Cabinet, and Type • System Administrator • Superuser
Extended Privileges	(Version 5.2 and later repositories only.) Sets the level of extended privileges for auditing. Superusers and Sysadmins cannot modify their own extended privileges. • **None**: The user cannot configure auditing, view audit trails, or purge audit trails. • **Config audit**: The user can configure auditing. • **Purge audit**: The user can purge existing audit trails. • **Config and Purge Audit**: The user can configure auditing and purge existing audit trails. • **View Audit**: The user can view audit trails. • **Config and View Audit**: The user can configure auditing and view existing audit trails.

Field Name	Value
	• **View and Purge Audit**: The user can view existing audit trails and purge them. • **Config, View, and Purge Audit**: The user can configure auditing and view and purge existing audit trails.
Alias Set	The default alias set for the user.
Workflow Disabled	Indicates whether a user can receive workflow tasks.
Turn off authentication failure checking	If checked, user may exceed the number of failed logins specified in the **Maximum Authentication Attempts** field of the repository config object

6. Click **Next** to continue.

   The **DCM Overrides** page appears.

7. To override any signoff codes, rejection codes, confirmation text, or notification text strings already defined on the system with custom options for this user, click the **Use Override** checkbox.

8. If you checked the **Use Override** checkbox, the tab refreshes to display new options.

9. If you selected **DCM Override** , modify the values for one or more of the fields.

   The DCM configuration values listed on the **DCM Override** page can be specified at each lifecycle state, or they can be specified per user, group, or role.

Table 26. DCM Override Page Fields

Field Name	Value
Allow manual promotion	If checked, this option specifies if the system should display a Promote button for a particular signatory in a specific lifecycle state.
Signoff Code	The justification, or reason code text, that a user must use when performing a signoff or promote process.

Field Name	Value
Signoff Confirmation Text	The text for the signoff confirmation. Click **Edit** to choose from a list of pre-existing signoff confirmation text strings. You can also type a new signoff confirmation. For example, "`Meets FDA requirements.`" This confirmation text applies to DCM workflow task processes which require signature (such as Finish, Reject, Delegate, and Close). When a controlled document is routed via a workflow, then the justification and confirmation defined for this document type will be enforced and the justification selected will be applied to its signature page.
Reject Code	The rejection text list that a user selects from when performing a reject operation.
Reject Confirmation Text	The text for the rejection confirmation.
Notification Text	Text used in the notification process.
TBR Notification Text	Text used in the to-be-read notification process.

10. When you have finished making changes, click **OK**.

Creating Groups

You create groups through the **DCM Administration** screen. The options displayed on this screen depend on the user role associated with your login ID. If your role is DCM administrator or business application owner, then you can create, modify, or remove existing groups.

To create a DCM group:

1. Navigate to **Administration > User Management > Groups**.
2. To create a user, choose **File > New > Group**.

3. Enter values for the following fields:

Table 27. Group Properties fields

Field Label	Value
Name	The name of the new repository group.
Class	The type of group. The default is **Group**. The other valid value is **Role**. Use this attribute so that your applications can distinguish between groups and roles. The server does not enforce the value of this attribute and does not set the attribute to any value other than group. Server 5.x only. If you set this to **Role**, the group does not appear on the **Groups** list page.
Email Address	The email address for the new group. If no value is entered in this field, the group email address defaults to the group name.
Owner	The name of a repository user who has the Create Group privilege and who owns this group. If you are a superuser, you can select the owner. Otherwise, you can set this to a group of which you are a member. To change the default owner name, click **Browse**.
Administrator	Specifies a user or group, in addition to a superuser or the group owner, who can modify the group. Click **Browse** to select the user or group. If this is null, only a superuser and the group owner can modify the group. Server 5.x only.
Alias Set	The default alias set for the group.

Field Label	Value
Description	A description of the group. For example, "`New Project Signoffs Committee`"
Is Private	Defines whether the group is private. If is unchecked, the group is created as a public group. By default, groups created by users with Sysadmin or Superuser privileges are public, and groups created by users with a lower user privilege level are private.

4. Click **Next** to continue.

   The **DCM Overrides** page appears.

5. To override any signoff codes, rejection codes, confirmation text, or notification text strings already defined on the system with custom options for this user, click the **Use Override** checkbox.

6. If you checked the **Use Override** checkbox, the tab refreshes to display new options.

7. If you selected **DCM Override** , modify the values for one or more of the fields.

   The DCM configuration values listed on the **DCM Override** page can be specified at each lifecycle state, or they can be specified per user, group, or role.

Table 28. DCM Override Page Fields

Field Name	Value
Allow manual promotion	If checked, this option specifies if the system should display a Promote button for a particular signatory in a specific lifecycle state.
Signoff Code	The justification, or reason code text, that a user must use when performing a signoff or promote process.

Field Name	Value
Signoff Confirmation Text	The text for the signoff confirmation. Click **Edit** to choose from a list of pre-existing signoff confirmation text strings. You can also type a new signoff confirmation. For example, "`Meets FDA requirements.`" This confirmation text applies to DCM workflow task processes which require signature (such as Finish, Reject, Delegate, and Close). When a controlled document is routed via a workflow, then the justification and confirmation defined for this document type will be enforced and the justification selected will be applied to its signature page.
Reject Code	The rejection text list that a user selects from when performing a reject operation.
Reject Confirmation Text	The text for the rejection confirmation.
Notification Text	Text used in the notification process.
TBR Notification Text	Text used in the to-be-read notification process.

8. Click **OK** to save your changes.

Viewing or Modifying Group Properties

You can edit properties, such as permissions, group membership or email address for existing groups in the repository by modifying the Properties page associated with a particular group. If your role is DCM administrator or business application owner, then you can create, modify, or remove existing groups.

To view or modify the properties of a DCM group:

1. Navigate to **Administration > User Management > Groups**.

2. To view or modify the properties for an existing group, right-click on a group and select **Properties**.
3. Refer to the table below for a description of the attributes:

Table 29. Group Properties fields

Field Label	Value
Name	The name of the new repository group.
Class	The type of group. The default is **Group**. The other valid value is **Role**. Use this attribute so that your applications can distinguish between groups and roles. The server does not enforce the value of this attribute and does not set the attribute to any value other than group. Server 5.x only. If you set this to **Role**, the group does not appear on the **Groups** list page.
Email Address	The email address for the new group. If no value is entered in this field, the group email address defaults to the group name.
Owner	The name of a repository user who has the Create Group privilege and who owns this group. If you are a superuser, you can select the owner. Otherwise, you can set this to a group of which you are a member. To change the default owner name, click **Browse**.
Administrator	Specifies a user or group, in addition to a superuser or the group owner, who can modify the group. Click **Browse** to select the user or group. If this is null, only a superuser and the group owner can modify the group. Server 5.x only.

Field Label	Value
Alias Set	The default alias set for the group.
Description	A description of the group. For example, "`New Project Signoffs Committee`"
Is Private	Defines whether the group is private. If **Is Private** is unchecked, the group is created as a public group. By default, groups created by users with Sysadmin or Superuser privileges are public, and groups created by users with a lower user privilege level are private.

4. Click **Next** to continue.

   The **DCM Overrides** page appears.

5. To override any signoff codes, rejection codes, confirmation text, or notification text strings already defined on the system with custom options for this user, click the **Use Override** checkbox.

6. If you checked the **Use Override** checkbox, the tab refreshes to display new options.

7. If you selected **DCM Override** , modify the values for one or more of the fields.

   The DCM configuration values listed on the **DCM Override** page can be specified at each lifecycle state, or they can be specified per user, group, or role.

Table 30. DCM Override Page Fields

Field Name	Value
Allow manual promotion	If checked, this option specifies if the system should display a Promote button for a particular signatory in a specific lifecycle state.
Signoff Code	The justification, or reason code text, that a user must use when performing a signoff or promote process.

Field Name	Value
Signoff Confirmation Text	The text for the signoff confirmation. Click **Edit** to choose from a list of pre-existing signoff confirmation text strings. You can also type a new signoff confirmation. For example, "`Meets FDA requirements.`" This confirmation text applies to DCM workflow task processes which require signature (such as Finish, Reject, Delegate, and Close). When a controlled document is routed via a workflow, then the justification and confirmation defined for this document type will be enforced and the justification selected will be applied to its signature page.
Reject Code	The rejection text list that a user selects from when performing a reject operation.
Reject Confirmation Text	The text for the rejection confirmation.
Notification Text	Text used in the notification process.
TBR Notification Text	Text used in the to-be-read notification process.

8. Click **OK** to save your changes.

The DCM Overrides Page

The DCM configuration values listed on the **DCM Overrides** page can be specified at each lifecycle state, or they can be specified per user, group, or role.

Table 31. DCM Override Page Fields

Field Name	Value
Allow manual promotion	If checked, this option specifies if the system should display a Promote button for a particular signatory in a specific lifecycle state.
Signoff Code	The justification, or reason code text, that a user must use when performing a signoff or promote process.
Signoff Confirmation Text	The text for the signoff confirmation. Click **Edit** to choose from a list of pre-existing signoff confirmation text strings. You can also type a new signoff confirmation. For example, "`Meets FDA requirements.`" This confirmation text applies to DCM workflow task processes which require signature (such as Finish, Reject, Delegate, and Close). When a controlled document is routed via a workflow, then the justification and confirmation defined for this document type will be enforced and the justification selected will be applied to its signature page.
Reject Code	The rejection text list that a user selects from when performing a reject operation.
Reject Confirmation Text	The text for the rejection confirmation.
Notification Text	Text used in the notification process.
TBR Notification Text	Text used in the to-be-read notification process.

Creating Roles

If your role is DCM administrator or business application owner, then you can create, modify, or remove existing roles.

To create a DCM role:

1. Navigate to **Administration > User Management > Roles**.
2. To define a new role, choose **File > New > Role**.
3. Refer to the table below for a description of the attributes.

Table 32. Role Info Page fields

Field Label	Value
Name	The name of the new role.
Class	The type of group. The default value is **Role**. The other valid value is **Group**. Use this attribute so that your applications can distinguish between groups and roles. The server does not enforce the value of this attribute and does not set the attribute to any value other than group.
Email Address	The email address for the new role. If no value is entered in this field, the role email address defaults to the role name.
Owner	The name of a repository user who has the Create Group privilege and who owns this role. If you are a superuser, you can select the owner. Otherwise, you can set this to a group of which you are a member. To change the default owner name, click **Browse**.
Administrator	Specifies a user or group, in addition to a superuser or the group owner, who can modify this role. Click **Browse** to select the user or group. If you do not specify a value, only a superuser and the role owner can modify the role.
Alias Set	The default alias set for the role.

Field Label	Value
Description	A description of the group. For example, "`New Project Signatories`."
Is Private	Defines whether the group is private. If **Is Private** is unchecked, the group is created as a public group. By default, groups created by users with Sysadmin or Superuser privileges are public, and groups created by users with a lower user privilege level are private.
Create Role as Domain	If checked, the role is created as a domain. A domain role represents a particular client domain. A domain role contains a set of role groups, corresponding to the roles recognized by the client application. If you create a role as a domain, it is listed on the **Groups** list page, not the **Roles** list page.

4. After entering values in the fields, click **OK** to save your changes.

Viewing or Modifying Roles

If your role is DCM administrator or business application owner, then you can modify or remove existing roles. For example, you may wish to expand or restrict the privileges associated with a particular role.

To modify an existing role:

1. Navigate to **Administration > User Management > Roles**.
2. To view or modify the properties for an existing role, right-click on a role and select **Properties**.
3. Refer to the table below for a description of the attributes.

Table 33. Role Info Page fields

Field Label	Value
Name	The name of the new role.
Class	The type of group. The default value is **Role**. The other valid value is **Group**. Use this attribute so that your applications can distinguish between groups and roles. The server does not enforce the value of this attribute and does not set the attribute to any value other than group.
Email Address	The email address for the new role. If no value is entered in this field, the role email address defaults to the role name.
Owner	The name of a repository user who has the Create Group privilege and who owns this role. If you are a superuser, you can select the owner. Otherwise, you can set this to a group of which you are a member. To change the default owner name, click **Browse**.
Administrator	Specifies a user or group, in addition to a superuser or the group owner, who can modify this role. Click **Browse** to select the user or group. If you do not specify a value, only a superuser and the role owner can modify the role.
Alias Set	The default alias set for the role.
Description	A description of the group. For example, "`New Project Signatories`."

Field Label	Value
Is Private	Defines whether the group is private. If **Is Private** is unchecked, the group is created as a public group. By default, groups created by users with Sysadmin or Superuser privileges are public, and groups created by users with a lower user privilege level are private.
Create Role as Domain	If checked, the role is created as a domain. A domain role represents a particular client domain. A domain role contains a set of role groups, corresponding to the roles recognized by the client application. If you create a role as a domain, it is listed on the **Groups** list page, not the **Roles** list page.

4. Click **Next** to continue.

   The **DCM Overrides** page appears.

5. To override any signoff codes, rejection codes, confirmation text, or notification text strings already defined on the system with custom options for this user, click the **Use Override** checkbox.

6. If you checked the **Use Override** checkbox, the tab refreshes to display new options.

7. If you selected **DCM Override** , modify the values for one or more of the fields.

   The DCM configuration values listed on the **DCM Override** page can be specified at each lifecycle state, or they can be specified per user, group, or role.

Table 34. DCM Override Page Fields

Field Name	Value
Allow manual promotion	If checked, this option specifies if the system should display a Promote button for a particular signatory in a specific lifecycle state.

Field Name	Value
Signoff Code	The justification, or reason code text, that a user must use when performing a signoff or promote process.
Signoff Confirmation Text	The text for the signoff confirmation. Click **Edit** to choose from a list of pre-existing signoff confirmation text strings. You can also type a new signoff confirmation. For example, "`Meets FDA requirements.`" This confirmation text applies to DCM workflow task processes which require signature (such as Finish, Reject, Delegate, and Close). When a controlled document is routed via a workflow, then the justification and confirmation defined for this document type will be enforced and the justification selected will be applied to its signature page.
Reject Code	The rejection text list that a user selects from when performing a reject operation.
Reject Confirmation Text	The text for the rejection confirmation.
Notification Text	Text used in the notification process.
TBR Notification Text	Text used in the to-be-read notification process.

8. When you have finished modifying values, click **OK** to save your changes.

Assigning Groups to Users

When you create a user, you can assign a default group.

The options displayed on this screen depend on the user role associated with your login ID. If your role is DCM administrator or business application owner, then you can create, modify, or remove existing permissions sets.

To assign a group permission to a user:

1. In Webtop, click the **Select Group** link on the **Create User** page.
2. To change the displayed options, select what kinds of items you want to see from the pull-down menu at the bottom of the page.
3. To select the name of a group or role to assign to a user, click the checkbox corresponding to that group or role name.
4. Click **OK** to save your changes and to return to the **Create User** page.

Assigning Permission Sets to Users

When you create a user, you can assign a default permission set to objects created by that user.

The options displayed on this screen depend on the user role associated with your login ID. If your role is DCM administrator or business application owner, then you can create, modify, or remove existing permissions sets.

To assign a permission set to a user:

1. In Webtop, click the **Select Permission Set** link on the **Create User** page.
2. To change the displayed options, select what kinds of items you want to see from the pull-down menu at the bottom of the page.
3. To select the name of a group or role to assign to a user, click the checkbox corresponding to that group or role name.
4. Click **OK** to save your changes and to return to the **Create User** page.

Business Rules

Business rules automatically assign signatories, distribution list, documents, methods, lifecycles, and other objects to a document class based on certain conditions.

Note: You must be in the Functional Area Supervisor role to display User List Rules.

Business rules automatically specify mandatory and available signatories and distribution lists for notifications of controlled documents in a lifecycle state based on criteria or attributes of the document

You can define a business rule to activate when certain conditions have been met for particular document instances. For example, you can specify that if cost on an item equals one million dollars, then the CFO is required to sign the document.

User List Rules and Lifecycle States

Once defined, user list rules are only applied when a document is promoted to the next lifecycle state (and for subsequent promotions). A user list rule will apply to all documents that meet certain criteria, as specified in the fields on the **User List Rules Properties** page.

User list rule signatories and distribution lists are tied to lifecycle state types. You can select users, roles or groups as signatories when defining a user list rule. Mandatory signatories are added to a document's Required Approver list if certain conditions are triggered.

Tip: Best Practices Tip: Because user list rules are based on lifecycle state types, make sure that you define signatories only for those states where signatory lists are relevant. For example, you may want to define a user list rule for documents when they are promoted to the In Approval state. However, it would be meaningless to define a list of signatories for an Obsolete lifecycle state.

Creating a User List Rule

User list rules allows you to define the parameters for a dynamic user list that is generated by a query and triggered when specified conditions are met. To create or modify a user list rule, you must log into DCM as a user who has a role of Functional Area Supervisor.

To create a user list rule:

1. Navigate to **Administration > DCM > User List Rules**.
2. To create a new rule, choose **File > New > Business Rule > User List Rule**.
3. Type a name for the new rule, then click **Next**.
4. Refer to the table below for a description of the attributes.

Table 35. New User List Rule: Info Page Fields

Field Name	Value
Business Application	Click **Select** to choose the name of an existing business application that you want this user list rule to apply to.
Document Class	Click **Select** to choose the name of an existing document class that you want this user list rule to apply to.

Field Name	Value
DCM State Type	Choose the lifecycle state type to which you want this user list rule to apply. The lifecycle states listed here are the default DCM state types.
User List Type	Click a radio button to define what kind of user list to create: • **TBR Distribution List:** Adds the dynamically selected users to a To-Be-Read notification list. • **Mandatory Signatories**: Adds the dynamically selected users to a mandatory signatories list when certain conditions (such as a lifecycle state change or workflow task) are met. • **Optional Signatories**: Adds the dynamically selected users to an optional signatories list when certain conditions (such as a lifecycle state change or workflow task) are met.
Property	Choose the name of an object property, a conditional operator, and type in the condition you want met in these fields. If the conditions you specify in these fields are true, then the user list will be added to the given process you specified with the radio buttons, above. To add additional conditions to the generated query, click **Add Property**.

5. Click **Next** to continue.
6. To add a user or group to the list of available users and groups for the dynamically-generated user list, click **Add**.
7. To change the displayed options, select what kinds of items you want to see from the pull-down menu at the bottom of the page.
8. To select the name of a user, group or role to assign as an signatory, click the checkbox corresponding to that group or user name, then click **Add**.

9. Click **OK** to save your changes and to return to the **New User List Rule: Users/Groups** page.
10. To specify that the selected users, group, or roles are Mandatory Signatories, click the **Mandatory Signatories** checkbox.
11. To specify that any member of a selected group or role can signoff a document on behalf of all the group or role members, click the **Any Members** checkbox.
12. Click **Finish** to save your new user list rule.

Any Member Group Signoffs

Administrators can configure groups and roles as signatories for user list rules and DCM lifecycle extensions. They can further specify that a single member of the selected group or role can signoff the document.

When you choose users, group, or roles as signatories in the DCM configuration screens, you can select the **Any Member** option, which indicates that any member of the selected group or role can signoff a document on behalf of the entire group or role. When the first person in this group signs off the document, then the remaining group members' signoff will be cancelled and they no longer have the task to signoff the document. Once approved by a single member of the group, the document can then be promoted if there are no other pending signoffs.

A user must specify which group they are signing off for from a pulldown so that if a user is set up as an individual and as a member of a group or as a member of multiple groups, he can sign off individually for each one.

Viewing or Modifying a User List Rule

To view or modify a user list rule, you must log into DCM as a user who has a Functional Area Supervisor role.

To view or edit a user list rule:

1. Navigate to **Administration > DCM > User List Rules**.
2. To view or modify the properties for an existing user list rule, right-click on a role and select **Properties**.
3. Refer to the table below for a description of the attributes.

Table 36. User List Rule Properties: Info Page Fields

Field Name	Value
Business Application	Click **Select** to choose the name of an existing business application that you want this user list rule to apply to.
Document Class	Click **Select** to choose the name of an existing document class that you want this user list rule to apply to.
DCM State Type	Choose the lifecycle state type to which you want this user list rule to apply. The lifecycle states listed here are the default DCM state types.
User List Type	Click a radio button to define what kind of user list to create: • **TBR Distribution List:** Adds the dynamically selected users to a To-Be-Read notification list. • **Mandatory Signatories**: Adds the dynamically selected users to a mandatory signatories list when certain conditions (such as a lifecycle state change or workflow task) are met. • **Optional Signatories**: Adds the dynamically selected users to an optional signatories list when certain conditions (such as a lifecycle state change or workflow task) are met.
Property	Choose the name of an object property, a conditional operator, and type in the condition you want met in these fields. If the conditions you specify in these fields are true, then the user list will be added to the given process you specified with the radio buttons, above. To add additional conditions to the generated query, click **Add Property**.

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

5. To view which users, groups, and roles have permissions for this user list rule, click the **Permissions** tab.

6. To add a user, group, or role to the list of users and groups with permissions on this user list rule:

   a. Click the Add icon ⊕.

      The **Add A User** page appears.

   b. Click the checkboxes associated with the names of the users, groups, or roles you want, then click **Add**.

   c. Click **OK** when you have finished making your selections.

   d. Click a radio button to assign a **Basic Permission** level.

   e. Click one or more checkboxes to assign **Extended Permissions**.

   f. Click **OK** to save your changes.

      The **Permissions** page reappears, reflecting the changes you made.

7. To change permissions for a user, group, or role already associated to this user list rule:

   a. Click the checkbox associated with the user, group, or role whose permission you want to edit.

   b. Click the Edit icon ✎.

   c. Optionally, click a radio button to change an assigned **Basic Permission** level.

   d. Optionally, click one or more checkboxes to change or add **Extended Permissions**.

   e. Click **OK** to save your changes.

      The **Permissions** page reappears, reflecting the changes you made.

8. To change the active permission set associated with this user list rule, click the **Select** link, select another permission set from the **Choose a permission set** page, then click **OK**.

9. To modify values on **User List Rule : Users/Groups** page, click the **Users/Groups** tab.

10. To add or remove a user or group to or from the list of available users and groups for the dynamically-generated user list, click **Add** or **Remove**.

11. To change the displayed options, select what kinds of items you want to see from the pull-down menu at the bottom of the page.

12. To select the name of a user, group or role to assign as an signatory, click the checkbox corresponding to that group or user name, then click **Add**.

13. Click **OK** to save your changes and to return to the **User List Rule: Users/Groups** page.

14. To specify that the selected users, group, or roles are Mandatory Signatories, click the **Mandatory Signatories** checkbox.

15. To specify that any member of a selected group or role can signoff a document on behalf of all the group or role members, click the **Any Members** checkbox.

16. Click **OK** to save your changes.

17. To view audit trail information for this user list rule, click the **History** tab.

Chapter 5

Creating and Administering Business Applications

This chapter introduces DCM business applications, and explains how to create and administer them. The topics discussed include:

Business Applications

A business application is a document-based custom application that defines how an organization enforces a regulatory process such as Manufacturing Procedures.

Note: The association between document class and business application is created by inheritance whereby the document class is set to inherit from the business application template, or to override it. Two document classes associated with the same business application behave similarly if they have the same inheritance settings, or differently if not

Document classes are grouped and managed in a business application. You can set the defaults and common properties for all the document classes included in a business application. Each business application is like a container, holding information about the type of documents, business rules, lifecycles, workflows, permissions, and users that apply to a particular regulatory process.

A business application typically consists of the following DCM components:

- document classes
- document relationships
- autonaming schemes

- document templates
- business rules

Each business application can have one or more document classes. Each document class can point to any Documentum object type. The document class's behavior is defined by several custom properties such as its default lifecycle and relationship types. You can define a single lifecycle or workflow for use in multiple document classes

The configuration parameters for your business application are stored as a DAR file. Once you create the configuration file, you can make further configuration changes. You can also use the configuration file to configure DCM on another machine. For example, you can create the business application on a test machine, then easily move to one or more production server machines.

Task Overview

To create a business application, perform the following task sequence:

1. Identify your business application requirements

   What kinds of documents and processes you want DCM to manage? What kinds of regulatory processes do you need to enforce?

2. Identify document classes and their desired properties
3. Identify change notice (CN) and change request (CR) types and their desired properties
4. Identify users, groups, permission sets, lifecycles, and workflows
5. Create users, roles, groups, and permissions sets.

   For more information on how to create users, roles, groups, and permissions sets, see the *Content Server Administrator's Guide*.

6. Create custom object types for your DCM application.

   For more information on how to create types, see the *Content Server Administrator's Guide*.

7. Choose whether to enforce unique naming for controlled documents at the repository level or the business application level:

   - To enforce unique naming at the repository level, you must set the value of the **uniquenamingatdocbase** property to `TRUE` in the app.xml configuration file.

     For detailed instructions on how to customize or configure DCM using the XML configuration files included with the product, see the *DCM Development Guide*.

- To enforce unique naming within a business application, select the **Unique Name** option on the Business Application configuration screen.

8. Create lifecycles.

   You create lifecycles using Documentum Application Builder. For detailed instructions, see the *Documentum Application Builder User Guide*.

9. Create custom workflow templates.

   You create workflows using Documentum Workflow Manager.

10. Create or import custom document templates

    The procedure for importing a template is the same as importing any other object into your repository. It is important to note, however, that when importing templates into your repository, you must import them into the Templates cabinet and you must import one template for each object type that you want to make available in your repository.

11. Optionally, define an autonaming scheme. For more information, see Chapter 10, Setting Up Autonaming Schemes.

12. Define document classes. For more information, see Chapter 6, Setting Up Document Classes and Document Relationships .

13. Optionally, create change notice and change request document classes.

    DCM provides default change notice and change request document classes. You can modify these existing document classes, or create new document classes for your change notices and change requests. For more information, see .

14. Define a business application.

15. Edit lifecycle states to add DCM-specific extensions. For more information, see Chapter 8, Creating and Editing Controlled Document Lifecycles.

16. Define document relationships. For more information, see Chapter 6, Setting Up Document Classes and Document Relationships .

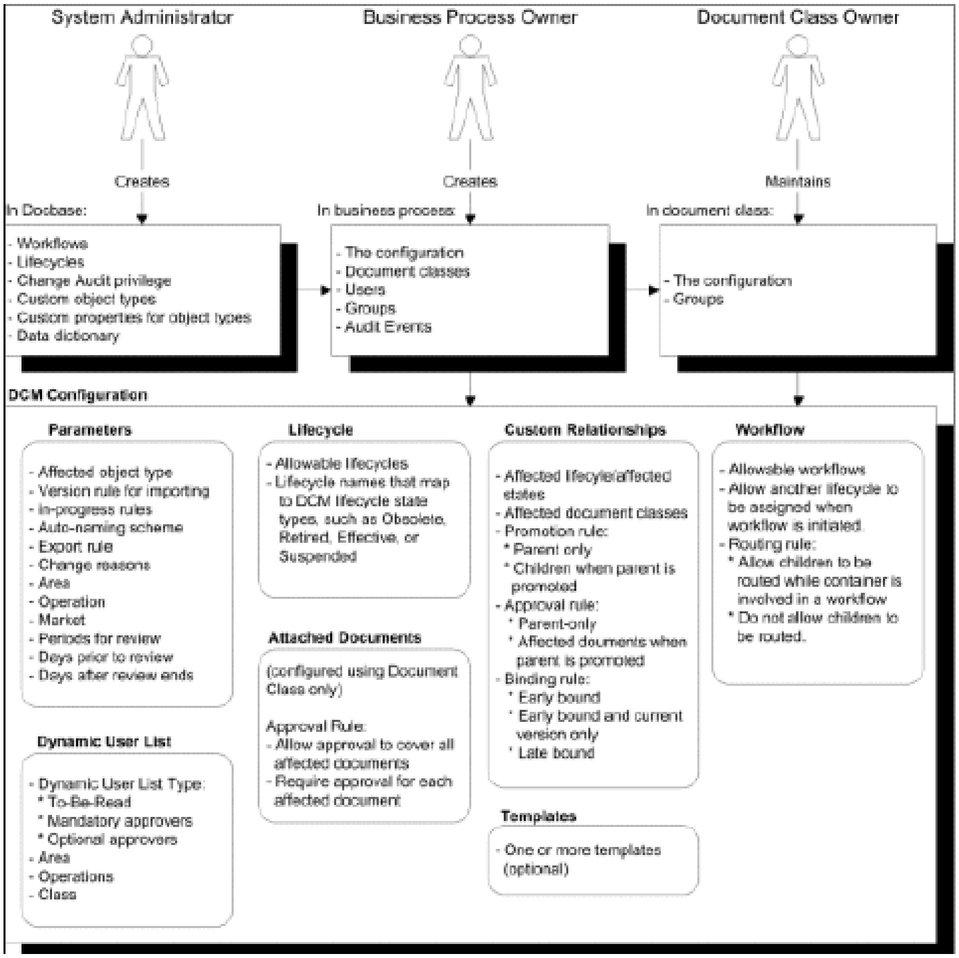
17. Define business rules. For more information, see Business Rules, page 101.

18. Put your company's logo and other images on the customized DCM screens.

    For more information about changing the appearance of DCM using branding and themes, see the *Web Development Kit and Client Applications Development Guide*.

The following figure shows the typical task sequence for creating a custom DCM business application.

Figure 9. Building a DCM business application



Note: Binding rules are defined by the relationship type you choose.

In most cases, you will initially perform these business application setup tasks in a development environment. Once you have finished and tested your business application, you package the application in a master DAR file. You can then install this business application on other machines. For detailed instructions on DARs, refer to *EMC Documentum Composer User Guide*, version 6.5.

Creating a Business Application

You define your business application only after you have defined your requirements, and created the necessary users/groups/roles, objects, workflows, and lifecycles.

The **Business Applications** page allows you to create a business application, as well as view existing business applications.

You must be a "business application owner" in order to create a business application. The Business Applications node is not visible if you are not a "business application owner".

To create a business application:

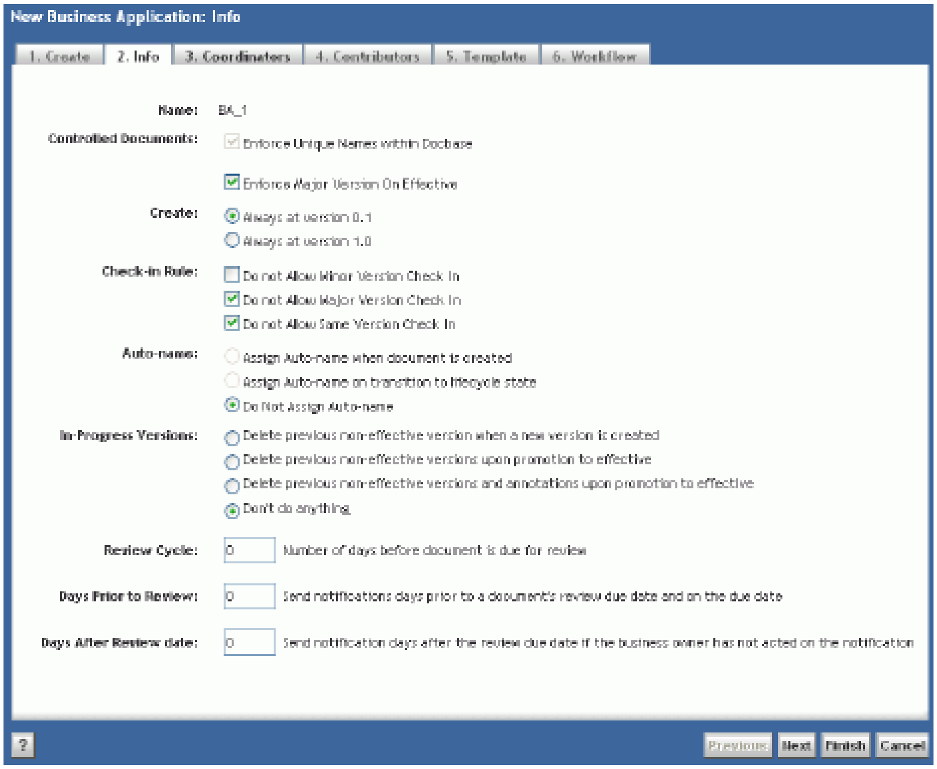
Note: You can create a business application according to this procedure by simply giving it a unique name and clicking **Finish** or, follow all steps to provide values where default settings need to change. Adding a value for the **Name** and clicking **Finish** however creates identical business applications with unique names. Only the **Name** field is mandatory so default values in any of the optional fields must be changed to make them truly different (apart from having unique names and the same default settings).

1. Navigate to **Administration > DCM > Business Applications** and select **File > New > Business Application**.
2. Type a name for the new business application, then click **Next** to accept the value and move onto the next tab. A value for the mandatory **Name** field, at the very least, must be provided to create a business application.

   The **Info** tab is displayed showing the default settings for each option.

   Note: Clicking **Finish** accepts the values you entered (along with any default values you did not change), closes the **New Business Application** screen, and lists the new business application object (with the name you typed) in the **Business Applications** pane. Though you can click **Finish** anytime to create the business application object, you can always return to it using its **Properties** screen to modify or complete the entries needed. **Cancel** aborts the operation.

Figure 10. New Business Application screen



Name: BA_1

Controlled Documents: ☑ Enforce Unique Names within Docbase

☑ Enforce Major Version On Effective

Create:
● Always at version 0.1
○ Always at version 1.0

Check-in Rule:
□ Do not Allow Minor Version Check In
☑ Do not Allow Major Version Check In
☑ Do not Allow Same Version Check In

Auto-name:
○ Assign Auto-name when document is created
○ Assign Auto-name on transition to lifecycle state
● Do Not Assign Auto-name

In-Progress Versions:
○ Delete previous non-effective version when a new version is created
○ Delete previous non-effective versions upon promotion to effective
○ Delete previous non-effective versions and annotations upon promotion to effective
● Don't do anything

Review Cycle: 0 Number of days before document is due for review

Days Prior to Review: 0 Send notifications days prior to a document's review due date and on the due date

Days After Review date: 0 Send notification days after the review due date if the business owner has not acted on the notification

3. Enter values for the following fields, as needed, on the **Info** tab. The default settings are enforced if you make no changes and click **Next**.

   Note: No notifications are sent if the values for the **Review Cycle**, **Days Prior to Review**, and **Days After Review date** are left as is with the default value set to **0** (zero).

Table 37. Business Application: Info Page Fields

Field Name	Value
Controlled Documents	DCM allows you to enforce unique naming of controlled documents at either the business application level or the repository (docbase) level. If repository-level enforcement of unique naming has been configured, then the **Enforce Unique Names within Docbase** option cannot be selected on this page.

Field Name	Value
	Click the **Enforce Major Version On Effective** checkbox to ensure unique names are enforced for controlled documents. If **Enforce Major Version On Effective** is selected, the repository version number for a controlled document will be promoted to the next major version when the document reaches the Effective state. For example, a document with a version number of 1.3 will be promoted to version 2.0 when it reaches the Effective lifecycle state. If the controlled document is already at a major version number when promoted to the Effective state, it will not be further promoted.
Create	To specify what repository version to assign the initial checkin of a new document created using this document class, choose one of the following options: • **Always at version 0.1** • **Always at version 1.0**
Check-in Rule	To restrict the repository versioning behavior for documents of this class, choose one of the following options: • **Do not Allow Minor Version Checkin**: If selected, forces the user to check in documents based on this class as major versions only (1.0, 2.0, 3.0, etc.) • **Do not Allow Major Version Checkin**: If selected, forces the user to check in documents based on this class as minor versions only (1.1, 1.2, 1.3, etc.) • **Do not Allow Same Version Checkin**: If selected, disables the

Field Name	Value
	Check In As Same Version option in the checkin dialog, and forces the user to check in documents as incremented major or minor versions only. **Note:** Though you can select two options, not all combinations are permitted, for example, you can not have the first two options selected nor all three.
Auto-Name	Specifies when an auto-naming scheme is applied to documents of this class: • **Assign Auto-name when document is created**: If selected, the auto-naming scheme is applied immediately upon document creation. • **Assign Auto-name on transition to lifecycle state**: If selected, the auto-naming scheme is applied only when the document reaches a certain lifecycle state (for example, Approved). • **Do Not Assign Auto-name**: If selected, disables the system auto-naming scheme for documents of this document class. **Note:** "dcm.autoname.service.allow_rename" in dcm.properties will determine whether the auto-naming scheme allows renaming of a controlled document or not. The default value setting for this property is *false*. This means that when **Assign Auto-name when document is created** is selected for a document class, the controlled document is named according to the auto-naming scheme upon checkin though renaming would not be allowed

Field Name	Value
	if reversioned, checked out and checked in again.
In-Progress Versions	Specifies how to handle in-progress versions of the document once the document is versioned or promoted to the next lifecycle state: • **Delete previous non-effective version when a new version is created**: If selected, when a new version of a document is created, deletes the previous non-effective version on checkin. • **Delete previous non-effective versions upon promotion to effective**: If selected, when an existing version of a document is promoted to the next lifecycle state, deletes all previous non-effective versions. • **Delete previous non-effective versions and annotations upon promotion to effective**: If selected, when an existing version of a document is promoted to the next lifecycle state, deletes all previous non-effective versions and their annotations. • **Don't do anything**: none of the above delete operations are performed.
***Review Cycle**	DCM includes the ability to define review periods for documents, and when to initiate review notifications. The value entered determines the number days before the document needs to be reviewed. **Note:** Review Cycle and Review Period are defined as follows: • Review Cycle: specifies a time period after the effective date. This value is used to determine the next review cycle for a document.

Field Name	Value
	• Review Period: the Document Review Notification job uses the value of this attribute to determine the number of days required to start the notification. Refer to Appendix Appendix A, The DCM Data Model and Object Types for additional details.
***Days Prior to Review**	The Document Review Notification job uses this value to determine the number of days required to start the notification.
***Days After Review Date**	The Document Review Notification job uses this setting to determine the number of days required to send a notification if the business owner has not acted on the notification.
* For Example: if you specify a **Review Cycle** of 30 days and give 2 **Days Prior to Review** and 2 **Days After Review Date**, you will get a notification on day 28 and another on day 32.	

4. Click **Next**, as needed, to skip a tab or, **Previous** to return to a tab. You can pick and choose the tab(s) you want to provide entries for. Otherwise, click **Finish** to accept your changes and any default values you did not change.

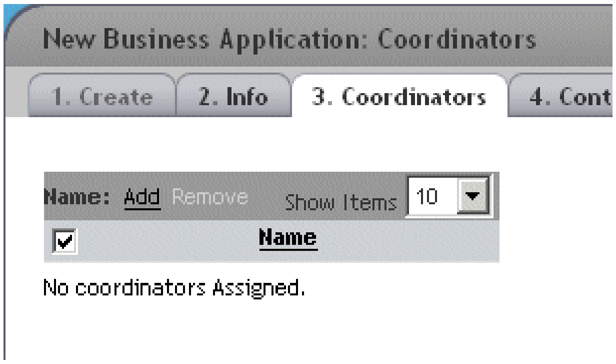
   A locator box is used in each of the remaining tabs to facilitate *adding* and *removing* the applicable value for:

   - **Coordinators**
   - **Contributors**
   - **Templates**
   - **Workflows**

   Note: Ensure items (users and groups) selected for Coordinators and Contributors have sufficient user privileges in the repository to create or delete DCM relationship types.

   The following is an example of the locator for Coordinators.

Figure 11. Locator box for Coordinators



No coordinators Assigned.

Follow these substeps to use a locator:

a. Click **Add** to display the **Choose an item** screen.

b. Select the checkbox of the item(s) you want and move it from the left pane, using the *Add button*, to the right pane then click **Ok**.

   The **Choose an item** screen disappears and the selected item(s) are displayed in the locator box. Use the *Remove button* if necessary to undo a selection before it gets added to and listed in the locator box. You can always remove an item after it gets added and listed in the locator box. The **Remove** option on the locator is available only when you select the checkbox of the item(s) to be removed.

Viewing or Modifying Business Applications

The **Business Applications** page allows you to select an existing business application, display its properties, and make changes.

To view or modify an existing business application:

1. Navigate to **Administration > DCM > Business Applications**.
2. To modify an existing business application, right-click on a business application and select **Properties**.
3. Refer to the table below for a description of the attributes.

Table 38. Business Application: Properties

Field Name	Value
Unique Name	DCM allows you to enforce unique naming of controlled documents at either the business application level or the repository level. If repository-level enforcement of unique naming has been configured, then the **Unique Naming** option will not appear on this page. Click a radio button to select whether unique names are enforced for controlled documents: • **Enforce Unique Names within Business Application** • **Do Not Enforce Unique Names**
Create	To specify what repository version to assign the initial checkin of a new document created using this document class, choose one of the following options: • **Inherit**: Use the same settings for this document class property as the associated business application. • **Always at version 0.1** • **Always at version 1.0**
Check-in Rule	To restrict the repository versioning behavior for documents of this class, choose one of the following options: • **Inherit**: Use the same settings for this document class property as the associated business application. • **Do not allow minor version checkin**: If selected, forces the user to check in documents based on this class as major versions only (1.0, 2.0, 3.0, etc.) • **Do not allow major version checkin**: If selected, forces the user to check

Field Name	Value
	in documents based on this class as minor versions only (1.1, 1.2, 1.3, etc.) • **Do not allow same version checkin**: If selected, disables the **Check In As Same Version** option in the checkin dialog, and forces the user to check in documents as incremented major or minor versions only.
Auto-Name	Specifies when an autonaming or autonumbering scheme is applied to documents of this class: • **Inherit**: Use the same settings for this document class property as the associated business application. • **Assign autoname on when document is created**: If selected, the autonaming scheme is applied immediately upon document creation • **Assign autoname on Lifecycle state**: If selected, the autonaming scheme is applied only when the document reaches a certain lifecycle state (for example, Approved). • **Do not assign autoname**: If selected, disables the system autonaming scheme for documents of this document class.
In-Progress Versions	Specifies how to handle in-progress versions of the document once the document is versioned or promoted to the next lifecycle state: • **Delete previous non-effective version when a new version is created**: If selected, when a new version of a document is created, deletes the previous non-effective version on checkin. • **Delete previous non-effective versions upon promotion to effective**: If selected, when an

Field Name	Value
	existing version of a document is promoted to the next lifecycle state, deletes all previous non-effective versions. • **Delete previous non-effective versions and annotations upon promotion to effective**: If selected, when an existing version of a document is promoted to the next lifecycle state, deletes all previous non-effective versions and their annotations. • **Don't do anything**: none of the above delete operations are performed.
***Review Cycle**	DCM includes the ability to define review periods for documents, and to automatically start review workflows. The value entered determines the number days before the document needs to be reviewed. **Note:** Review Cycle and Review Period are defined as follows: • Review Cycle: specifies a time period after the effective date. This value is used to determine the next review cycle for a document. • Review Period: the Document Review Notification job uses the value of this attribute to determine the number of days required to start the notification. Refer to Appendix Appendix A, The DCM Data Model and Object Types for additional details.
***Days Prior to Review**	The Document Review Notification job uses this value to determine the number of days required to start the notification.

Field Name	Value
***Days After Review Date**	The Document Review Notification job uses this setting to determine the number of days required to send a notification if the business owner has not acted on the notification.
* For Example: if you specify a **Review Cycle** of 30 days and give 2 **Days Prior to Review** and 2 **Day After Review Date**, you will get a notification on day 28 and another on day 32.	

4. To navigate to a particular page in the business application's set of **Properties** pages, click on one of the navigation choices on the upper left hand corner of the page.

5. To view which users, groups, and roles have permissions for this business application, click the **Permissions** tab.

6. To add a user, group, or role to the list of users and groups with permissions on this business application:

   a. Click the Add icon: ⊕.

   b. Click the checkboxes associated with the names of the users, groups, or roles you want, then click **Add**.

   c. Click **OK** when you have finished making your selections.

   d. Click a radio button to assign a **Basic Permission** level.

   e. Click one or more checkboxes to assign **Extended Permissions**.

   f. Click **OK** to save your changes.

7. To change permissions for a user, group, or role already associated to this business application:

   a. Click the checkbox associated with the user, group, or role whose permission you want to edit.

   b. Click the Edit icon: ✎.

   c. Optionally, click a radio button to change an assigned **Basic Permission** level.

   d. Optionally, click one or more checkboxes to change or add **Extended Permissions**.

   e. Click **OK** to save your changes.

8. To change the active permission set associated with this business application, click the **Select** link, select another permission set from the **Choose a permission set** page, then click **OK**.
9. To add or remove a user or group to or from the list of available Authors/Contributors for the business application, click **Add** or **Remove**.
10. To change the displayed options, select what kinds of items you want to see from the pull-down menu at the bottom of the page.
11. To select the name of a user, group or role to assign as a signatory, click the checkbox corresponding to that group or user name, then click **Add**.

    When choosing users and groups, ensure that the selected users and groups have sufficient user privileges in the repository. Otherwise, these users may not have sufficient privileges to create or delete DCM relationship types.
12. Click **OK** to save your changes and to return to the **Business Application: Authors/Contributors** page.
13. When you have finished selecting users and groups, click the **Template** navigation link in the upper left hand corner of the page to continue.
14. To add a predefined document template, click the **Add** link. .
15. Click the checkboxes corresponding to the templates you want to add, then click **Add**.
16. When you have finished selecting templates, click **OK** to save your changes and to return to the **Template** page.

    The selected templates are listed on the **Template** page.
17. When you have finished entering values, click the **Workflow** navigation link in the upper left hand corner of the page to continue.
18. To add a predefined workflow you want to associated with this business application, click , click **Add** to choose from a system-wide list of workflows.
19. Click the checkboxes corresponding to the workflows you want to add, then click **Add**.
20. When you have finished selecting templates, click **OK** to save your changes and to return to the **Workflow** page.

    The selected workflows are listed on the **Workflow** page.
21. When you have finished selecting workflows, click the **History** navigation link in the upper left hand corner of the page to continue.
22. The History page provides audit information.
23. Click **OK** to save your changes.

Deleting Business Applications

The owner of a business application can delete it from the system.

To remove a business application:

1. Navigate to **Administration > DCM > Business Applications**.
2. Click the checkbox associated with the business application you want to delete.
3. Choose **File > Delete**.

Chapter 6

Setting Up Document Classes and Document Relationships

This section describes the following:

Document Classes

Document classes provide the flexibility to create a controlled document type required for a particular business process, and to associate a set of default behavior and characteristics with that type. You can base a document class on any Documentum object type. Using the custom properties associated with a document class, you can define specific kinds of behavior, such as the document class's versioning behavior, associated autonaming scheme, and an associated lifecycle.

For example, you can set up document classes for any kind of controlled documents commonly used by your organization, such as change notices, change requests, Standard Operating Procedures (SOPs), or plant drawings.

You can associate relationship types (such as Closes or Supports) to determine how a document class treats attached or associated documents. For example, you can define the following things:

- How do documents belonging to this document class handle promotion?
- How do documents belonging to this document class handle signoff?

You can create multiple document classes on a single object type.

Each business process can have one or more document classes. Each document class can point to any Documentum object type and its behavior is defined by several custom properties such as its default lifecycle and relationship types.

If you choose not to use the predefined document classes provided with DCM, you must create the controlled document types, change request, and change notice document classes you want DCM to manage.

To use these predefined controlled document types, choose one of the following options from the Filter pull-down menu when creating a document class:

- **Controlled Document types**
- **Change Request types**
- **Change Notice types**

Creating a Document Class

Document classes are administered by the document class owner, who is responsible for creating and maintaining one or more document classes. To create a document class, you must have the role of business application owner or document class owner.

To create a document class:

1. Navigate to **Administration > DCM > Document Classes**.
2. To create a new document class, choose **File > New > Document Class**.
3. Refer to the table below for a description of the attributes.

Table 39. New Document Class: Create Tab Fields

Field Name	Value
Name	Enter a unique name for your document class.

Field Name	Value
Business Application	From the pull-down menu, choose the name of the business application you want to associate with this document class. After creating a document class, you cannot change its associated business class.
Controlled Documents Filter	From the pull-down menu, choose the class of the object type you want to associate with this document class. The options you see for **Docbase Type** on the following screen will depend on the choice you make here.

4. When you have finished entering values, click **Next** to continue.

5. Enter values in the fields of the Info tab:

Table 40. Document Class Properties: Info Tab

Field Name	Value
Name	This is the value you provided in the Create tab.
Business Application	This field displays the associated business application you selected on the Create tab. You can modify this value.
Filter	This field displays the Controlled Document Type you selected from the Create tab.
DC Owner	By default, the name of the user creating the document class appears in this field. If you are creating this page as the business application owner, type the name of a document class owner.

Field Name	Value
Object Type	From the pull-down menu, choose the name of a repository type to associate with this document class. The options that appear on this list vary, depending on what value you chose for the **Filter** field on the previous page.
Lifecycle	From the pull-down menu, choose the name of a predefined lifecycle to associate with this document class. Make sure members in Contributor roles have Relate permission to the lifecycle. Contributor will otherwise not be able to convert a document to a controlled document of this document class.
Alias Set	From the pull-down menu, choose the name of a predefined alias.
Create	To specify what repository version to assign the initial checkin of a new document created using this document class, choose one of the following options: • **Inherit**: Use the same settings for this document class property as the associated business application. If the **Inherit** option is disabled, it means that there are no values for this field to inherit from the associated business application. • **Always at version 0.1** • **Always at version 1.0**
Check-in Rule	To restrict the repository versioning behavior for documents of this class, choose one of the following options: • **Inherit**: Use the same settings for this document class property as the associated business application. If the **Inherit** option is disabled, it means that there are no values for this

Field Name	Value
	field to inherit from the associated business application. • **Do not allow minor version checkin**: If selected, forces the user to check in documents based on this class as major versions only (1.0, 2.0, 3.0, etc.) • **Do not allow major version checkin**: If selected, forces the user to check in documents based on this class as minor versions only (1.1, 1.2, 1.3, etc.) • **Do not allow same version checkin**: If selected, disables the **Check In As Same Version** option in the checkin dialog, and forces the user to check in documents as incremented major or minor versions only.

6. When you have finished entering values, click **Next** to continue.

7. Choose the Coordinators for this document class:

   - If the **Inherit** checkbox is selected, all subsequent tabs of the New Document Class dialog box inherit the values you provided in this tab.

8. When you have finished entering values, click **Next** to continue.

9. Select a **Name** from the **Contributors** tab and click **Next**.

10. Select a **Template** from the **Template** tab and click **Next**.

11. Select a **Workflow** from the **Workflow** tab and click **Finish**.

Viewing or Modifying Document Class Properties

To view or modify the properties defined for a document class:

1. Navigate to **Administration > DCM > Document Classes**.

2. To view or modify the properties for an existing document class, right_click on a document class and select **Properties**.

3. View or modify values in the fields provided, refer to Table 40, page 129 for details.

4. When you have finished entering values, click **OK** to save your changes.

5. To view which users, groups, and roles have permissions for this document class, click the **Permissions** tab.
6. To add a user, group, or role to the list of users and groups with permissions on this document class:
   a. Click the Add icon: ⊕.
   b. Click the checkboxes associated with the names of the users, groups, or roles you want, then click **Add**.
   c. Click **OK** when you have finished making your selections.
   d. Click a radio button to assign a **Basic Permission** level.
   e. Click one or more checkboxes to assign **Extended Permissions**.
   f. Click **OK** to save your changes.
7. To change permissions for a user, group, or role already associated to this document class:
   a. Click the checkbox associated with the user, group, or role whose permission you want to edit.
   b. Click the Edit icon: ✎.
   c. Optionally, click a radio button to change an assigned **Basic Permission** level.
   d. Optionally, click one or more checkboxes to change or add **Extended Permissions**.
   e. Click **OK** to save your changes.
8. To change the active permission set associated with this document class, click the **Select** link, select another permission set from the **Choose a permission set** page, then click **OK**.
9. To view or modify the users and groups designated as authors/contributors for this document class, click the **Authors/Contributors** tab.
10. Choose the users and groups you want to define as authors for this document class:
    - If the **Inherit** checkbox is selected, you see a predefined list of users and groups. To select one or more of the displayed users or groups, click the checkbox associated with the name.
    - If the **Inherit** checkbox is not checked, click **Add** to choose from a system-wide list of users and groups.

    When choosing users and groups, ensure that the selected users and groups have sufficient user privileges in the repository. Otherwise, these may not have sufficient privileges to create or delete DCM relationship types.

    When creating controlled documents, only the selected users and groups will see this document class in a drop-down list of available document classes.

11. When you have finished entering values, click **OK** to save your changes.
12. To view or modify the document templates associated with this document class, click the **Templates** tab.
13. To select a predefined document template, click the checkbox associated with the template name, then click **Next** to continue.
14. When you have finished entering values, click **OK** to save your changes.
15. To view or modify the workflows associated with this document class, click the **Workflows** tab.
16. Choose the workflow you want to associated with this document class:
    - If the **Inherit** checkbox is selected, you see a predefined list of workflows. To select a workflow, click the checkbox associated with the name.
    - If the **Inherit** checkbox is not checked, click **Add** to choose from a system-wide list of workflows.
17. To view a Document History report for the document class, click the **History** tab.

    This report displays complete history information for a specific document version or for all versions of a document. The Document History report displays the following information:

    - Event name
    - Version number
    - User name who performed the action
    - Date performed
18. To view the history of a specific document version, choose the version from the pull-down menu in the upper right-hand corner of the History page.
19. Click **OK** to save your changes to this document class.

Deleting Document Classes

The owner of a document class can delete it from the system if the document class is not currently being referenced by any controlled document. Before deleting a document class, you must delete all of the controlled documents associated with the class.

To remove a document class from your repository:

1. Navigate to **Administration > DCM > Document Classes**.
2. Click on **Document Classes**.
3. Click the checkbox associated with the document class you want to delete.

4. Choose **File > Delete**.

About Document Relationships

Document relationships define dependencies between documents. Relationships between documents are linked to specific versions.

Only administrators or super users can create new relationship types. Users, other than administrators or super users, can still see the Relationship Types node under the Administration node but cannot create or modify existing relationship types.

When a document is signed off or promoted, DCM looks up the relationships defined for that document to decide whether the document will be signed off or promoted separately or as part of a group.

For example, if a change request is issued against an SOP, then the change request has a relationship with the SOP. Relation types are unique to specific document types.

Relationships in DCM are useful if you want to:

- tie content together in related documents
- use a Process relationship type to associate documents with change requests
- manage change sets (that is, signoff groups of content as a whole) using change notices and their related documents

You can define the following kinds of relationships:

- Process

  This relationship references a controlled document with processing options to be performed on the controlled document.

  This relationship uses process rules to define what processing to perform on a child documents with the parent document. Currently, you can define two kinds of process rules

  — Promotion Rule

  A Promotion rule defines the behavior of promoting a document.

  — Signoff Rule

  A Signoff rule defines the behavior of signing off a document.

- Reference

  This relationship references a controlled document.

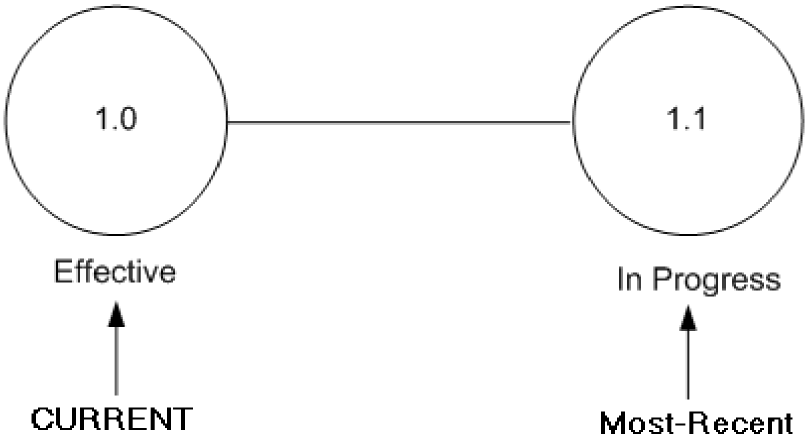
- Supports

  This relationship references a non-controlled document.

Binding rules for each relationship type are as follows:

- Reference relationships are late bound - always point to the "Most-Recent" version of the child.
- Supports and Against (Change Request) relationships are Early bound - always point to a specific version.
- Process relationships are latest bound until Approved (or Effective if not Approved) - always point to the latest version of the document until it is in the approved state and then it points to a specific version like early bound.

Figure 12. Effective versus Latest Versioning



Optionally, you can install the following predefined relationships when you install DCM:

- Process
- Against
- Reference
- Supports

Supporting Relationships and Uncontrolled Objects

Uncontrolled objects are objects that are not managed in a controlled environment. The content may be a supporting document, email, or other documents. Uncontrolled documents are documents of type dm_document with no equivalent document class.

When these uncontrolled documents are related to a document class, they are usually referred to as *supporting* or *reference* documents. By default, DCM associates a Reference relationship with uncontrolled objects. Since the uncontrolled document is only attached as a reference, the relationship rule has a value of “n/a”. You can rename the Reference relationship to Supporting relationship, but you cannot delete or add rules to the Supporting relationship.

Attachment Rules

Attachment rules define the document classes and lifecycle states for attaching this relationship. All relationship types can define both parent and child attachment rules, but the Supports relation type cannot define child attachment rules, since the child is a non-controlled document.

The attachment state type indicates the DCM state types of a controlled document. The parent attachment state type indicates the specific state types of the parent document for which you can create this relationship. Only parent documents in these states will be able to use this relationship. Likewise, the child attachment state type indicates the specific state types of the child documents that may be attached using this relationship. Only documents in these states may be attached to the parent using this relationship.

Relationships for Signed Off and Effective Documents

Any time you check in a new version of an Signed Off or Effective document, the associated relationships for that document are removed for the new version. This is because you don't want to automatically promote or demote these kinds of documents, especially since the automatic promotion for an Effective document is to the Obsolete state, no matter what interim states your lifecycle defines for that document class.

Creating a Relationship Type

To create a relationship, you must log in to DCM as a user with a role of business application owner or document class owner.

Only administrators or super users can create a new relationship type. Users, other than administrators or super users can still see the Relationship Types node but cannot create or modify existing relationship types.

To create a document relationship:

1. Navigate to **Administration > DCM > Relationship Types**.
2. To create a relationship type, choose **File > New > Relationship Type**.
3. Type a name for the new relationship type, then click **Next**.
4. Refer to the table below for a description of the attributes.

Table 41. New Relationship Type: Info

Field Name	Value
Description	Type a brief description of this relationship type.
Relation Type	From the pull-down menu, choose one of the following options: • *Process*: this type defines processing that needs to be performed on a **controlled document**. • *Against*: this type defines a reference to a **controlled document**. This reference always refers to the original version of a document when it was attached. • *Reference*: this type defines a reference to a **controlled document**. This reference always refers to the current version. • *Supports*: this type defines a reference to a **non-controlled document**.
Parent	From the pull-down menu, choose the name of the document class you want to designate as a parent class. **Note:** Users can specify a Business Application as a Parent.
Parent Lifecycle State Type	This setting determines in which lifecycle states the parent document class specified above can have children documents. To choose all lifecycle states, click the **All** radio button. To choose one or more specific lifecycle states, click the **Specified** radio button, then click the **Edit** hyperlink to select from a list of available lifecycle states.

Field Name	Value
Child Document Class	This setting allows you to define which document classes are designated as children to the parent document class specified above. To choose all available document classes, click the **All** radio button. To choose one or more specific classes, click the **Specified** radio button, then click the **Edit** hyperlink to select from a list of available document classes.
Child Lifecycle State Type	This setting determines in which lifecycle states the children documents must be in before they can be attached to the parent document. To choose all lifecycle states, click the **All** radio button. To choose one or more specific lifecycle states, click the **Specified** radio button, then click the **Edit** hyperlink to select from a list of available lifecycle states.
Promote	This option is not available unless you chose Process as your relation type. If you chose Process as your relation type, click a radio button to select one of the following options to define how DCM should handle lifecycle promotions: • **Just the parent**: Only the parent document is promoted. • **Group promotion (parent and child)**: When the parent document gets promoted, by default, child documents will be promoted, too. **Note:** If you use a process relationship type and specify a parent document class that is a Business Application, the check boxes for Group Signoff, Group Promotion, and Aggregate are disabled.

Field Name	Value
	You need to know the lifecycle, defined on the Document Class, before you can specify rules.
Signoff	This option is not available unless you chose Process as your relation type. If you chose Process as your relation type, click a radio button to select one of the following options to define how DCM should handle document signoffs: • **Just the parent**: Only the parent document is signed off. • **Group Signoff (parent and child)**: When the parent document gets signed off, by default, child documents will be signed off, too.

5. To save your changes and create the relationship type, click **Finish**.

Viewing or Modifying Relationship Types

To modify an existing relationship, you must log in to DCM as a user with a role of business application owner or document class owner.

To modify or delete a document relationship:

1. Navigate to **Administration > DCM > Relationship Types**.
2. To view or modify the properties for an existing relationship type, right- click on a relationship type and select **Properties**.
3. You can view or modify values in the fields provided, refer to Table 41, page 137 for details.
4. When you have finished viewing or modifying properties, click **OK** to save your changes.

Deleting Relationship Types

The owner of a relationship type can delete it from the system.

To remove a relationship type from your repository:

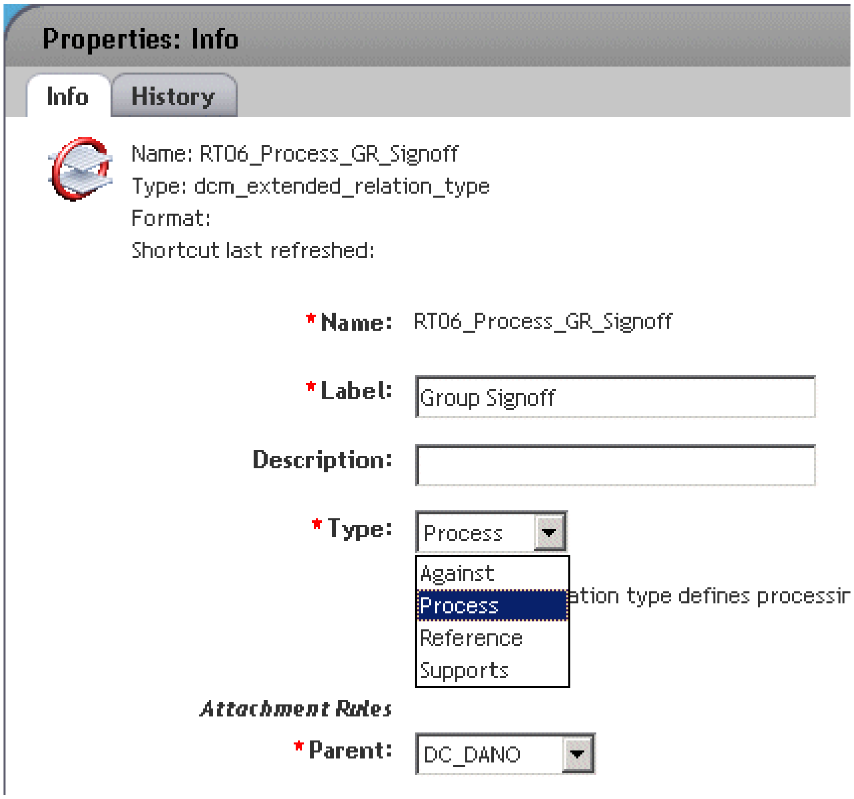
1. Navigate to **Administration > DCM > Relationship Types**.
2. Click the checkbox associated with the relationship type you want to delete.
3. Choose **File > Delete**.

Processing Relationships

Parent and child document relationships are associated and processed by Attachment Rules and Process Rules. Such relationships must be in a lifecycle and in association with a document class. Process Rules for your document Relationship Types are set in association with document classes. Processing rules for your document relationships are set for specific lifecycle states which are associated with document classes, not business applications. A document class must be associated with at least one business application template, or multiple business application templates. If the parent is a business application, you cannot set up **Process Rules**. Business applications do not specify a lifecycle—a lifecycle is required to set up Process Rules.

Note: Two mandatory fields: **Type** and **Parent** of the Properties Info tab must be specifically set to enable Process Rules. In the **Type** field you must select **Process**, any other selection disables Process Rules. To enable Process Rules for your document relationships you must select an item in the drop down list, of the **Parent** field under Attachment Rules, that is of a **document class**. If the item you select from the drop down list of the Parent field disables the Process Rules, the selected item is not of a document class.

Figure 13. Relationship Properties Info Tab — Type Set to Process



Name: RT06_Process_GR_Signoff
Type: dcm_extended_relation_type
Format:
Shortcut last refreshed:

*Name: RT06_Process_GR_Signoff

*Label: Group Signoff

Description:

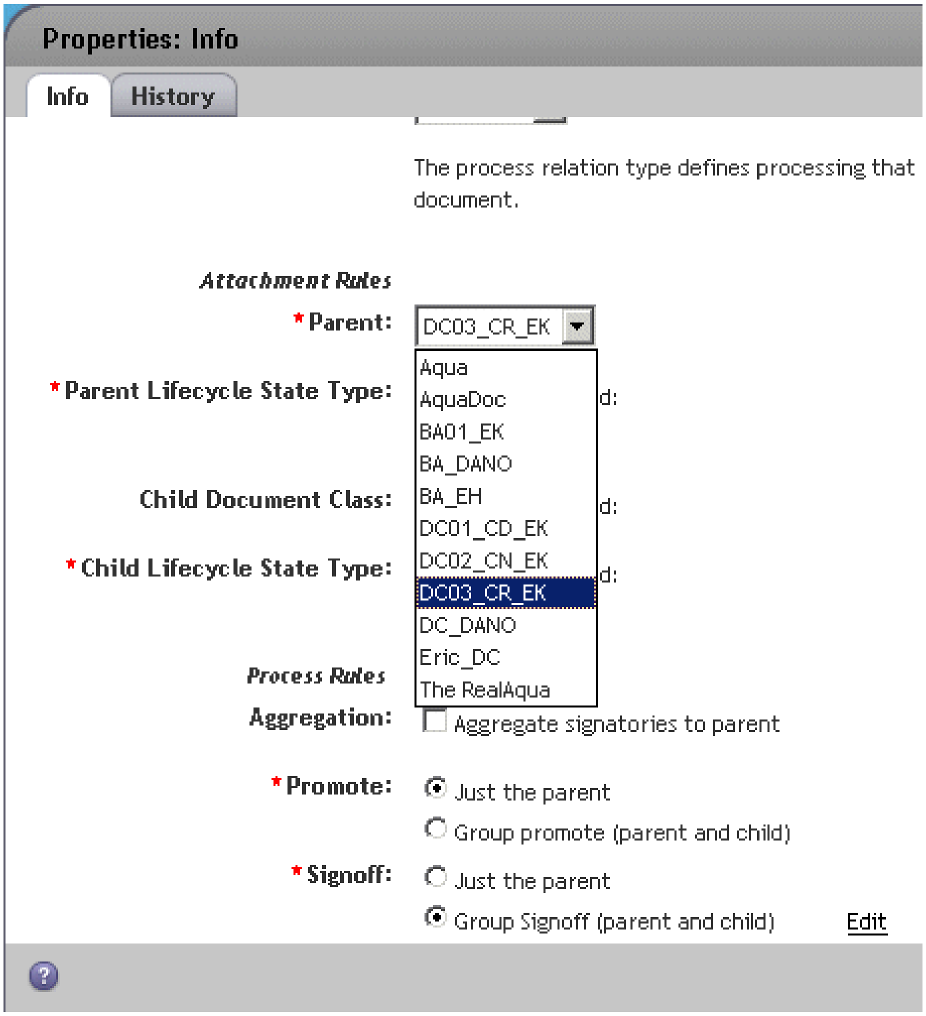
*Type: Process

Attachment Rules

*Parent: DC_DANO

The relationship properties **Parent** field must have a document class type item selected to enable Process Rules. Any other selection disables the Process Rules.

Figure 14. Relationship Properties Info Tab — Selected Parent Item Must be of a Document Class



The process relation type defines processing that document.

Attachment Rules

*Parent: DC03_CR_EK

*Parent Lifecycle State Type: d:

Child Document Class: d:

*Child Lifecycle State Type: d:

Process Rules

Aggregation: □ Aggregate signatories to parent

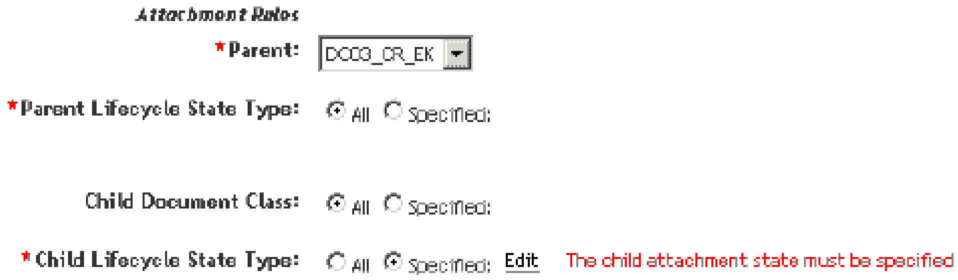
*Promote: ● Just the parent ○ Group promote (parent and child)

*Signoff: ○ Just the parent ● Group Signoff (parent and child)

You can specify a different lifecycle state type for child documents and have child documents signed off differently from the parent document. The Attachment Rules as they relate to the Parent document may be specified differently for child documents. You can have a child document associated with a different document class as well as a different lifecycle state from the parent document. Although it is optional to specify a Child Document Class, it is mandatory to specify a Child Lifecycle State Type.

You can **Edit** the **Attachment Rules** when you choose to specify the available options. Selecting **Specified** allows you to edit the selected option. There is nothing to edit when you select **All**.

Figure 15. Attachment Rules Showing Option Used to Edit the Child Lifecycle State Type



Attachment Rules

*Parent: DCO3_CR_EK

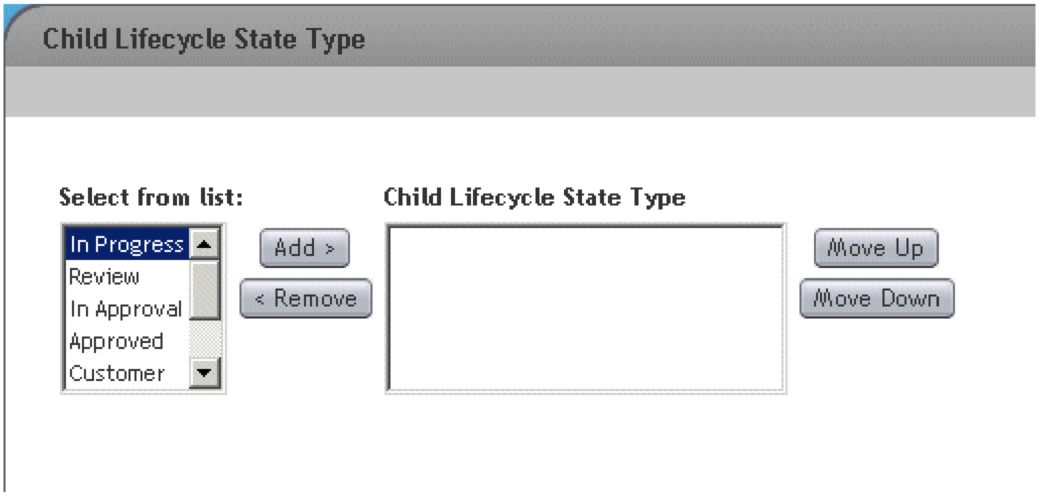
*Parent Lifecycle State Type: ● All ○ Specified:

Child Document Class: ● All ○ Specified:

*Child Lifecycle State Type: ○ All ● Specified: Edit The child attachment state must be specified

The **Child Lifecycle State Type** page is displayed when you click **Edit** to specify a different lifecycle state for the child document.

Figure 16. Editing Child Lifecycle State Type



Child Lifecycle State Type

Select from list:

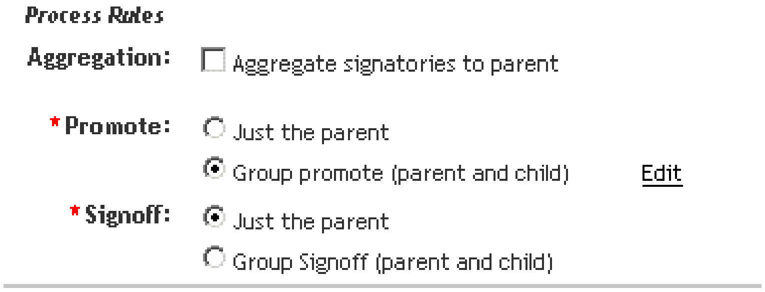
- In Progress
- Review
- In Approval
- Approved
- Customer

Child Lifecycle State Type

Similarly, you can **Edit** the **Process Rules** when you choose to specify the available options.

Selecting **Group Promote** or **Group Signoff** allows you to edit the selected option. There is nothing to edit when you select **Just the Parent**.

Figure 17. Editing Process Rules



Process Rules

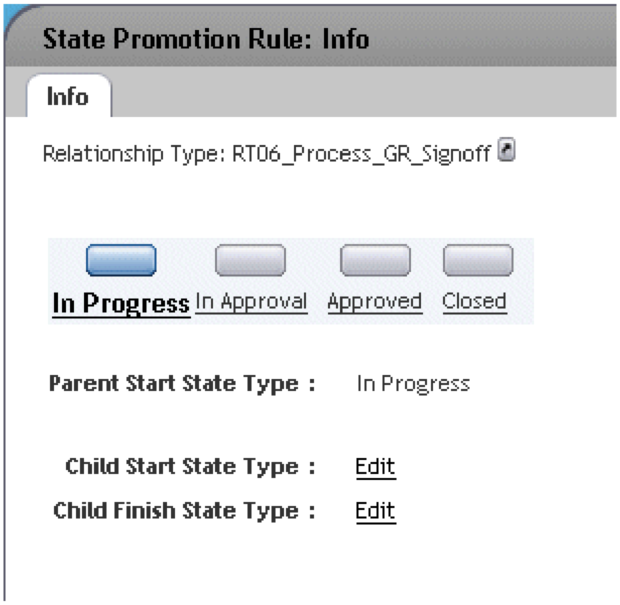
Aggregation: □ Aggregate signatories to parent

*Promote: ○ Just the parent
● Group promote (parent and child) Edit

*Signoff: ● Just the parent
○ Group Signoff (parent and child)

The **State Promotion Rule: Info** page is displayed when you click **Edit** to specify a different lifecycle state for the child document to start and finish in.

Figure 18. State Promotion Rule Info Tab—Child Start and Finish State Types



State Promotion Rule: Info

Info

Relationship Type: RT06_Process_GR_Signoff

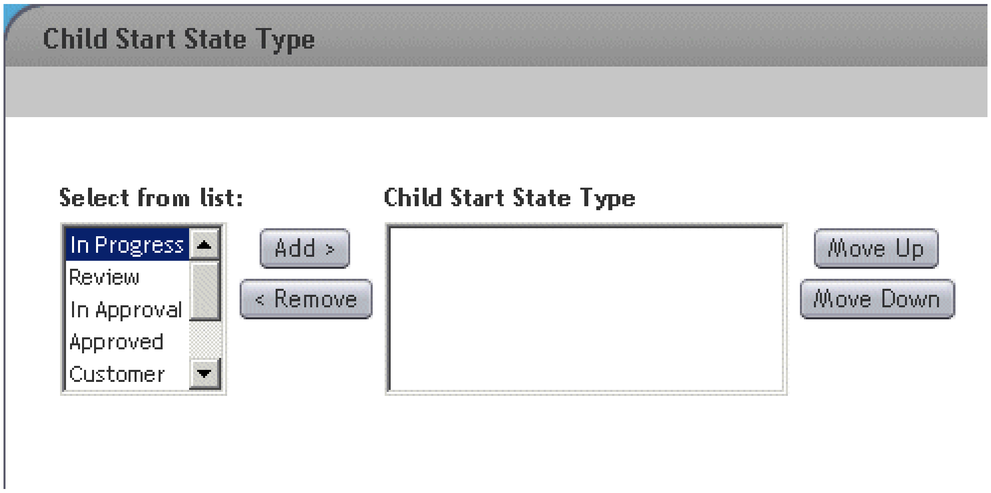
Parent Start State Type : In Progress

Child Start State Type : Edit

Child Finish State Type : Edit

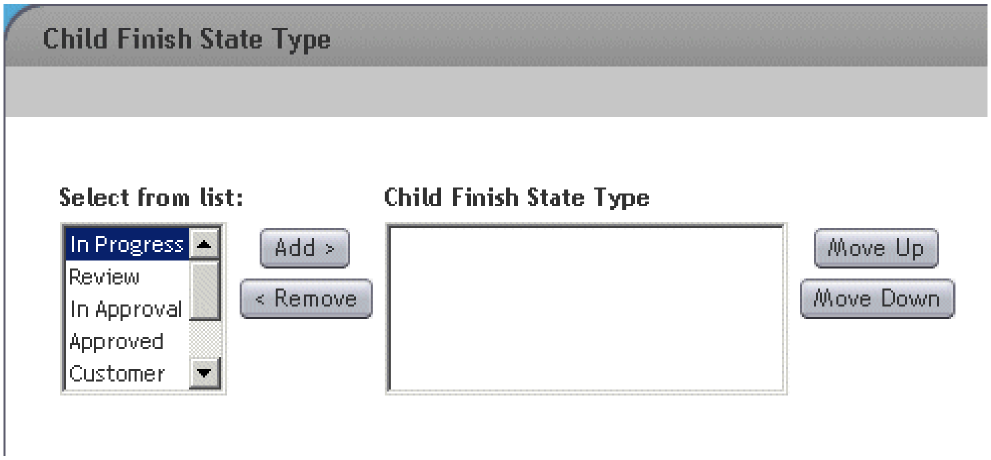
The **Child Start State Type** page is displayed when you click **Edit** to specify a different lifecycle state for the child document to start in.

Figure 19. Specifying Child Start State Type



The **Child Finish State Type** page is displayed when you click **Edit** to specify a different lifecycle state for the child document to finish in.

Figure 20. Specifying Child Finish State Type



Option settings in the Process Rules affect your selections in the State Promotion Rule Info tab as follows:

Aggregation Disabled (in which case the Aggregation checkbox is empty)

- When you choose to Edit the **Group Promote** field, you can edit both start and finish state types.

- When you choose to Edit the **Group Signoff** field, you can edit only the Child Start State Type.

Note: Aggregation is used to align child and parent documents for signoff. If aggregation is not used, child documents may be signed off independently from the parent. For Example, there are signatories A and B assigned to the parent document. Signatory C is assigned to the child document. When Aggregation is enabled, all signatories A, B, and C are assumed by the parent. When Aggregation is disabled, signatories remain with their respective documents and are not aggregated into the parent document, A and B signoff the Parent document and C signs off the child document.

Aggregation Enabled (in which case the Aggregation checkbox is checked)

- When you choose to Edit the **Group Promote** field, a drop down list is provided to specify the Child Finish State Type.
- When you choose to Edit the **Group Signoff** field, you can edit only the Child Start State Type.

Chapter 7

Lifecycle Extensions

A lifecycle extension listed under Lifecycle Extensions is displayed when a document class is created for a controlled document. All of the attributes for a particular document class are edited from the Properties of its lifecycle extension. The attributes for each state in the lifecycle can be modified. For example, you can modify, edit, select, and deselect, the following settings on the **Info** page/tab:

- Promote after last signoff
- Send to be read notifications
- Use autoname scheme
- Allow manual promotion
- Allow demote to previous state
- Manifest e-signature
- Signoff Code
- Signoff Confirmation Text
- Reject Code
- Reject Confirmation Text
- Notification Text
- TBR Notification Text

You can also modify entries on the other tabs next to the Info tab, that is add and remove entries for the **Signatories**, **Distribution List**, **Workflows**, and **Overrides** tabs. Procedures used to modify lifecycle extensions are available in the *Documentum Compliance Manager User Guide*, version 6.5.

DCM allows you to add or modify information about the mandatory and optional signatories, and to-be-read (TBR) groups, roles, and users values for each state defined in a DCM or customized lifecycle. These values are stored as properties in an instance of a repository object, named dcm_state_extension. You can use these properties to specify that your system perform specific tasks, such as applying an autonaming or autonumbering scheme to the document, during particular document lifecycle states.

For the lifecycle extension GUI, the state type is read-only and is read from the lifecycle state. If the lifecycle state changes, you are warned to save changes to allow for updating to occur. Users have the ability to specify that e-signatures can be employed for specific states. Users also have the ability to demote for a specific state (disabled based on lifecycle and state type).

Chapter 8

Creating and Editing Controlled Document Lifecycles

This chapter describes the task sequence for creating a document lifecycle (DLC) for use with DCM, and provides step-by-step instructions for creating DLCs using the Documentum Lifecycle Editor included with Documentum Application Builder. This chapter also explains how to edit extended properties for lifecycle states using the DCM Administration screens.

Topics discussed in this chapter include:

DLCs, Workflows, and Business Processes

Many companies have policies that govern how specific types of documents work their way through their useful lives, and beyond. Such policies typically specify the stages that a document passes through and the activities that occur at each stage. Documentum separates these aspects of document policy into two concepts: *lifecycles*, which define the stages of a document's life, and *workflows*, which define sequences of actions that users must perform on the document.

Document lifecycles and workflows complement one another. The DLC does not define what activities happen to a document while it resides in a state, who is responsible for those activities, or when the document should move to another state. A workflow defines a connected set of activities, including who performs the activities and when. For more information about Workflow Manager, see its online help and the *Content Server Fundamentals Guide*.

You can simulate a document lifecycle in a workflow definition, but keeping the two concepts separate leads to simple, reusable designs. This is because the simulated lifecycle applies only within that workflow, not in any other workflow. In addition, the lifecycle aspects of the workflow can become entangled with a potentially complex network of activities.

You can use a workflow to drive the activities associated with a state in a document lifecycle. For example, you can create a workflow to route a document to several reviewers, then use a procedure associated with the appropriate state of the lifecycle to launch that workflow.

States and Transitions

This section describes both the structure of a DLC as a succession of *states*, which model the life stages of the document, and the rules for transitions between those states. It provides details of how to design and specify the elements that define states and control transitions.

How DLC States Model Document Life Stages

Documents have natural life stages. A product specification, for example, might begin as an engineer's draft, pass through review and signoff, and finish as a controlled reference document. While all product specs might pass through approximately the same stages, a vacation request or an expense report passes through different stages.

A DLC models a document's life stages as a set of possible states and rules that govern transitions from one state to another. The states fall into two categories: *normal* and *exception*. Normal states follow a linear progression, from the first (*base*) state to the last (*terminal*) state.

Exception states model the case when a document must go into limbo. An unanticipated temporary failure of a key piece of equipment may force you to suspend a released standard operating procedure (SOP). The SOP may not need to change—when the equipment comes back on line, the SOP goes back into force.

Each normal state has either zero or one exception state into which documents can move from that normal state. An exception state can serve more than one normal state.

Document Lifecycle State Diagram

The main window of the DLC editor represents the formal structure of a DLC in a diagram called a lifecycle state diagram. Each state appears as a pentagon with a state name label. Arrows represent possible state transitions in the life of a document that the DLC controls.

Each normal state in the diagram has at most one arrow leading to it from a normal state and at most one arrow leading from it to a normal state. Exactly one normal state, called the base state, has no arrow leading to it from a normal state. Exactly one normal state, called the final state, has no arrow leading from it to a normal state. That is, the normal states form a non-branching linear path from the base state to the final state:

Base state –> . . . –> Final state

A normal state has at most one arrow leading from it to an exception state. An exception state has one or more arrows leading to it. Different normal states can use the same exception state.

Transitions

A transition is a state change. There are five kinds of transition: attachment, promotion, demotion, suspension, and resumption.

The system carries out transition in two parts. The first part is a transaction. If it fails, nothing changes. Failures in the second part, called the *post-change procedure*, do not prevent the state change. Transitions have the following form:

```
Determine target state
If transition is not allowed
  abort the transition
Transaction
{
  Evaluate entry criteria and entry criteria procedure
    If entry criteria are not met or procedure returns FALSE
      abort the transition and back out changes
  Perform state actions and action procedure
    If an action fails or action procedure returns FALSE
      abort the transition and back out changes
 }
Perform post-change procedure
 (failures do not abort the transition)
```

The DLC state editor allows you to view or modify the properties of a DLC state. Each state has properties in the following categories, corresponding to the tabs of the editor:

- General – The state's name and its transition restrictions.
- Entry criteria – Attribute tests to qualify the object for entry into the state, and a procedure to execute upon entry.
- Actions – One or more operations from a predefined set (for example, request a PDF rendition) to carry out upon entry, and a procedure to execute after the operations.
- Post-change Procedure – A procedure to execute after the transition transaction is complete.
- Attribute property changes – Modify an attribute's label, help text, comment, and whether the attribute is read only, required, cannot be blank, hidden, or can be modified on immutable objects in this state. The required and cannot be blank properties are checked when the client application validates an object (which typically occurs on saving or checking in an object)--not when it enters the state.
- Functionality changes – Changes, only effective in this state, from the usual functionality for objects of this type.

Attachment

When you define a document lifecycle, you specify a primary object type to which it applies. You can also specify which subtypes of the primary type it applies to. A document lifecycle resides in a repository as a dm_policy object.

Attachment is the process of associating a document with a lifecycle and setting document-specific values for all of the lifecycle's aliases. When you develop a document lifecycle, you specify which states allow attachment.

When you attach a document to a lifecycle you can specify a state for it to start in. This does not need to be the base state.

Promotion and Demotion

Moving from a normal state to a later one is called *promotion*. Moving from a normal state to an earlier one is called *demotion*.

A document in a normal state can be promoted only to the next state. It can be demoted only to the previous state or to the base state.

Suspension and Resumption

Moving from a normal state to an exception state is called *suspension*. Moving from an exception state to a normal state is called *resumption*. Suspension moves a document into its state's designated exception state. Resumption moves a document either to the normal state from which it was suspended or to the base state.

Returning Versioned or Saved Objects to the Base State

When an object is checked in or saved, you have the option of returning the object to the base state or allowing it to remain in its current state. (For check in operations, the affected object is the new version.) To set this option, check the Demote document to base state on checkin/save option in the DLC Editor (or set a state's return_to_base attribute to TRUE).

Note: To avoid possible lifecycle attachment problems, it is not recommended to have the "Save as new" option selected if the document owner name is changed and the document is moved to the base state.

The object is tested against the base state's entry criteria and if it fails, the checkin or save method fails. If it passes the entry criteria, the checkin or save succeeds and the state's actions are executed. If the actions do not succeed, the object remains in the base state and the checkin or save is not backed out.

Attribute Constraints

The definition of an object type includes constraints on the possible values of the attributes of objects of that type. Subtypes of the type inherit those constraints and can add others. The definition of a subtype cannot relax the constraints inherited from the supertype.

When you design a DLC state, you can specify that while a document is in that state, its attributes must satisfy define additional constraints, beside those defined for the given type or subtype. The additional constraints no longer apply when the document leaves that state.

Functionality

Users manipulate documents through client programs. The client program matches specific capabilities—like checking in a document or moving it—with executable components designed to implement those capabilities.

Desktop matches capabilities and components dynamically. It uses the functionality of the component, the user's client capability (consumer, contributor, coordinator, or administrator), and the document's type and DLC state to make the association.

The definition of an object type includes a set of *functionality classes*, which are symbolic names for capabilities. The functionality of an object type is a set of associations between functionality classes and specific components that implement that functionality. An object type in a typical DAR inherits most of its functionality classes from its supertype and uses components of the default DAR to implement most of that functionality. For each functionality class the object type definition includes:

- The permissions required to apply the functionality to a document.
- The identity of a binary component to implement the functionality.

When you design a DLC, you can provide different capabilities, permissions, and binary components for each state. You can make the following changes to capabilities:

- Remove functionality classes that appear in the object type definition
- Reassign a functionality class to a different component
- Add new functionality classes and implementations

For each association of a functionality class and a component, you can specify the permissions a user must have to invoke the functionality.

Aliases

Aliases are the parameters of a document lifecycle. They enable you to assign symbolic names to aspects of the DLC that you expect to differ from one context to the next.

When you define a document lifecycle you can assign aliases to such instance-specific items as user and group names, permission sets, and repository locations. For example, you might specify `%SpecRepository` rather than `Engineering\Documentum 4i Project\Product Specs` for the target location of a move action.

When the system attaches a document to a lifecycle, it must provide specific values for all aliases. Alias sets facilitate this process.

Alias Sets

An *alias set* contains aliases. When you create a DLC, you can specify several alias sets (including a default alias set) for it. One of these alias sets can be assigned to resolve aliases when an object is attached to the DLC. If you did not specify an alias set for the DLC, the system uses its scope to resolve the aliases.

Using Scopes to Resolve Aliases

The *scope* of an alias is the context in which the binding of a value to its name takes place. The system recognizes four scopes: *DLC*, *session*, *user*, and *system*.

At runtime the system's ResolveAlias method examines the alias sets associated with each scope, starting with DLC, then session, user, and system. If it finds one that contains the given alias name, it uses the corresponding value.

If you know at design time the name of the alias set containing an alias you want resolved at runtime, you can increase performance by specifying the name as an argument to the ResolveAlias method. The system needs only to search the specified scope.

Document Lifecycle Scope

Using Application Builder, you can associate one or more alias sets with a DLC. When you attach the DLC to a document, the system looks first in the alias sets to resolve all aliases.

Note: Users with at least Write permission for the lifecycle object can associate alias sets with it. Only the owner or a system administrator can modify an alias set. No one can modify an alias set that is referred to by an installed lifecycle.

Session Scope

Use a session config object to specify values for a server session.

Table 42. Alias Set Attributes in Session Config Object

Attribute Name	Datatype	Repeating	Description
alias_set	DM_STRING(32)	Single	Alias set for the session

User Scope

Use a dm_user object to specify values for a Documentum user.

Table 43. User Scope Attributes in dm_user

Attribute Name	Datatype	Repeating	Description
alias_set_id	DM_ID	Single	User's default alias set.
_alias_set	DM_STRING(32)	Single	Label for the user's default alias set. The system computes this attribute.

System Scope

Use a dm_server_config object to specify values for the system running the server.

Table 44. System Scope Attributes in dm_server_config

Attribute Name	Datatype	Repeating	Description
alias_set_ID	DM_ID	Single	Default alias set for the system running the server.
_alias_set	DM_STRING(32)	Single	The label for the system's default alias set. The system computes this attribute.

Versions

Both documents and DLCs can have versions. When you attach a document to a DLC, you attach a specific version of the document to a specific version of the DLC. You can attach different versions of a document to different DLCs. When you upgrade a DLC to a new version, existing documents remain attached to the old version of the DLC.

Newly created documents of the DLC's designated object type have the new version as their default DLC.

When you design a DLC, you can specify that the document returns to the base state whenever anyone checks it in. This is the default behavior if you design the DLC with Application Builder, but you can override it.

When this option is in effect, the system attaches the newly checked in Document, which is now a new version, to the latest version of the DLC.

DCM Lifecycles

A lifecycle defines the different stages a document goes through as it is created, edited, approved, and, eventually, retired. For example, an employee might create a new human resources form, another employee might review it and return it for revision, and a third employee might give the signoff necessary to make the file available to all employees. The lifecycle defines which stage the file is in at each point in the process.

Example 8-1. How a lifecycle works

In this example, an insurance company is developing a new claim form. The claim form is developed in the following stages:

- Drafted by an attorney
- Reviewed by claims adjusters and customer service representatives
- Approved for use by a manager
- Placed in use
- Revised by an attorney to conform to changes in the law or in internal procedures
- Reapproved for use by a manager
- Retired upon becoming obsolete

Administrators design lifecycles. In a lifecycle, the first state is called the *base state*. The lifecycle moves through each state in single step increments called *step states*, either forward or backward to complete the cycle. The last state is called the *terminal state*. The claim form could have states called Draft, Reviewed, Approved, In Use, Revised, Reapproved, and Retired. You would move the file through the states as its status changed over time.

You can configure document classes or business applications to automatically associate a DCM lifecycle with a controlled document when that document is created or imported.

A typical DCM lifecycle should consist of an initial state such as In Progress or Draft. During this initial state, the document may be undergoing several iterations. The subsequent state is usually the In Approval state, followed by the Approved state. In DCM, a document is considered the main document if it is in an effective state.

About Defining Controlled Document Lifecycle States

A lifecycle can only be associated with a document class using the DCM GUI. The DCM GUI does not allow a lifecycle to be associated with a business application.

You select the lifecycle when creating or modifying a document class.

If you edit a lifecycle associated with a document class and check in the modified lifecycle to the repository as the same version, you must verify and reinstall the lifecycle so that it will be available once again to the document class or business application.

If you do not define a state type for all of the individual states comprising the custom lifecycle, DCM cannot properly access the status of objects that use the custom lifecycle.

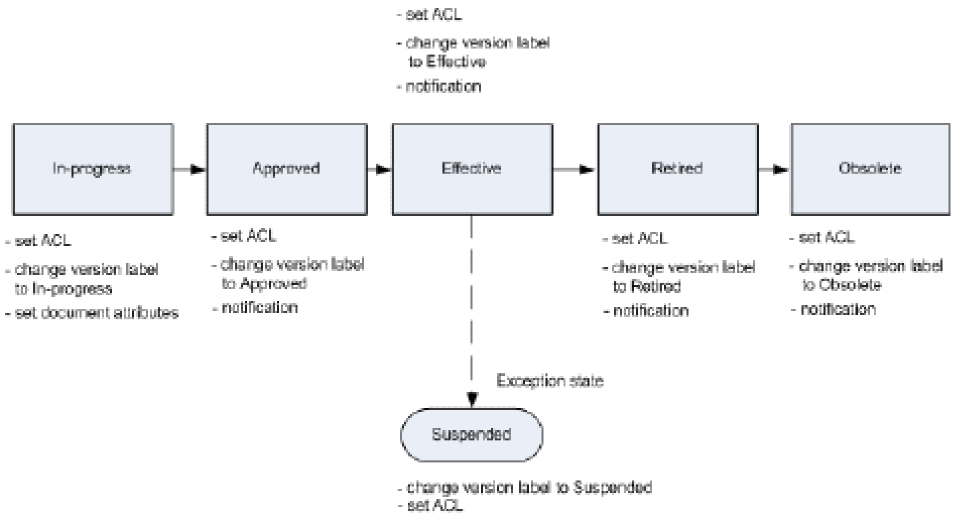
The DCM lifecycle states are defined as:

- **Initial (In Progress)**: document is created, imported, or versioned, and then routed for signoff
- **Review**: document has been routed for review
- **In Approval**: document has been routed for signoff but is not yet approved.
- **Approved**: document is approved and ready for users to be trained
- **Effective**: document is in-use indicating the latest version that viewers can access
- **Retired**: document version is no longer in-use and was superceded by a new effective version

  Checkout and reuse of a retired document is allowed and controlled by two options, dcm.lifecycle.checkout.AllowCheckoutRetired and dcm.lifecycle.checkout.AllowCheckoutRetiredIfEffective, in the dcm.properties file.
- **Suspended**: document version is temporarily retired and was suspended by a new temporary effective version, but will be made effective again at some point
- **Obsolete**: all document versions are no longer in use

During each document lifecycle state, it is good practice to have your system perform specific tasks, as shown in the following figure.

Figure 21. Recommended System Tasks for Each Controlled-Document Lifecycle State


- set ACL
- change version label to Effective
- notification
In-progress
Approved
Effective
Retired
Obsolete
- set ACL
- change version label to In-progress
- set document attributes
- set ACL
- change version label to Approved
- notification
- set ACL
- change version label to Retired
- notification
- set ACL
- change version label to Obsolete
- notification
Exception state
Suspended
- change version label to Suspended
- set ACL

For instructions on how to assign actions to a particular lifecycle state, see the section named "Editing Lifecycle State Assignments," in the *Documentum Compliance Manager Administration Guide*.

How Controlled Documents are Versioned

A DCM controlled document is labeled "CURRENT" once it is promoted to the Effective state and remains CURRENT until it is suspended or retired and replaced with a newer more recent version labeled "Most-Recent" in the In-Progress state. Users with Version permission can check in a newer version, 1.1 as opposed to 1.0 for example, of an Effective document.

When an Effective document is versioned, the Effective version keeps the "CURRENT" label, while the In Progress version does not, until it becomes Effective. This way the published or Effective version is always visible to consumers while the In Progress versions are not.

The only exception to this is when a controlled document is first created and no Effective version exists. In this case, the first In Progress version will not have a value for the version label until a version becomes Effective. If multiple In Progress versions are created, the subsequent In Progress versions do not receive a version label. The first In Progress version that becomes Effective will receive the "CURRENT" label.

When you promote a controlled document form Approved to Effective, the previous Effective document is automatically transitioned to the Retired state.

You can configure a DCM business application to specify whether to version a document to the next major version (1.0, 2.0, 3.0, etc.) upon promotion to the Effective state. For more information on how to do this, see Chapter 5, Creating and Administering Business Applications .

Making changes to a lifecycle does not affect existing documents. Making changes to a lifecycle extension does however affect existing documents. For example, changing properties such as Allow Manual Promotion and Promote On Signoff take effect immediately. It really depends on the attributes. For example, changing properties such as **Allow Manual Promotion** and **Promote On Signoff** take effect immediately.

When adding or removing signatories for a lifecycle, the change goes into effect only when a lifecycle state transition occurs. For example, if you modify an existing lifecycle that is attached to a document, and the document transitions to the In Progress state, only then do the changes take effect.

If a document is checked in as a new version to the repository, all required signatories defined for the previous version are added to the new version, and the signoff status is reset to Pending for all signoffs. However, if a document is checked into the base state (demoting the document from the Effective state to In Progress, for example), the required signatories for the new version are reset to the mandatory signatories defined for the document's base state.

Sign off relations

Current DCM signoff statuses:

- Signed off
- Pending
- Rejected
- In Rejection
- Standby
- Revising
- Parent Signoff

Single sign off chain of events:

1. Once a user is added as a signatory, their status changes to "Standby".
2. Once signoff is initiated, the status changes to "Pending".
3. Once document is signed off, the signatory status changes to "Signed off".
4. If a document is rejected, the signatory status changes to "Rejected".
5. If a document was checked out prior to signing off or rejecting, the signatory status changes to Revising. Once checked in, the status changes returns to Standby.

Aggregated sign off chain of events:

1. In case of group aggregation, once child document is added as attachment to parent, its status changes to “Parent Signoff”, parent remains "Standby".
2. Once signoff initiated against parent document, its status changes to “Pending”, child status remains "Parent Signoff".
3. Once parent document is signed off, its signatory status changes to “Signed off”, child status remains "Parent Signoff".
4. If parent document rejected, its signatory status changes to “Rejected”, child status remains "Parent Signoff". If parent object has more than one signatory and it has been rejected, other signatories status changes to “In Rejection”. DCM does not update child status and it remains "Parent Signoff" for all signatories.
5. Once parent document is checked out, it’s signatory status changes to "Revising", child status remains "Parent Signoff". Once parent document checked in, it’s status changes to Standby again, child status remains "Parent Signoff".
6. Once relationships between parent and child is broken (it was aggregate signatory to parent process relationship), status of child signoff changes to Standby from Parent Signoff. To start signoff, user should initiate signoff (because previous signoff was initiated by parent and is not valid anymore once relation is broken).

Best Practices Recommendations for Creating or Modifying DCM Lifecycles

This section provides some recommendations for configuring DCM lifecycle extensions.

Attaching Autonaming Schemes to Lifecycle States

DCM provides the option to assign an autonaming scheme to documents based on their lifecycle state. We recommend that you only assign autonames when a document hits the Approved state of a lifecycle. This is because autonames are unique. Once assigned, they cannot be re-used. If you assign the autoname at a lifecycle state other than Approved (for example, you specify an autonaming scheme for In Progress documents), then if a particular document is not approved, you will see gaps in your naming sequence for existing approved documents.

Also, if you share autonaming schemes between document classes, the two classes will share the numbering sequence, which may cause gaps in the sequence of numbers in a particular document class, since each generated autoname is unique per scheme rather than per document class.

In Progress State Best Practices

Enable manual promotion for the In Progress state.

In Approval State Best Practices

Define at least one mandatory signatory for the In Approval state. Specify automatic promotion to the Approved state when the signatories have finished processing a document.

Approved State Best Practices

Do not specify manual promotion or signatories for this lifecycle state—neither of these conditions applies to an Approved document.

The permission set for documents in an Approved state should severely restrict Write privileges, since altering an Approved document in a regulated environment will invalidate the signoff. We recommend defining the permission set for Approved and following lifecycle states as World: Relate. Give the document owner Write permission, so that the owner can modify the **Next Review Date**, **Effective Date**, and **Expiration Date** properties on the **Lifecycle Extensions** page.

Note: The expiration date feature is used in both (1) the temporarily effective document and (2) the effective document. When the expiration date occurs, DCM will automatically do the following: In case (1), retires the "CURRENT" temporarily effective document (moves object to RETIRED state) and reinstates the suspended previously effective document (moves object from SUSPENDED to EFFECTIVE state). In case (2), retires the EFFECTIVE "CURRENT" document (moves object to RETIRED state). Only if the user does a manual promote of an effective document, will the document be moved to the OBSOLETE state. DCM will provide a warning to the user when this happens. The version tree of a controlled document in the RETIRED state can be displayed. The version treee for the same controlled document in the OBSOLETE state is no longer available for display.

Effective State Best Practices

To make Effective documents easier for your users to find, you may want to remove the Effective folder in the DCM repository cabinet, and specify that Effective documents are copied to a higher-level folder in your repository.

Retired State Best Practices

When specifying DCM lifecycle extensions for the Retired state, do not select the **Manual Promote** option for change notice or change notice lifecycles.

Select the **Manual Promote** option for all other DCM lifecycles.

Creating or Modifying a DLC

To create a DLC:

1. In DAB, choose Insert > Document Lifecycle and then double-click the document lifecycle in the left pane.
2. On the DLC editor's General tab, specify a name and description in the appropriate text boxes.
3. Click the Primary Type field's ellipsis (. . .) button to open the DLC type selection window, and choose a primary object type from the Primary Type drop-down list.

   If clicking the drop-down list does not display the list, click two more times.

   A DLC applies to one primary object type. You can specify which of the primary type's subtypes can also use the DLC. Initially, the Available Subtypes window contains all of the primary type's subtypes, and the Acceptable Subtypes window is empty.
4. Use the arrow keys to choose acceptable subtypes from the available subtypes, and click OK to return to the DLC editor.
5. Click the Default Alias Set field's ellipsis (. . .) button to open the Select Alias Sets window, and choose a default alias set for the DLC to use to resolve aliases.

   You can add more alias sets for the DLC to use to resolve aliases. You might want to do this if an alias set has aliases that you want the DLC to use, but you do not want to add them to the DLC's default alias set. However, if the same alias name is defined in the default alias set and one of the additional alias sets, the alias name in the default alias is used.
6. On the Advanced tab, specify a procedure that performs additional validation to run when you validate the document lifecycle.

   See Specifying a document lifecycle validation procedure, page 165.
7. Use the Add State, Add Exception, Delete, and Layout buttons to draw the DLC state diagram.
8. For each state in the diagram, highlight its icon and click Edit State to open the DLC state editor.
9. Specify the state's properties.

10. Check in the document lifecycle.
11. To validate the document lifecycle, click Validate.

    Validating a document lifecycle makes sure that it does not have any errors that might prevent it from running. If you have specified a validation procedure, then it is executed.
12. To install the document lifecycle, make sure the Status in repository is Validated and click Install.

    If the Status in repository is Draft, then click Validate to validate it before clicking Install.

    Installing a document lifecycle makes it available to users.

To modify a DLC:

1. Double-click the document lifecycle object in the left pane and click Edit.

   If you have not inserted the document lifecycle object into your DAR, then choose Insert > Object from repository > Document Lifecycle, select the desired document lifecycle object, and then double-click the document lifecycle object.
2. In the document lifecycle editor, modify the desired fields and states.
3. Check in the document lifecycle.

   If you have not uninstalled the document lifecycle and want to check it in as the same version, it is uninstalled and placed into the Draft state, and running instances of the document lifecycle are suspended; otherwise, check it in as a new version, check it out, validate and install it.
4. To validate the document lifecycle, click Validate.

   Validating a document lifecycle makes sure that it does not have any errors that might prevent it from running.
5. To install the document lifecycle, make sure the Status in repository is Validated and click Install.

   If the Status in repository is Draft, then click Validate to validate it before clicking Install.

   Installing a document lifecycle makes it available to users and any running instances of the document lifecycle are restarted.

Tip: In any DLC editor textbox, you can use the context menu to cut, copy, and paste text, delete text, select all text in the textbox, or undo the last change. (There are no corresponding keyboard shortcut keys.)

To cut or copy, and paste text:

1. Select the text.

2. Right-click in the textbox and select either Copy or Cut.
3. Position the cursor in the same or another textbox, right-click in the textbox, and select Paste.

To delete text, select the text, right-click in the textbox and select **Delete**.

To select all text in a textbox, right-click in the textbox and select **Select All**.

To undo the last change, right-click in the textbox and select **Undo**.

Specifying a document lifecycle validation procedure

When you validate the document lifecycle, the specified validation procedure executes the document lifecycle and is executed.

This feature is only available when connected to server 5.2.5 (or greater).

To specify a post-change procedure

1. Select the PostChange tab in the DLC state editor.
2. Click the ellipsis button next to the repository Pathname field to open the Select Action Procedure dialog box, and navigate to the desired procedure.

   If the version label field does not display the procedure version, select the procedure again.

Designing the validation procedure

Write the validation procedure as a Docbasic procedure that contains code to perform validation tasks, such as verifying that states you require are specified in the document lifecycle. The validation procedure has the following signature:

```
Public Function ValidationProc( _
ByVal SessionId as String, _
ByVal PolicyId as String, _
ByVal UserName as String, _
ByRef ErrorStack as String) As Boolean
```

where:

SessionId	Session ID (session established by the calling procedure)
PolicyId	r_object_id of the dm_policy (document lifecycle) object.
UserName	Name of the user who is executing the procedure.
ErrorStack	Error message

Note: In Docbasic code, the function declaration must either appear on a single line or (as shown) have continuation characters (underscores) at the ends of all lines except the last.

Making Lifecycles Available to Custom Document Types in DCM

This section describes how to make lifecycles created using Documentum Application Builder (DAB) available for use when creating custom document classes in DCM.

To make existing lifecycles available for selection when creating a custom document class:

1. Start DAB.
2. In DAB, highlight the lifecycle you want to make available to DCM document classes or business applications.
3. Right-click the selected lifecycle, and choose **Edit** from the pop-up menu.
4. Under **Primary Type**, click on the ellipsis (...).
5. Add your object type as an **Acceptable Subtype** and click **Close**.
6. Check in the selected lifecycle as the same version.

   The lifecycle is in a Draft state.
7. Right-click the selected lifecycle, and choose **View** from the pop-up menu.
8. Click the **Validate** button.
9. Click the **Install** button under the **Status in repository** heading.
10. To associate a document template for the new object type:
    a. Start DCM and log in to the repository.
    b. Open the Templates cabinet.
    c. Click on the name of the template you want to associate with the document type.

       The Documentum Content Transfer applet launches the editing application (for example, Microsoft Word) associated with the selected template.
    d. From within the editing application, choose **Save As** and choose a temporary folder to save a copy of the template.
    e. In DCM, navigate to a cabinet, then choose **File→Import** to re-import the selected template.
    f. Click **Add Files**, and navigate to the template file you saved in Step d.

g. Select the file, then click **Next**.

h. Choose your custom object type from the **Type** pull-down menu, then click **Finish**.

Specifying actions and a procedure for a DLC state

After a document enters a state, the system performs the set of actions that you specify when you design the DLC.

The system uses the following steps to perform a state's actions:

1. Perform all of the predefined actions (referred to as Standard Actions in Application Builder).
2. Perform the action procedure.

State transitions occur within a transaction. Failure to satisfy entry criteria, failure of any action, or an error or a FALSE return value from the entry procedure or the action procedure causes the system to abort the transition transaction and back out all changes.

To make your DAR portable and flexible at runtime, use aliases as much as possible.

To specify new actions or modify an existing ones:

1. On the Action tab of the DLC state editor, click Add Action and select one of these actions:
   - To add a value to one of the object's repeating-value attributes, select Add to Repeating Attribute or Set Attribute.
   - To add a version label to the object, select Add Version Label.
   - To create a link to the object in a new location, select Link to New Location.
   - To move the object and all its links to a new location, select Move All links to Location.
   - To delete a value from one of the object's repeating-value attributes, select Remove from Repeating attribute.
   - To delete a link to the object, select Remove Link from Existing Location.
   - To delete a version label from the object, select Remove Version Label.
   - To request renditions for the object, select Request Rendition.
   - To specify a value for one of the object's single-value or repeating-value attributes, select Set Attribute.
   - To specify an owner for the object, select Set Owner.

- To specify a permission set for the object, select Set Permission Set.

These actions will be performed on the object after it enters this state.

2. Click Next.
3. If you selected, the Add Version Label or Remove Version Label options, enter the version label that you want to add or remove as a string in the text field.
4. If you selected the Link to New Location, Move All links to Location, or Remove Link from Existing Location action, choose one of these options:
   - To specify the location as a cabinet or folder, select Specify a Cabinet or Folder, click the ellipses button, and select the cabinet or folder.
   - To specify the location using an alias, select Specify a Location Alias and select an alias.

     Only aliases in alias sets in the DAR are available.
   - To specify a location using a series of attribute values, select Specify a Location Expression, enter a repository path in the text field that includes the current value of one of the object's attributes specified using the $value keyword:

     ```
     $value(attribute)
     ```
     where *attribute* is the name of the attribute from which the $value keyword retrieves a value. The $value keyword retrieves the current value of the object's attribute.

     For more information see the DAB online help.

     The Specify a Location Expression option is only available on server 5.2.5 (or greater).

     To insert the correct $value syntax for a valid attribute, position your cursor at the desired location in your repository path, select an attribute from the drop-down list, and click Insert Attribute.
5. If you selected the Add to Repeating Attribute action, perform these tasks:
   a. Select the attribute from the Repeating Attribute drop-down list.

   b. To add the value to the last position in the repeating value attribute, select Add to end of list.

   c. To add the value to a specific position in the repeating value attribute, select Add in position, and select a number that represents its position.

   d. To specify the value to add, perform one of these tasks:
      - To manually enter a value, select Specify a Value and enter a value in the text field
      - To specify an alias that will be resolved at runtime, select Specify an Alias, and select an alias in the drop-down list.

        Only aliases in alias sets in the DAR are available.

6. If you selected the Set Attribute action, perform these tasks:

    a. Select the attribute from the Attribute to set drop-down list.

    b. If the attribute is a repeating-value attribute, select a number that represents its position in the Item number field.

    c. To specify the value to add, perform one of these tasks:

        - To manually enter a value, select Specify a Value and enter a value in the text field
        - To specify an alias that will be resolved at runtime, select Specify an Alias, and select an alias in the drop-down list.

          Only aliases in alias sets in the DAR are available.

7. If you selected the Remove from Repeating Attribute action, perform these tasks:

    a. Select the attribute from the Repeating Attribute drop-down list.

    b. To delete all the values in the attribute, select Remove all values.

    c. To delete a specific value, perform one of these tasks:

        - To manually enter a value, select Specify a Value and enter a value in the text field
        - To specify an alias that will be resolved at runtime, select Specify an Alias, and select an alias in the drop-down list.

          Only aliases in alias sets in the DAR are available.

8. If you selected the Set Owner action, perform one of these tasks:

    - To specify a particular user, select Specify a repository User and select a user from the drop-down list.
    - To specify an alias that will be resolved at runtime, select Specify a User Alias and select an alias from the drop-down list.

      Only aliases in alias sets in the DAR are available.

9. If you selected the Set Permission Set action, perform one of these tasks:

    - To specify a particular permission set, select Specify a Permission Set and select a user from the drop-down list.

      Only system permission sets (including templates) — except for automatically generated ones — are available.

    - To specify an alias that will be resolved at runtime, select Specify a Permission Set Alias and select an alias from the drop-down list.

      Only aliases in alias sets in the DAR are available.

10. If you selected the Request Rendition option, select the option that corresponds to the operating system on which you have Documentum AutoRender Pro installed.

You must have AutoRender Pro installed and correctly configured for this action to complete successfully.

11. Click Finish.
12. To add additional actions, repeat Step 1 through Step 11.
13. To specify the order of execution of the actions, select an action, and click Up to move it up, and click Down to move it down.

    Actions are executed in sequence from top to bottom.
14. To edit an action, select the action and click Edit.
15. To delete an action, select the action and click Delete.

To specify a procedure:

1. Click the ellipsis button to the right of the repository Pathname field, select the desired procedure, and click Open.

   If the version label field does not display the procedure version, select the procedure again.

   To create a new action procedure, see Writing Action Procedures, page 170.
2. If you do not want to specify any action procedure, click Clear.

Writing Action Procedures

Write the *action procedure* as a Docbasic procedure and specify the attribute names of the document's object type as variables in the procedure. The action procedure has the following signature:

```
Public Function Action(                    _
ByVal SessionId as String,              _
ByVal ObjectId as String,               _
ByVal UserName as String,               _
ByVal TargetState as String,            _
ByRef ErrorString as String) As Boolean
```

where:

SessionId	Session ID (session established by the calling procedure)
ObjectId	Document ID
UserName	User ID
TargetState	The state the document is trying to enter
ErrorString	Error message

Note: In Docbasic code, you must specify the function declaration either on a single line or have line continuation characters (underscores) at the ends of all lines except the last (as shown).

Associated Mandatory Reviewers with a Lifecycle State

If you use the default lifecycle settings installed with DCM, and check in a new controlled document, the document automatically appears on the My Signoffs page for users defined as mandatory signatories.

Signatories can be automatically assigned to a document in two ways:

- defined as a mandatory signatory on a lifecycle's Properties page
- defined as a mandatory signatory with a user list rule

Adding or Removing Mandatory Signatories in a Lifecycle

You can add or remove users, or change the status of mandatory signatories to optional signatories for specified lifecycle states by modifying the Signatories tab of the lifecycle's Properties page. For step-by-step instructions, see Editing Lifecycle State Assignments, page 172.

For example, if you want to set up separate mandatory reviewer and signatory lists for a document, allowing that document to be routed for review multiple times while it is still being drafted, and then routed for signoff once the document is ready to be published, you can set up a custom DCM lifecycle with two different states, such as Draft and In Approval. When the document is ready to be reviewed, the author uses the Manual Promote feature to promote the document from the Draft state to the In Approval state.

In a DCM lifecycle, you can define as many draft states as you want before signoffs are asked for. You can also define different sets of signatories for each state, until the document becomes officially approved (which is denoted by the DCM state type for Approved).

To set up this kind of custom DCM lifecycle:

- Editing the Draft lifecycle state assignment to select the **Allow Manual Promotion** option.
- Do not specify a list of mandatory signatories for the Draft lifecycle state.

- Edit the In Approvallifecycle state assignment to set up the signatories for the In Approval state
- Specify a **Notification Text** message for the In Approval state, to notify all mandatory signatories of the required task.

Editing Lifecycle State Assignments

DCM allows you to add or modify information about the mandatory, optional, and to-be-read groups, roles, and users values for each state defined in a particular lifecycle. These values are stored as properties in an instance of a repository object, named dcm_state_extension. You can use these properties to specify that your system perform specific tasks, such as applying an autonaming or autonumbering scheme to the document, during particular document lifecycle states.

To define or modify the properties associated with lifecycle state assignments:

1. Navigate to **Administration > DCM > Lifecycle Extensions**.
2. To display all the lifecycles currently defined for DCM, choose **All Lifecycles** from the pull-down menu in the upper right-hand corner of the page.
3. To modify an existing lifecycle, right-click on a lifecycle extension and select **Properties**..
4. Optionally, modify the values one or more of the extended properties fields.

   The DCM configuration values listed on the **Lifecycle Properties Info** page are specified for each lifecycle state.

Table 45. Lifecycle Properties Info Page Fields

Field Name	Value
State Name	The name of the current lifecycle state whose properties are displayed.

Field Name	Value
State Type	The name of the DCM state that corresponds to the custom state name in the **State** field. From the pull-down menu, choose the DCM state whose extended properties you want to modify: • **In Progress** • **Review** • **In Approval** • **Approved** • **Effective** • **Retired** • **Obsolete** DCM State lists only the DCM lifecycle state types. If you have defined a custom lifecycle in DAB you must associate any custom state names with the corresponding DCM state type.
Promote after last signature	If checked, the document will automatically be promoted to the next defined lifecycle state upon signoff.
Send To Be Read Notifications	If checked, the system will automatically send To Be Read notifications to the users or groups specified on the **Distribution List** page.
Use Autoname scheme	If checked, the system automatically renames the document when it enters this lifecycle state, using the autonaming scheme you defined for the document class.
Allow manual promotion	If checked, this option specifies if the system should display a **Promote** button for a particular signatory for the specific lifecycle state that you are modifying.

Field Name	Value
Allow demote to previous state	If checked, this option specifies if the system should display a **Demote** button for a particular signatory for the specific lifecycle state that you are modifying.
Manifest e-signature	If checked, the specified lifecycle state prepends a signoff page to a PDF rendition of the controlled document.
Signoff Code	To select a signoff justification code, click **Edit**. This justification specifies the reason code that a user must use when performing a signoff or promote process When the **signoff_justification_text** page appears, you can either select an existing signoff justification code from the displayed list (for example, “Meets FDA requirements.”) or you can type in a new justification code. Click **OK** to save your selection and to return to the **Lifecycle Properties Info** page.
Signoff Confirmation Text	To select a signoff confirmation statement, click **Edit**. This signoff confirmation statement is displayed when a user performs a signoff task. When the **signoff_confirmation_text** page appears, you can either select an existing signoff confirmation code from the displayed list (for example, “Are you certain you want to signoff this document?”) or you can type in a new confirmation code. Click **OK** to save your selection and to return to the **Lifecycle Properties Info** page.

Field Name	Value
Reject Code	To select a rejection justification code, click **Edit**. This justification specifies the reason code that a user must use when rejecting a document. When the **reject_justification_text** page appears, you can either select an existing rejection justification code from the displayed list (for example, "Does not meet internal requirements.") or you can type in a new rejection justification. Click **OK** to save your selection and to return to the **Lifecycle Properties Info** page.
Reject Confirmation Text	To select a rejection confirmation statement, click **Edit**. This rejection confirmation statement is displayed when a user tries to reject a document. When the **reject_confirmation_text** page appears, you can either select an existing rejection confirmation code from the displayed list (for example, "Are you certain you want to reject this document?") or you can type in a new rejection code. Click **OK** to save your selection and to return to the **Lifecycle Properties Info** page.
Notification Text	To select a notification statement, click **Edit**. This text appears when the system sends out a notification to the users and groups defined on the **Distribution List** page. When the **signoff_notification_text** page appears, you can either select an notification statement from the displayed list (for example, "Your signoff is required.") or you can type in a new notification statement.

Field Name	Value
	Click **OK** to save your selection and to return to the **Lifecycle Properties Info** page. **Note:** You can only send To Be Read notifications for documents in the Effective state.
TBR Notification Text	To select a TBR notification statement, click **Edit**. This text appears when the system sends out a to-be-read notification to the users and groups defined on the **Distribution List** page. When the **tbr_notification_text** page appears, you can either select an notification statement from the displayed list (for example, “This document must read and understood.”) or you can type in a new notification statement. Click **OK** to save your selection and to return to the **Lifecycle Properties Info** page.

5. To save your changes and exit the **Lifecycle Extension Properties** page, click **OK**. Otherwise, click one of the navigation links in upper left hand corner of the page to view or modify the properties on other tabs.

6. To add or remove signatories for the selected lifecycle state, click the **Signatories** tab from the page navigation menu on the top of the page.

7. To add users as signatories for the selected lifecycle state, click the **Add** link.

8. To change the displayed options, select what kinds of items you want to see from the pull-down menu at the bottom of the page.

9. To select the name of a user, group or role to assign as a signatory, click the checkbox corresponding to that group or user name, then click **Add**.

10. Click **OK** to save your changes and to return to the **Lifecycle Properties: Signatories** page.

11. To make the user, group, or role a mandatory signatory, click the **Mandatory** checkbox that corresponds to the signatory name.

When signoff is required for the document in a particular lifecycle state, the document will appear in the My Signoffs window for the selected user, group, or role.

12. To specify that any member of a selected group or role can signoff a document on behalf of all the group or role members, click the **Any Members** checkbox.
13. To save your changes and exit the **Lifecycle Extension Properties** page, click **OK**. Otherwise, click one of the navigation links in upper left hand corner of the page to view or modify the properties on other tabs.
14. To choose which users and groups receive notification (including TBR notices) for the selected lifecycle state, click the **Distribution List** link from the page navigation menu on the right side of the page.
15. To add users or groups to the notification distribution list, click the **Add** link.
16. To change the displayed options, select what kinds of items you want to see from the pull-down menu at the bottom of the page.
17. To select the name of a user, group or role, click the checkbox corresponding to that group, role, or user name.
18. Click **OK** to save your changes and to return to the **Lifecycle Properties: Distribution List** page.
19. To save your changes and exit the **Lifecycle Extension Properties** page, click **OK**. Otherwise, click one of the navigation links in upper left hand corner of the page to view or modify the properties on other tabs.
20. To choose a workflow to use at this particular lifecycle state, click the **Workflows** link from the page navigation menu on the right side of the page.
21. To add a workflow to this lifecycle state, click the **Add** link.

    The selected workflow will be used if a user chooses to start a workflow for the document when it is in this lifecycle state. If you specify multiple workflows, then the user must choose which workflow to use.
22. To change the displayed options, select whether to view only the CURRENT versions of available workflows, All versions, or Most-Recent versions from the pull-down menu at the bottom of the page.
23. To select the name of a predefined workflow, click the checkbox corresponding to that workflow, then click **Add**.
24. Click **OK** to save your changes and to return to the **Lifecycle Properties: Workflows** page.
25. To save your changes and exit the **Lifecycle Extension Properties** page, click **OK**. Otherwise, click one of the navigation links in upper left hand corner of the page to view or modify the properties on other tabs.

26. To override the default signoff codes, rejection codes, justification text, and notification text already defined for users or groups in particular lifecycle states, and to replace the default codes assigned to those user or groups with custom codes or text, click the **Overrides** link from the page navigation menu on the right side of the page.
27. To view or modify the default signoff codes, rejection codes, justification text, and notification text defined for a particular user or group, click the associated **Show Info** radio button.
28. If you selected **Show Info** for a displayed user or group, you can now modify the values for one or more of the fields.

    The DCM configuration values listed on the **DCM Override** page can be specified at each lifecycle state, or they can be specified per user, group, or role.

Table 46. DCM Override Page Fields

Field Name	Value
Signoff Code	The justification, or reason code text, that a user must use when performing a signoff or promote process.
Signoff Confirmation Text	The text for the signoff confirmation. Click **Edit** to choose from a list of pre-existing signoff confirmation text strings. You can also type a new signoff confirmation. For example, "`Meets FDA requirements.`" This confirmation text applies to DCM workflow task processes which require signature (such as Finish, Reject, Delegate, and Close). When a controlled document is routed via a workflow, then the justification and confirmation defined for this document type will be enforced and the justification selected will be applied to its signature page.
Reject Code	The rejection text list that a user selects from when performing a reject operation.
Reject Confirmation Text	The text for the rejection confirmation.

Field Name	Value
Notification Text	Text used in the notification process.
TBR Notification Text	Text used in the to-be-read notification process.

29. To save your changes and exit the **Lifecycle Extension Properties** page, click **OK**. Otherwise, click one of the navigation links in upper left hand corner of the page to view or modify the properties on other tabs.
30. When you have finished modifying the lifecycle configuration, click **OK** to save your changes.
31. For the changes to take effect, log out of DCM and log in again.

Enabling Manual Promotion for a Lifecycle State

You must specify that manual promotion is allowed in a particular lifecycle for each specific state that you want to promote to.

⚠ **Caution:** Do not enable manual promotion for change notice and change request lifecycles

To enable manual promotion for a lifecycle state:

1. Navigate to **Administration > DCM > Lifecycle Extensions**.
2. Select **Lifecycle Extensions**.
3. Right-click a lifecycle extension displayed in the content pane and select **Properties**
4. Click the lifecycle state in the graphic you want to modify and select the checkbox next to **Allow Manual Promotion**.
5. Click **OK** to save your changes.
6. Log out of DCM and then log in again for the changes to take effect.

Chapter 9

Using Workflows with DCM

This chapter describes how DCM uses workflows to assign and manage tasks. Topics discussed in this chapter include:

Workflows and Uncontrolled Documents

A workflow is an automatic process that assigns specific tasks to specific users, in sequence, in order to carry out organizational procedures. Workflows let you pass files and instructions from person to person according to a predefined sequence. For example, an organization might use workflows to process insurance claims or develop new products.

To start a workflow, you choose the workflow template that includes the sequence of tasks you want performed. Some workflow templates specify the users who receive the tasks; others allow you to select the users.

You can use a workflow template repeatedly to initiate task sequences. Multiple users can start workflows from the same template at the same time. A single user can start multiple workflows from the same template at the same time (the user must have at least Relate permission on the template).

Developers and administrators create workflow templates using Documentum Workflow Manager. For details on creating workflow templates see the Workflow Manager online Help.

Users receive workflow tasks in their Inboxes. When a user completes a task, the user forwards it from the Inbox, and the workflow automatically notifies the next user in sequence. The users in a workflow are called the workflow's *performers*.

A workflow template might let you direct a task to a group of users, in which case the first user to accept the task becomes the one who performs it. The task is removed from the other users' Inboxes.

When you start a workflow, you can attach files you want users to view or reference. A file can be attached to only one workflow at a time. Users in the workflow can attach and remove files as the workflow progresses.

Users can edit attached files. The workflow template determines whether the edited version or the original version stays with the workflow as the workflow progresses.

Each workflow has a workflow supervisor, who can pause, stop or make other changes to the workflow as the workflow is active.

Workflows can include automatic tasks, such as the execution of scripts. If an automatic task fails, the workflow supervisor is notified and can retry, perform or stop the task. Automatic tasks allow you to integrate workflows with lifecycles — for example allowing you to promote files to new lifecycle states as they progress through a workflow.

The Workflow Reporting utility lets you perform additional management functions and lets you view all workflows in a repository.

Once started, a workflow is in one of three states:

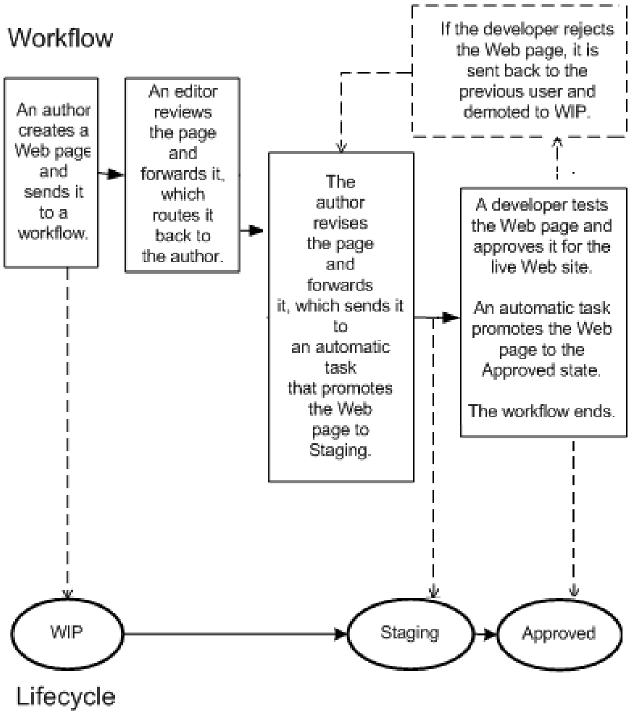
- Running: The workflow running normally according to the workflow template.
- Paused: The workflow is temporarily halted, but expected to be reinstated. If reinstated, it continues from the point at which it was halted.
- Terminated: The workflow is aborted and cannot be reinstated.

The following icons are used in workflows:

- : A workflow template
- : A workflow.
- : A package, which is a container for attaching a file. You click this icon to attach a file. Text adjacent to the package tells you whether an attached file is optional or mandatory.
- ▶: A currently running workflow.
- ❚❚: A workflow that is paused.
- 🟥: A workflow that is stopped.

The following figure shows the integration of a workflow with a lifecycle. A lifecycle defines the different stages a file goes through as it is created, edited, approved, and, eventually, retired. An author creates a file for the Web and forwards the file to a workflow, which sends a review task to an editor. The editor suggests changes and sends the file back to the author, who revises the file and forwards the task, which initiates an automatic task. The automatic task promotes the file from the WIP to Staging and sends the file to a developer, who tests it on a Staging Web server. If the developer rejects the file, DCM demotes the file to WIP and returns it to the author. If the developer signs off the file, an automatic task promotes it to Approved, and the workflow ends.

Figure 22. Integration of a workflow and lifecycle


Workflow
An author creates a Web page and sends it to a workflow.
An editor reviews the page and forwards it, which routes it back to the author.
The author revises the page and forwards it, which sends it to an automatic task that promotes the Web page to Staging.
A developer tests the Web page and approves it for the live Web site.
An automatic task promotes the Web page to the Approved state.
The workflow ends.
If the developer rejects the Web page, it is sent back to the previous user and demoted to WIP.
WIP
Staging
Approved
Lifecycle

Workflows in DCM and Controlled Documents

Default controlled workflows are provided with DCM. — The following controlled workflows are available with DCM out of the box:

- **DCM Signoff Process (In Progress)**

  Use this default workflow when the Version Label of your document indicates In-Progress.

- **DCM Signoff Process**

  Use this default workflow when the Version Label of your document is listed as "Most-Recent".

- **DCM Promote on Signoff Process**

  Use this default workflow when your document is "Most-Recent" and needs to be promoted after the last person has signed off.

The default workflows are available in the **Document Classes** and **Business Applications** nodes. You can use the default workflows provided for your convenience or customize your own workflows using Workflow Manager or Business Process Manager.

Note: Workflow is a Webtop feature leveraged by DCM. Although DCM supports workflows it also supports controlled workflows. DCM, in Documentum 4, does not support routing of controlled documents for signoff or auto-promotion. DCM, in Documentum 5, does support workflows for both controlled documents and uncontrolled documents.

Also note, DCM currently limits packages to one for controlled workflows.

Typical or normal workflows or simply workflows, as described in Workflows and Uncontrolled Documents, page 181, are used to manage uncontrolled documents. A controlled workflow however is used to manage both controlled documents and uncontrolled documents, primarily controlled documents. An uncontrolled document however must first be converted to a controlled document before it is attached to a controlled workflow. DCM allows normal workflows to be used in addition to controlled workflows. The DCM user interface inherits the menu option for workflows from Webtop and adds a separate menu item for controlled workflows.

Figure 23. “Workflow” for Uncontrolled Documents is listed Under the Tools Menu

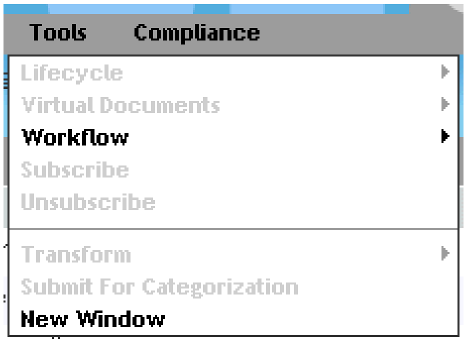
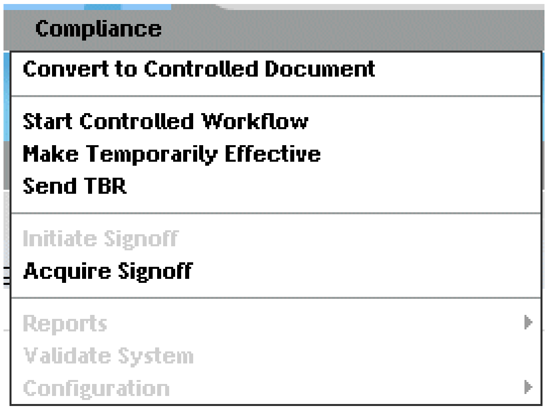


Figure 24. “Start Controlled Workflow” for Controlled Documents is listed under the Compliance Menu



A typical DCM controlled workflow consists of a review or approval process. During the review cycle for example, multiple tasks may occur. For instance, a reviewer task may consist of a group. Within the group, there may be several options on who should signoff, such as *wait for all responses*, the *first one to respond*, *reject if at least one rejects*, and/or notify all users in the group if someone rejects. Other activities may include forwarding the package to other groups for review.

A typical DCM controlled workflow means a workflow which does not allow "authoring". The current workflow would be aborted to enforce the re-run of the signoff process if authoring were to occur. DCM "halts" the controlled workflow when a user checks out the attached task (attachment) and then "resumes" the workflow when the user performs a "Cancel Checkout". DCM "aborts" the controlled workflow if the user checks in the attachment. DCM uses dm_superuser_dynamic group to handle the workflow as the halt/resume/abort APIs requires sysadmin or superuser privileges.

You create controlled document workflows using Documentum Workflow Manager or Business Process Manager (BPM). You then select the workflow using the DCM configuration screens, such as the Document Class or Lifecycle Extensions configuration pages.

Note: BPM extends the functionality of Documentum Workflow Manager. BPM is a separate utility which you should be familiar with as an Administrator. While you can use either tool to create process templates, BPM offers enhanced options, including most notably the ability to create templates for custom activity types. Using BPM for workflow design you can specify people or groups to signoff the document in an ordered manner.

DCM workflows are used to drive the Signoff process and are more easily customized to particular client requirements. The Signoff process relies on signoff and signoff itself is reflected, and verifiable, in audit trails. DCM workflows can be used to connect activities required to push controlled documents through their lifecycle states. Unless a controlled document is signed off it will not get pushed to the next lifecycle state.

Note: The pre-existing “DCM signoff initiation” implementation, which does not leverage workflow, is still in place. Both mechanisms coexist, in this release, so that users are not forced to use only the new controlled workflow mechanism. The point at which the decision is made to branch control of the signoff process to workflow or to DCM signoff initiation occurs when the signoff process for signoff is started. The pre-existing business component is enhanced to account for the possibility that a DCM signoff is happening within a workflow context.

Tasks resulting from DCM workflows appear in your Inbox with a priority rating depicted by the following icon ☰ next to the respective checkbox. Signoff tasks and notifications, for example, appear in the Inbox.

Starting a DCM Workflow with Attachment — You can start a workflow with an attachment if the document is not marked as a controlled document. DCM marks controlled documents. To place a controlled document into a workflow you must select “Start Controlled Workflow”. An uncontrolled document must first be converted to a controlled document before attaching it to a controlled workflow. The Start Controlled Workflow option is not available in the Compliance menu if you have an uncontrolled document selected.

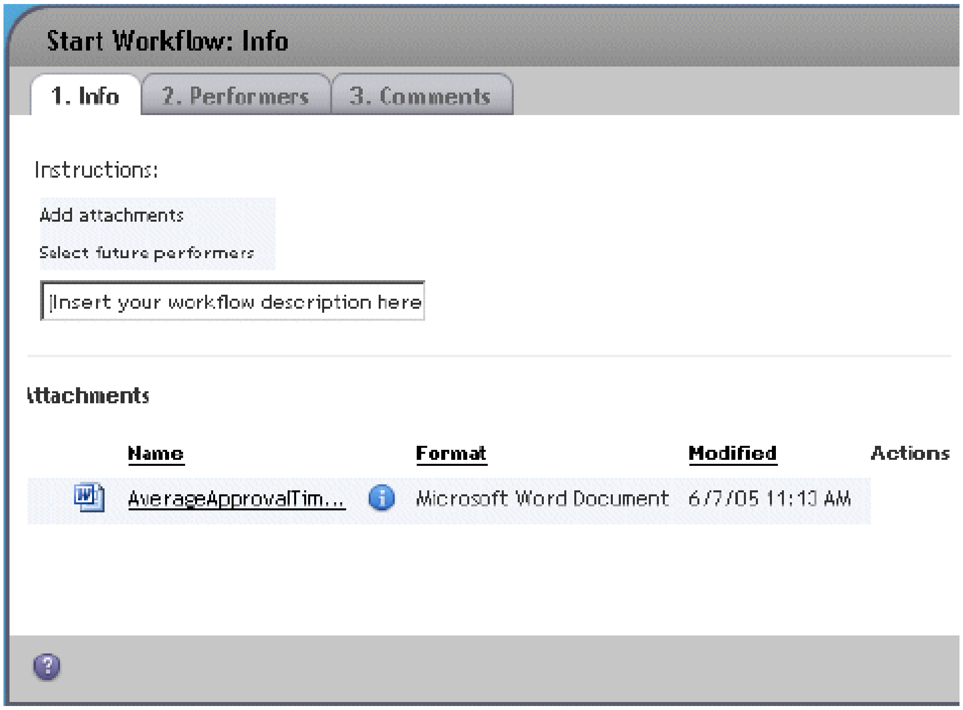
The document's state-type determines the choice of workflows available for the first screen. These workflows are defined by the DCM lifecycle extension GUI. Anyone attempting to start a controlled workflow is prevented if one already exists for the same document. Users are also prevented from attaching additional documents to the workflow, they are instead able to attach documents directly to the controlled document.

Accessing DCM Workflow and Attachments — Navigate to DCM Workflows as follows: **Tools>Workflow>Start**. If a workflow attachment is available the Start option is modified to appear as follows: **Tools>Workflow>Start Attachments**. The Workflow menu item in DCM is customized to meet DCM functionality. There are minor visual differences between the DCM and the Webtop user interface Workflow menu items when you access the Start Workflow dialog boxes. DCM for example removes the ability to add or remove attachments.

Accessing Controlled Workflows — Navigate to DCM Controlled Workflows as follows: **Compliance>Start Controlled Workflow**. The Compliance menu item is specific to DCM and is therefore not available in Webtop.

Note: A document must be selected before you start a controlled workflow.

Figure 25. DCM Start Workflow Dialog Box Info Tab



Start Workflow: Info

Instructions:

Add attachments

Select future performers

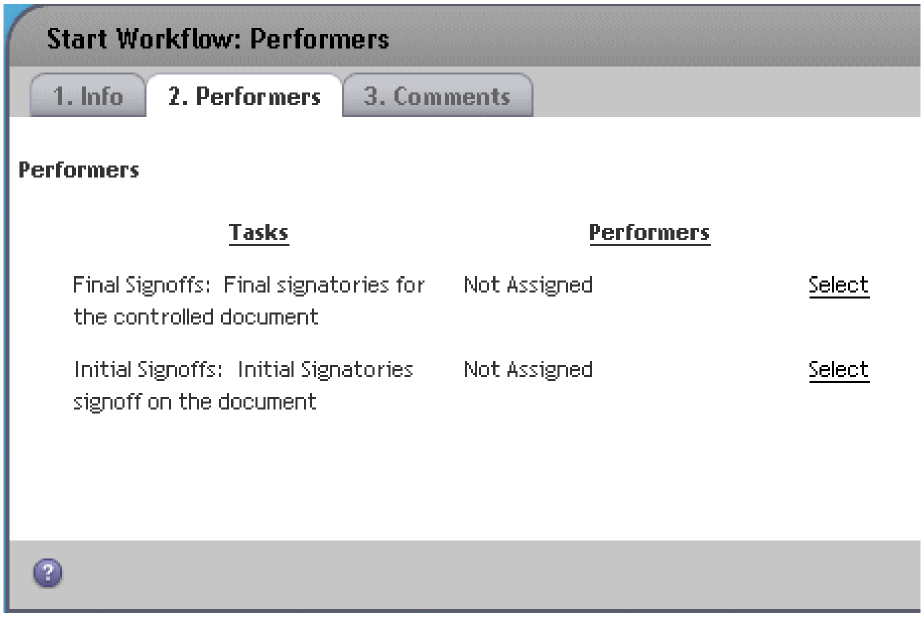
Insert your workflow description here

ttachments

Name	Format	Modified	Actions
AverageApprovalTim...	Microsoft Word Document	6/7/05 11:13 AM	

Note: The Performers Tab is not available unless Dynamic Performers are specified at workflow initiation.

Figure 26. DCM Start Workflow Dialog Box Performers Tab



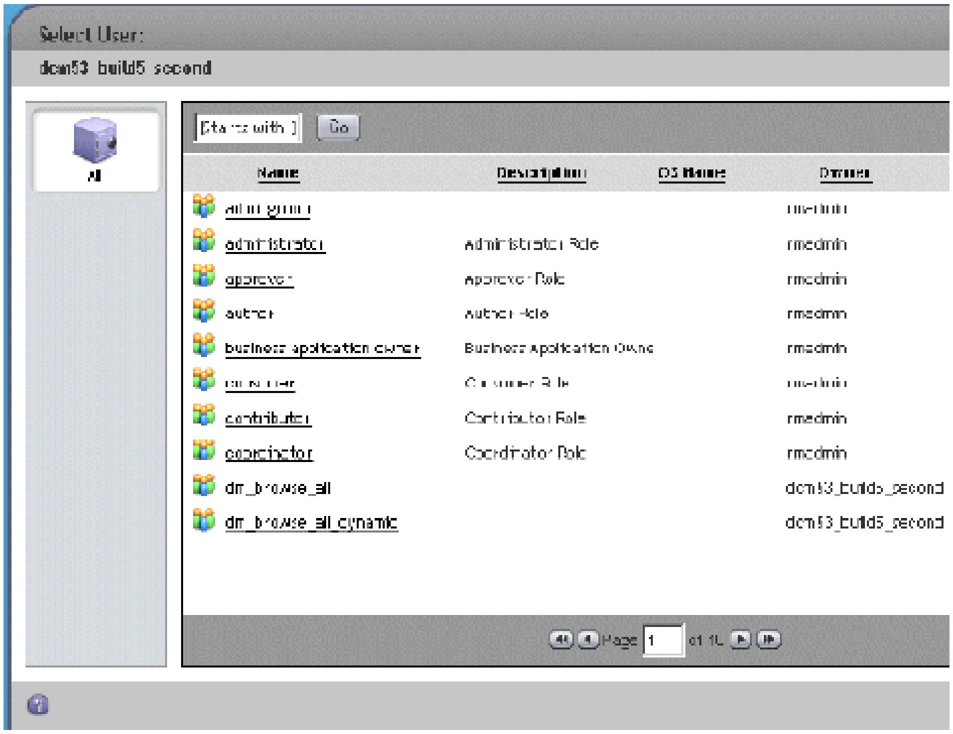
Performers

Tasks	Performers	
Final Signoffs: Final signatories for the controlled document	Not Assigned	Select
Initial Signoffs: Initial Signatories signoff on the document	Not Assigned	Select

Note: The Comments tab has not been modified for DCM; it is the same for both DCM and Webtop.

The Select User dialog box appears when you click **Select**.

Figure 27. Select Performers Dialog Box



Select User:

dcm53_build5_second

[Starts with] Go

Name	Description	OS Name	Owner
admingroup			rmadmin
administrator	Administrator Role		rmadmin
approver	Approver Role		rmadmin
author	Author Role		rmadmin
business application owner	Business Application Owne		rmadmin
consumer	Consumer Role		rmadmin
contributor	Contributor Role		rmadmin
coordinator	Coordinator Role		rmadmin
dm_browse_all			dcm53_build5_second
dm_browse_all_dynamic			dcm53_build5_second

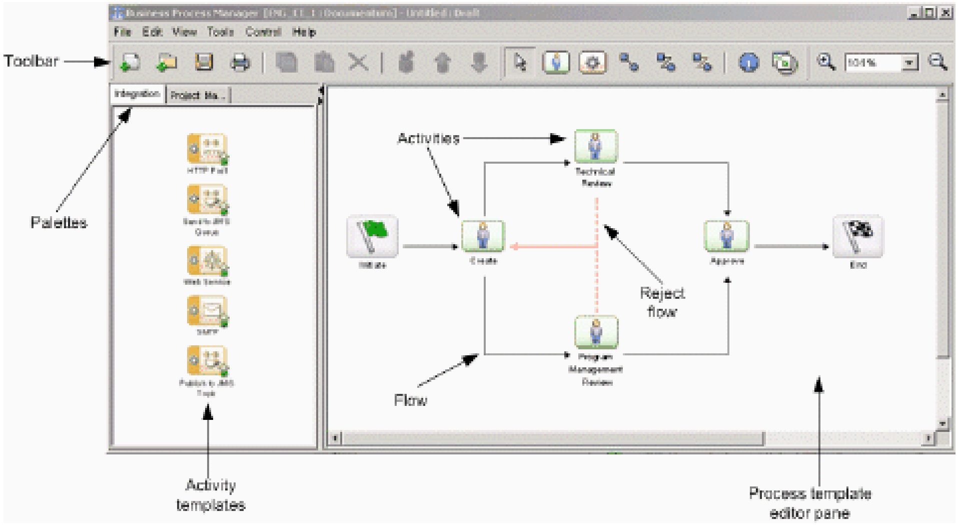
DCM Workflow Customization Using BPM

DCM is deployed with certain out-of-the-box workflows that drive controlled document lifecycle states. Using Business Process Manager you can create workflow designs specifying people or groups who must signoff a document in an ordered manner.

Note: BPM is installed separately from DCM. You can only view existing workflow templates if you connect BPM 5.3 to a 5.2.5 repository. Saving and editing of workflow templates requires repository version 5.3 or later.

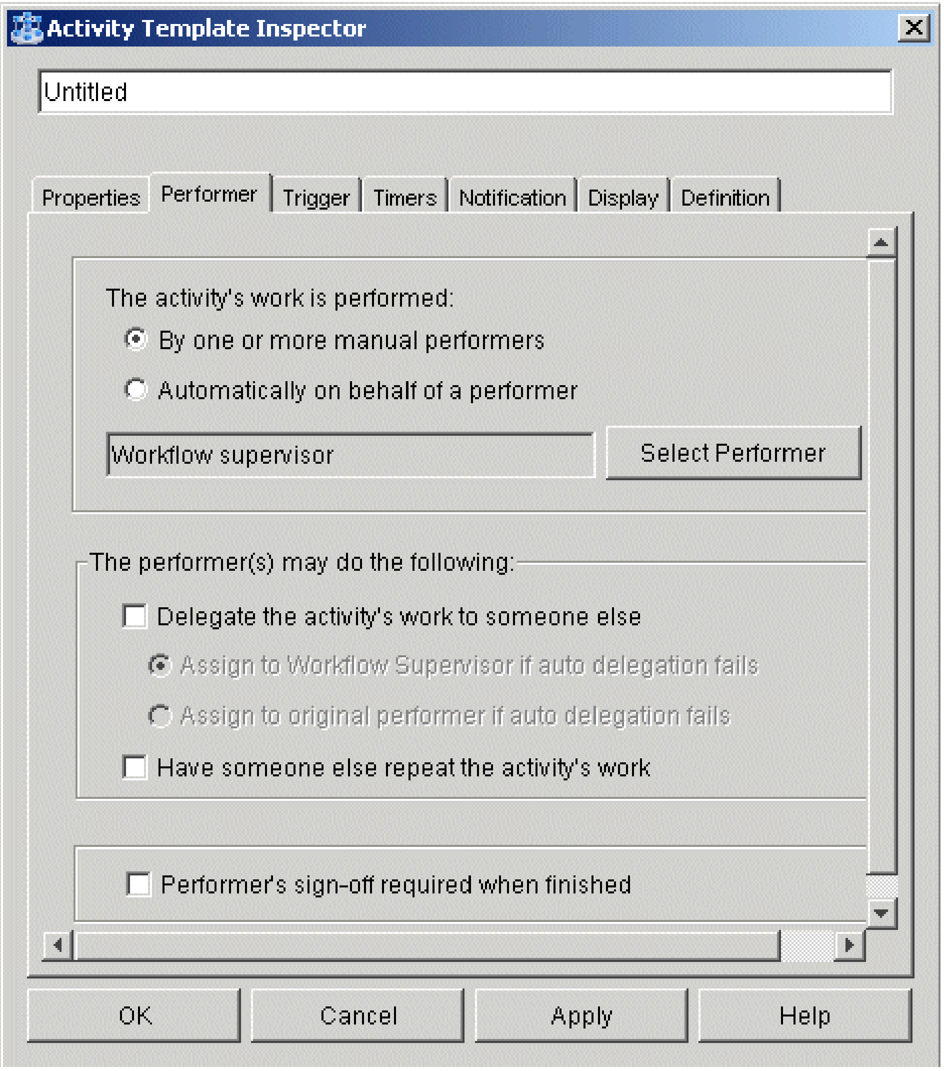
Business Process Manager is a graphical tool for laying out and defining your workflow. The Business Process Manager window is divided into two major panes:

Figure 28. Business Process Manager



Required signatories are the Performers for a task. DCM is responsible at workflow initiation time to populate this group with the DCM required signatories.

Figure 29. BPM Activity Template Inspector



Untitled

The activity's work is performed:

● By one or more manual performers

○ Automatically on behalf of a performer

Workflow supervisor

The performer(s) may do the following:

□ Delegate the activity's work to someone else

● Assign to Workflow Supervisor if auto delegation fails

○ Assign to original performer if auto delegation fails

□ Have someone else repeat the activity's work

□ Performer's sign-off required when finished

The steps for completing the workflow activity performer's configuration for a task are detailed in the Business Process Manager 5.3 Administration Guide. You will gain valuable information and complementary understanding of workflow terminology and design after you have read the *Business Process Manager Administration Guide*.

How to Configure a Controlled Workflow

You must set up a controlled workflow before you can start one. You may need to reconfigure your workflow template to suit a controlled workflow. Workflow templates can be set up for a controlled workflow or set up for an uncontrolled workflow.

Note: These procedures require an understanding of Business Process Manager (BPM). BPM has its own documentation set you can refer to for complete details.

Follow these procedures to configure a controlled workflow:

1. Start Business Process Manager (**BPM**).
2. Open one of the sample DCM controlled workflow templates provided in the following folder:

   File>Open>System> Applications>**DcmWorkflows**

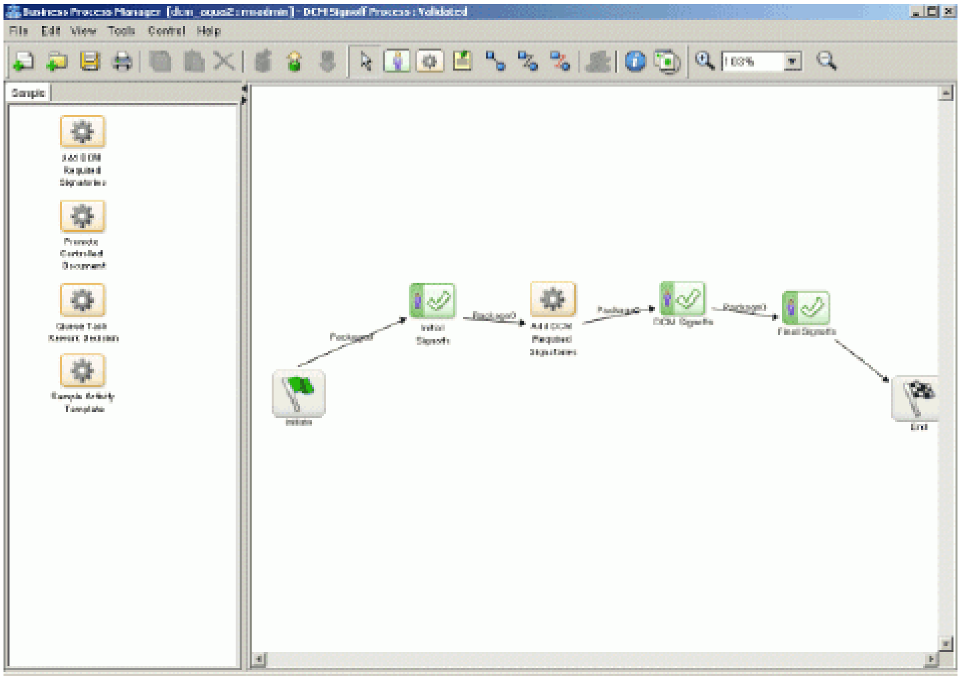
   The three controlled workflow templates to choose from include:

   - DCM Promote on Signoff Process
   - DCM Signoff Process
   - DCM Signoff Process (In Progress)

   Note: You will have to uninstall the selected workflow template if you get a message telling you the selected workflow template is in use. To uninstall the selected workflow template, click the Uninstall Template button in the Toolbar.

   An example of the DCM Signoff Process is depicted in the following figure.

   Figure 30. DCM Signoff Process

   
   Business Process Manager [dcm_aqua2 : msadmin] - DCM Signoff Process : Validated
File
Edit
View
Tools
Control
Help
103%
Sample
Add DCM Required Signatories
Promote Controlled Document
Queue Task Resubmit Decision
Sample Activity Template
Initiate
Package0
Initial Signoffs
Package0
Add DCM Required Signatories
Package0
DCM Signoffs
Package0
Final Signoffs
End

3. Using the Activity Template Inspector verify if **activity templates** are configured for controlled workflows.

The Create Manual Activity templates and Create Automatic Activity templates can be configured to suit a controlled workflow or an uncontrolled workflow. You may need to change the settings from an uncontrolled workflow to a controlled workflow in the activity templates.

Note: If there are no required signatories added to the automated activity template called **Add DCM Required Signatories**, the template is empty and the **Workflow Supervisor** receives notification in their **Inbox** that the auto-task failed. In such an event the workflow supervisor must finish the task as a trigger to make the workflow go to the next activity, without signoff. You can avoid this situation by selecting a workflow template, to begin with, which bypasses the DCM Signoffs activity template and takes the signoff task to the Final Signoffs activity template.

4. Configure all activity templates for a controlled workflow.

   To modify or reconfigure an activity template, you need to double-click the activity template appearing in the right pane of the selected workflow template and include the following information in the **Task name** field:

   (dcm_signoff): {dmi_package.r_component_name}: {dmi_queue_item.task_name}

   If in a controlled workflow, the signoff task does not have *dcm_signoff* in the task name, DCM will allow the activity to finish. A DFC log (level DEBUG) is created when a performer accepts such a task. The message is "Accepting a non-DCM activity <activityName> on a controlled document <docName>". However, there will be no auditing happening for tasks like this because the activity is not a DCM activity.

5. Configure the selected **controlled workflow** using **Webtop**:

   a. Navigate to the following folder in DCM.

      System>Applications>**DcmWorkflows**

   b. Select the applicable controlled workflow, in this example the DCM Signoff Process, and select a checkmark to Show All Objects.

   c. Open the **Properties Info** tab of the workflow template.

   d. Type **dm_dcm** in the **Application Type** field.

   e. Click **OK**.

      You now have a controlled workflow template.

Note: Follow these instructions if you are using Documentum Workflow Manager instead of BPM:

To set up a controlled workflow template to be used for a specific version of a document, edit the **Package Info** from the **Flow Inspector**.

To set up an activity to be a signoff activity, edit the activity properties within DCM. When you edit, set the **Task Subject** field to be: **(dcm_signoff): {dmi_package.r_component_name}: {dmi_queue_item.task_name}**

The following figures are provided for your information.

Figure 31. Sample workflow template configuration used to bypass Required Signatories.

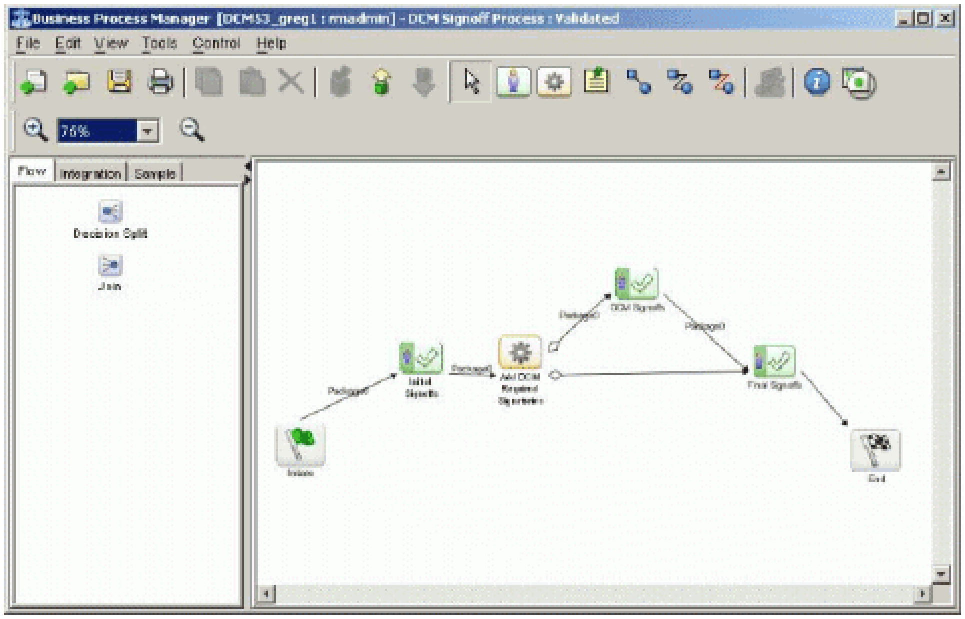
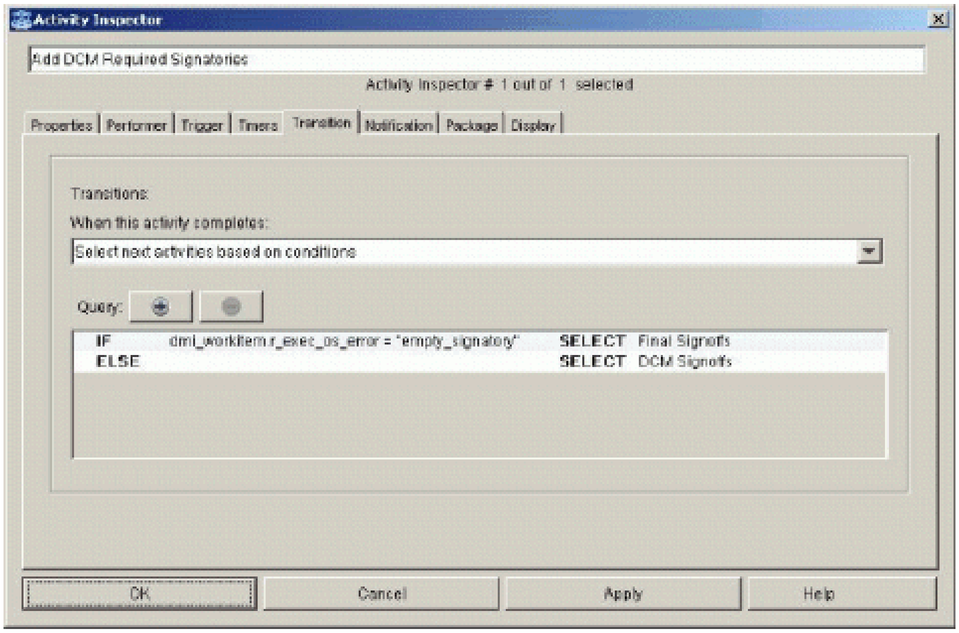

Business Process Manager [DCM53_greg1 : rmadmin] - DCM Signoff Process : Validated
File Edit View Tools Control Help
75%
Flow Integration Sample
Decision Split
Join
Package0
Initial Signoffs
Package0
Add DCM Required Signatories
Package0
DCM Signoffs
Package0
Final Signoffs
Initiate
End

Figure 32. Use these settings to configure the Auto-task to transition to the next activity.


Activity Inspector
Properties Performer Trigger Timers Transition Notification Package Display
Query:
OK Cancel Apply Help

Add DCM Required Signatories

Activity Inspector # 1 out of 1 selected

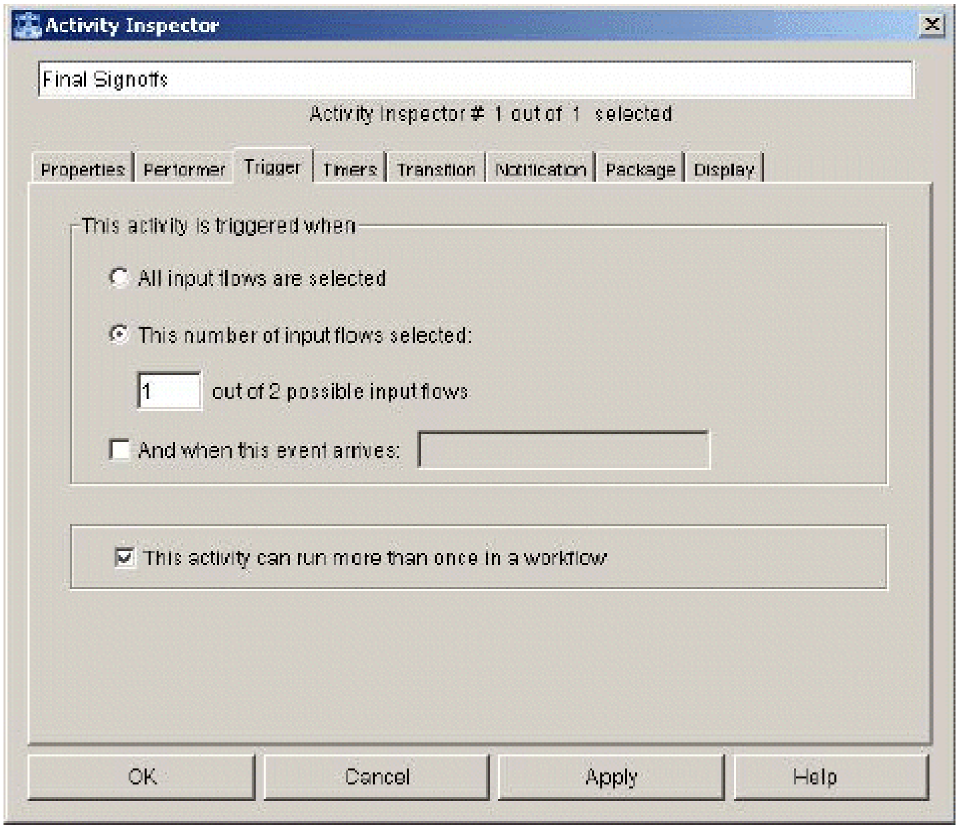
Transitions:

When this activity completes:

Select next activities based on conditions

IF	dmi_workitem.r_exec_os_error = "empty_signatory"	SELECT	Final Signoffs
ELSE		SELECT	DCM Signoffs

Figure 33. Use these settings to configure the Final Signoff activity so that it can bypass the DCM Signoff activity.



Final Signoffs

Activity Inspector # 1 out of 1 selected

This activity is triggered when

○ All input flows are selected

● This number of input flows selected:

1 out of 2 possible input flows

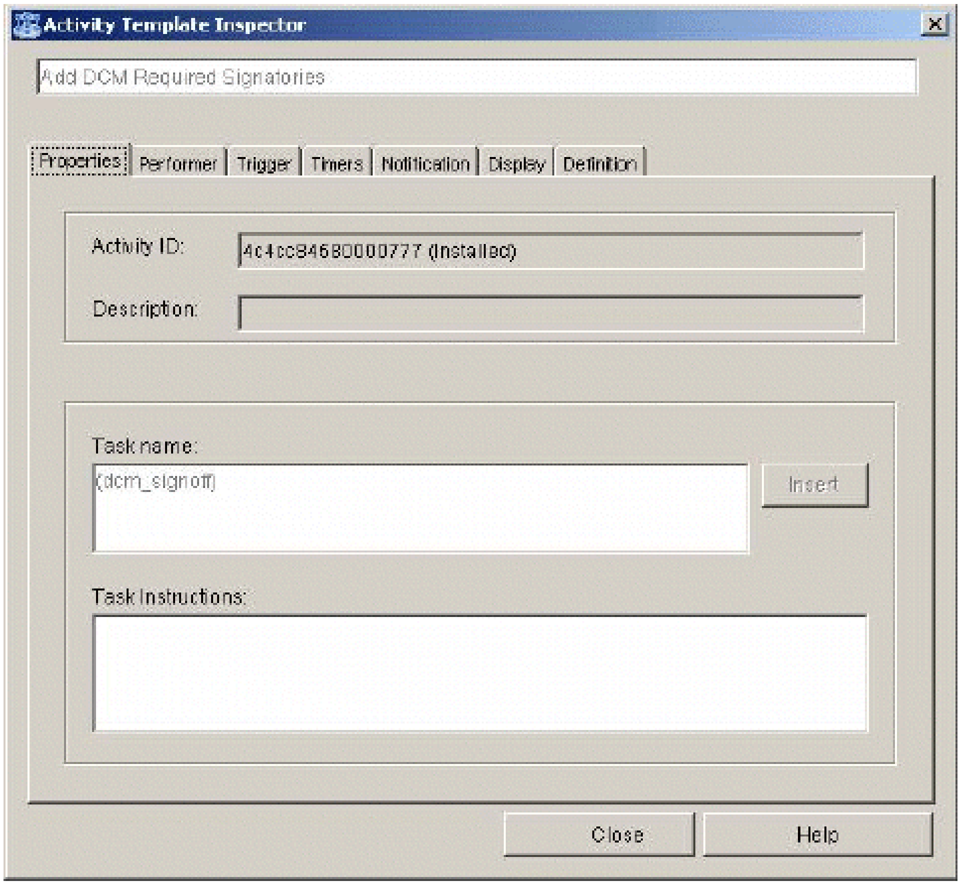
□ And when this event arrives:

☑ This activity can run more than once in a workflow

In the above configuration, if there are no required signatories on the document when the auto-task runs, the next activity triggered will be the ‘Final Signoff’ rather than the ‘DCM Signoff’.

The following two figures depict the Add DCM Required Signatories.

Figure 34. Use htese settings to Add DCM Required Signatories on the Properties tab.



Add DCM Required Signatories

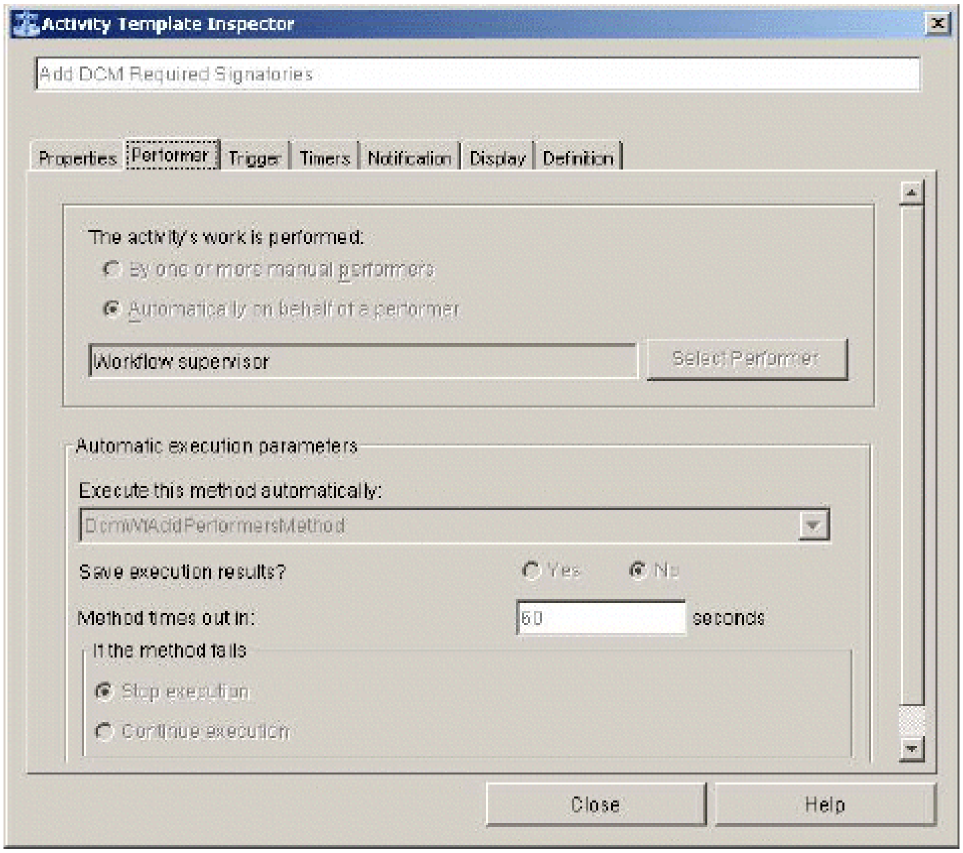
Activity ID: 4c4cc84680000777 (Installed)

Description:

Task name: {dcm_signoff}

Task Instructions:

Figure 35. Us ethese settings to Add DCM Required Signatories on the Performer tab



Add DCM Required Signatories

The activity's work is performed:

○ By one or more manual performers

● Automatically on behalf of a performer

Workflow supervisor

Automatic execution parameters

Execute this method automatically: DcmWfAddPerformersMethod

Save execution results? ○ Yes ● No

Method times out in: 60 seconds

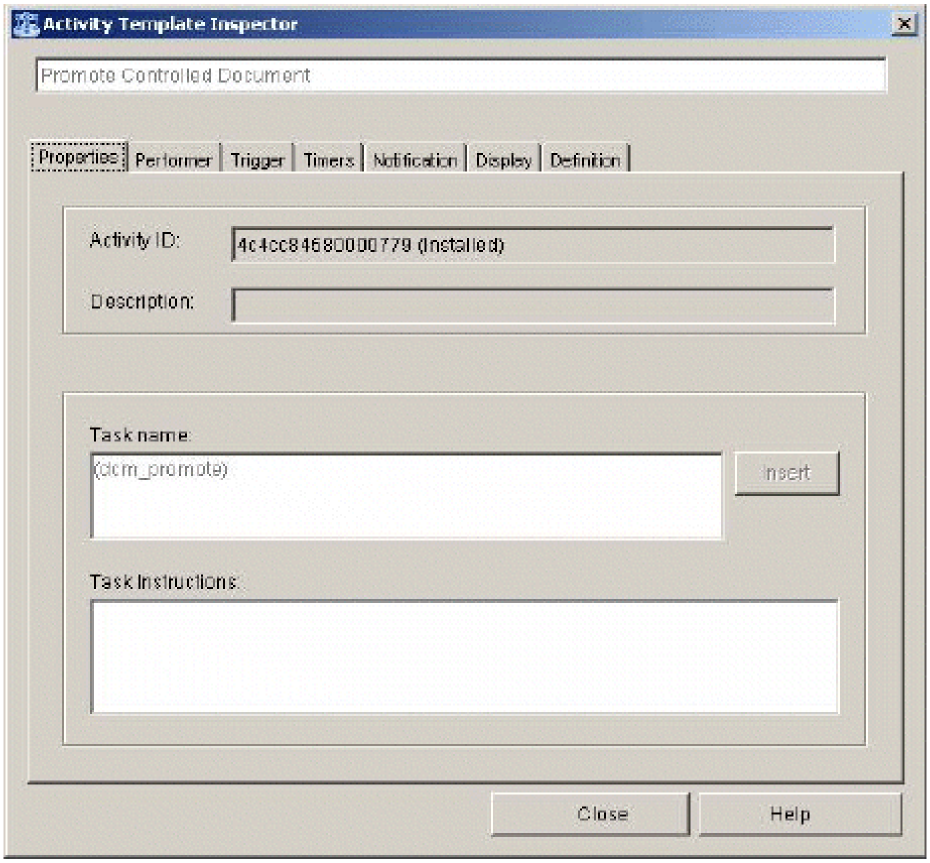
If the method fails

● Stop execution

○ Continue execution

The following two figures depict the Promote Controlled Document.

Figure 36. Use these settings to Promote Controlled Document on the Properties tab.



Activity Template Inspector

Promote Controlled Document

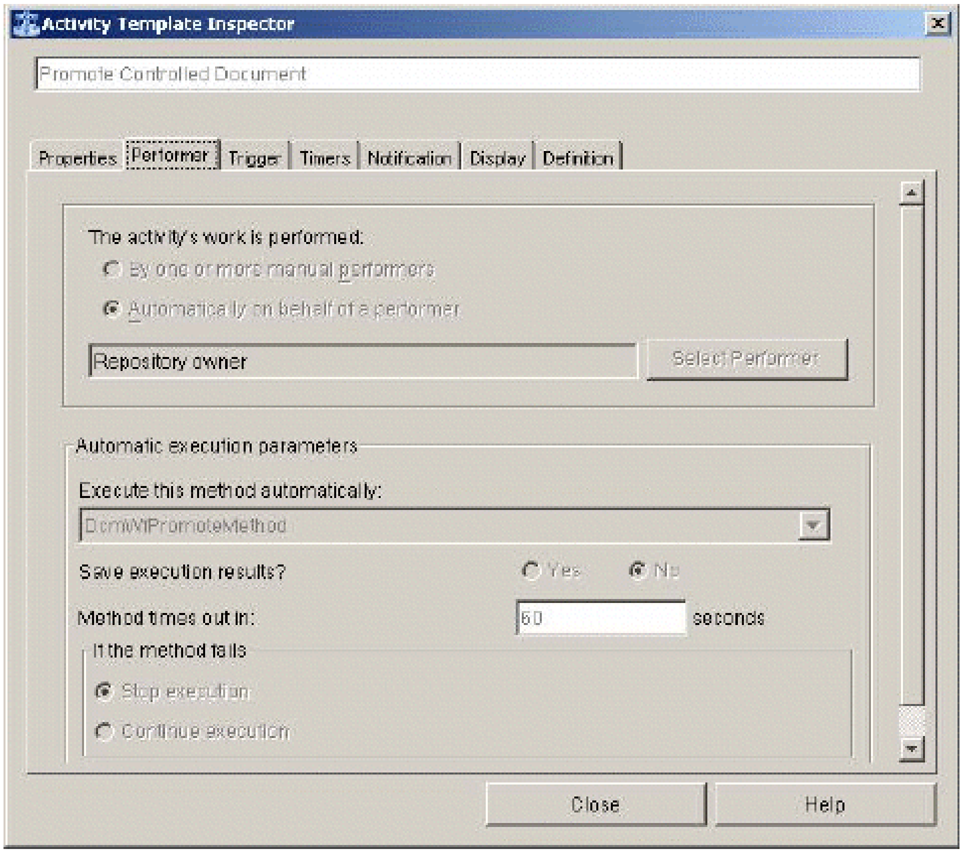
Activity ID: 4c4cc84680000779 (Installed)

Description:

Task name: (dcm_promote)

Task Instructions:

Figure 37. Use these settings to Promote Controlled Document on the Performer tab.



Promote Controlled Document

The activity's work is performed:

○ By one or more manual performers

● Automatically on behalf of a performer

Repository owner

Automatic execution parameters

Execute this method automatically: DcmWfPromoteMethod

Save execution results? ○ Yes ● No

Method times out in: 60 seconds

If the method fails

● Stop execution

○ Continue execution

Chapter 10

Setting Up Autonaming Schemes

This chapter explains how to set up and manage autonaming schemes, which are used to automatically assign a document name upon creation, checkin, or when the document is promoted to a certain lifecycle state. Topics discussed include:

Autonaming Schemes

Autonaming schemes allow you to predefine a prefix and number range to be automatically assigned to documents brought into the repository. A unique numbering scheme can be associated with any document class or business application. Once an autoname scheme has been defined for a particular document class, you can specify that the autoname will automatically become the name of document when the document is created or imported into the repository. You can also specify that an autonaming scheme be applied to a document only if that document reaches a certain lifecycle state, such as Approved.

When configuring a document class or business application, you can also specify that DCM assign a particular autonaming scheme when the document is approved or when it is promoted to a specific lifecycle state.

The unique name for each document will have a fixed alphanumeric prefix, an incrementing number within a valid range, and a fixed alphanumeric suffix. Each of these is optionally set with a value.

Document class owners and business application owners can create, modify, or delete autonaming schemes.

Creating Autonaming Schemes

The **Create Auto-naming Scheme** page allows you to predefine a prefix and number range to be automatically assigned to documents brought into the repository. This automatically assigned name and number becomes the name of a controlled document when the document is created or imported into the repository.

To create an autonaming scheme:

1. Navigate to **Administration > DCM > Auto-Namaing Schemes**.
2. To create an autonaming scheme, choose **File > New > Auto-naming Scheme**.
3. Refer to the table below for a description of the attributes.

Table 47. Create Autonaming Scheme Page Fields

Field Name	Value
Prefix	Used as part of the name
Suffix	Used as part of the name
Range: To	Specifies the starting number for the autonaming scheme.
Unlimited maximum range	If checked, allows autonumbers to be generated without an upper limit.
Range: From	Appears only when **Unlimited maximum range** is unchecked. Specifies the ending (maximum) number for the autonaming scheme.
Prefix Numeric Component with Zeroes	Specifies whether DCM automatically adds leading zeroes in the numerical prefix or suffix.

4. Click **Finish** to save your changes.
5. Verify that the changed autonaming scheme appears in the list of displayed schemes on the **Autonaming Schemes** page.

   The scheme will be renamed as a combination of *Prefix+Range:From–Range:To+Suffix*

Modifying Autonaming Schemes

Autonaming schemes allow you to predefine a prefix and number range to be automatically assigned to documents brought into the repository. This automatically assigned name and number becomes the name of a controlled document when the document is created or imported into the repository.

To modify an existing autonaming scheme:

1. Navigate to **Administration > DCM > Auto-Namaing Schemes**.
2. To modify an existing autonaming scheme, right-click on an auto-naming scheme and select **Properties**.
3. Refer to the table below for a description of the attributes.

Table 48. Autonaming Scheme Properties: Info Page Fields

Field Name	Value
Prefix	Used as part of the name
Suffix	Used as part of the name
Range: To	Specifies the starting number for the autonaming scheme.
Unlimited maximum range	If checked, allows autonumbers to be generated without an upper limit.
Range: From	Appears only when **Unlimited maximum range** is unchecked. Specifies the ending (maximum) number for the autonaming scheme.
Prefix Numeric Component with Zeroes	Specifies whether DCM automatically adds leading zeroes in the numerical prefix or suffix.

4. Click **OK** to save your changes.
5. To view which users, groups, and roles have permissions for this autonaming scheme, click the **Permissions** tab.
6. To add a user, group, or role to the list of users and groups with permissions on this autonaming scheme:

   a. Click the Add icon: ⊕.

b. Click the checkboxes associated with the names of the users, groups, or roles you want, then click **Add**.

c. Click **OK** when you have finished making your selections.

d. Click a radio button to assign a **Basic Permission** level.

e. Click one or more checkboxes to assign **Extended Permissions**.

f. Click **OK** to save your changes.

7. To change permissions for a user, group, or role already associated to this autonaming scheme:

   a. Click the checkbox associated with the user, group, or role whose permission you want to edit.

   b. Click the Edit icon: ✎.

   c. Optionally, click a radio button to change an assigned **Basic Permission** level.

   d. Optionally, click one or more checkboxes to change or add **Extended Permissions**.

   e. Click **OK** to save your changes.

8. To change the active permission set associated with this autonaming scheme, click the **Select** link, select another permission set from the **Choose a permission set** page, then click **OK**.

9. To view audit trail information for this autonaming scheme, click the **History** tab.

Deleting Autonaming Schemes

The owner of a business application can delete it from the system.

To remove an autonaming scheme:

1. Navigate to **Administration > DCM > Auto-Namaing Schemes**.

2. Click the checkbox associated with the autonaming scheme you want to delete.

3. Choose **File > Delete**.

Best Practices For Creating Autonaming Schemes

This section offers some recommendations for using autonaming schemes within your DCM configuration.

Using a Single Autonaming Scheme in Multiple Document Classes

We recommend that you create a unique autonaming scheme for each document class.

You can assign a single autonaming scheme to multiple document classes; however, if you do this, the numbering within each document class may not be sequential.

This happens because if you share autonaming schemes between document classes, the two classes will share the numbering, which may cause the numbering in a particular doc class to have gaps, since each generated autoname is unique per scheme rather than per document class.

Assigning Autonaming Schemes to Specific Lifecycle States

When configuring document classes or business applications, you can specify that DCM apply particular autonaming schemes to documents depending on which lifecycle state they are in. For example, if you specify that an autonaming scheme be applied only if a document reaches the Approved state, you will prevent gaps from occurring in your document numbering sequence due to rejected or unapproved documents.

Chapter 11

Notifications

Notifications prior to DCM 6.0 were configured using an XML configuration file. Notifications are now configured from a user interface instead of using an XML configuration file.

Administering notifications

Notifications are intended to prompt a response against a task, such as a **To be Read** or **Signoff required** notification, and to advise of any follow-up actions and possible errors know as event notifications, user and system triggered events as described in *About Notifications* under *DCM key concepts and features* in the *Introduction*.

Notifications are scoped to a Document Class or Business Application. Signatories for **Signoff required** notifications assigned to a particular Document Class for example, are set on the **Signoff** tab. The **TBR** tab sets the priority of a TBR notification. The distribution list for a TBR is set on the lifecycle state of a particular Document Class or Business Application whereas the distribution list for Signoff required notifications is set on the **Signoff** tab when you edit a scope, Document Class or Business Application. Separate procedures are provided below to administer TBRs.

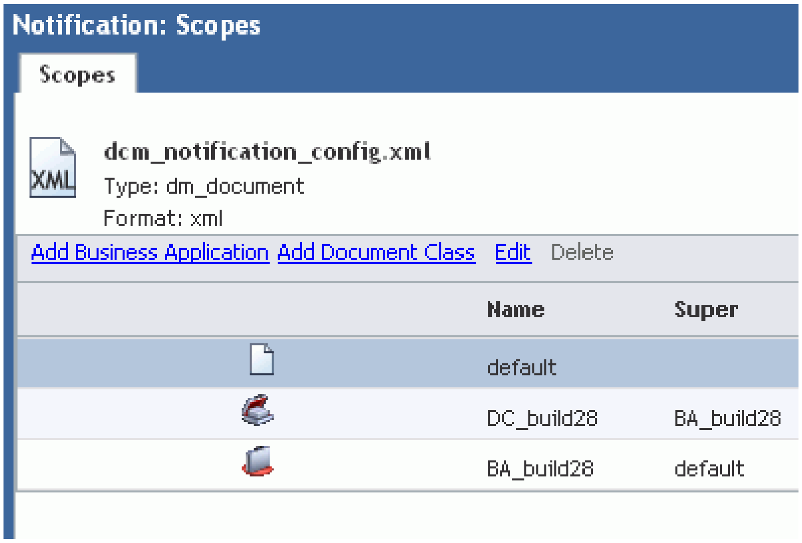
Use the **dcm_notification_config.xml** file according to the procedure provided below to administer notifications. Its available out-of-the-box and contains the **default** scope with all of the settings pre-configured for you. The **default** scope can NOT be deleted when selected as can any other scope that may be listed below it. The procedure can be used to **Edit**, **View**, or **Unlock** the file. Using the **Edit** option, you can create a new scope by adding a business application and/or a document class, and when necessary modify the settings for any of the scopes listed.

Scopes, other than the **default** scope, when edited include options to **Enable** or **Inherit** on each of the tabs displayed. The **default** scope however does not include the **Inherit** option on any of its tabs, see for example, the TBR tabs illustrated below. Settings are inherited from the **Super** when the **Inherit** option is selected. The **Super** specifies the default if the document class or business application is not added/listed. There is no lookup for the **Super** if the **Inherit** option is turned off (the checkbox is deselected). When it is turned on however, lookup is conducted against the three layers:

- Top - default
- Middle - business application
- Bottom - document class

For example, if a configuration at the document class level has **Inherited** turned on, DCM will use the same document class parent configuration which is its business application. If the parent business application is not configured, DCM will use the **default** configuration/scope. The **default** scope for this reason can not be deleted.

Figure 38. Scopes listed for dcm_notification_config.xml



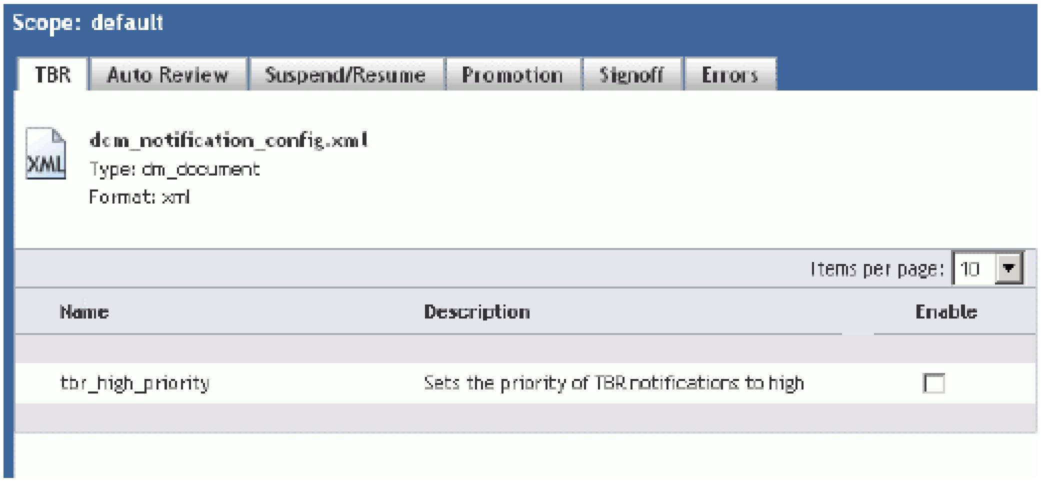
Notification: Scopes

dcm_notification_config.xml
Type: dm_document
Format: xml

	Name	Super
	default	
	DC_build28	BA_build28
	BA_build28	default

The following screen is displayed when you **Edit** a scope.

Figure 39. Editing the default scope



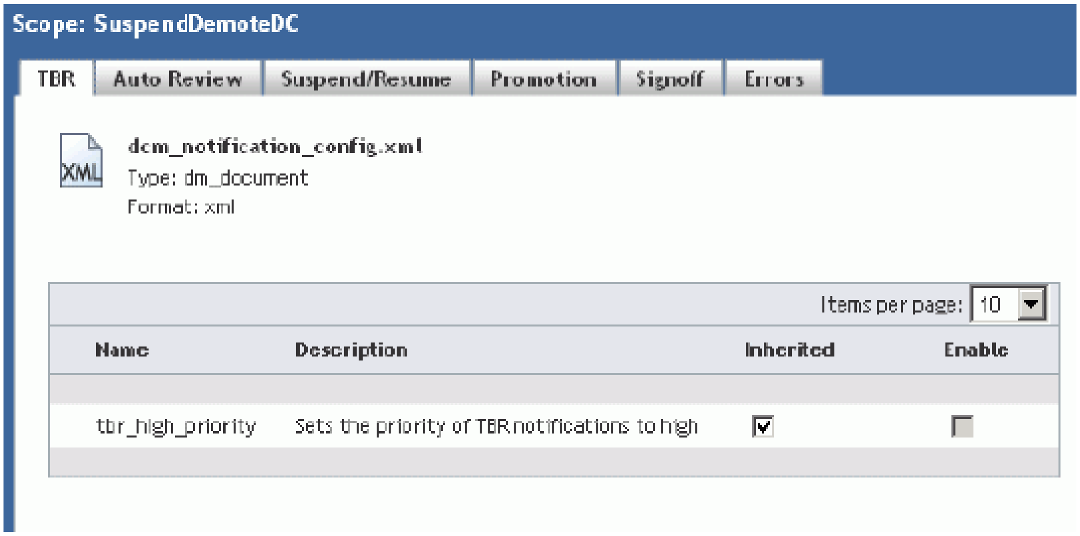
Scope: default

dcm_notification_config.xml
Type: dm_document
Format: xml

Items per page: 10

Name	Description	Enable
tbr_high_priority	Sets the priority of TBR notifications to high	□

Figure 40. Editing a scope other than the default scope



Scope: SuspendDemoteDC

dcm_notification_config.xml
Type: dm_document
Format: xml

Items per page: 10

Name	Description	Inherited	Enable
tbr_high_priority	Sets the priority of TBR notifications to high	☑	□

To create a scope or to view or modify settings of a scope contained in the notification configuration file:

Notifications are configured/scoped according to the default scope for a document class or business application if no other scopes are listed in the notification configuration file under the **default** scope.

1. Navigate to **DCM > Administration > Notification**.
2. Right-click **dcm_notification_config.xml** displayed in the content pane and select:

- **Edit**: to create a new scope or to modify an existing scope. The xml document format of the scope is checked out whenever it is edited. The xml is checked in when the administrator saves the changes.
- **View**: to view the xml file with no checkout/checkin actions.
- **Unlock**: to cancel checkout if the xml file is inadvertently locked. The xml file gets locked if the user inadvertently exits the Edit mode or closes the browser. The xml in such instances remains checked out and needs to be checked in with a cancel-checkout to restart the edit.

3. Optionally, click **Add Business Application** or **Add Document Class** to create new scope when necessary.
4. Right-click an existing scope and select **Edit** to modify it or when necessary, **Delete** to dispose of it. The **default** scope is required and can NOT be deleted since the option for it is not available as it is for other scopes added/listed.

   Any scope selected for **Edit** is checked out and locked and can only be unlocked when it is checked in by the user who checked it out. Other users can only view it when it is checked out and locked.
5. Modify the settings on the applicable tab(s) for the notification type(s) you want to affect and then click **OK**. Select or deselect the checkboxes on the applicable tab to modify the settings.

   Settings are inherited by default from the **Super** displayed for the selected scope.
6. Click **Save** to accept changes for the scope(s) edited/modified.
7. Click **Save New** to complete the process and save all changes against **dcm_notification_config.xml**.

   The new notification configuration file displayed in the content pane is identified with a new CURRENT version number.

   Right-click **dcm_notification_config.xml** if it is locked, and select **Unlock** to checkin the file.

Chapter 12

PDF Stamping Service

This chapter describes the following topics:

PDF stamping service overview

The PDF Stamping Services is one of the fundamental components of DCM that supports electronic signature capture and manifestation. Customers have the ability to apply an electronic signature to a piece of content in a way that provides an indisputable record of a user approving (signing off according to 21 CFR part 11 compliance) a piece of content. Users:

- are able to view all signatures applied to a particular version of content, on a signature page attached to the PDF rendition of the content, either at the beginning or at the end of the rendition
- can verify if the rendition to which the initial signature page is applied is the actual rendition (content that has not changed since the rendition) of the content in whichever format the content is in
- can also verify that both the content and the associated signed PDF rendition are not altered in any way from the initial signature through to the final signature and thereafter until disposition.

There are two functional components to PSS:

1. electronic signatures
2. watermark stamping

The following two DARs must be installed, in the order listed after the DCM DAR is installed, for PSS to be fully functional:

- pssEsign

- pssStamp

PSS manages PDF documents by allowing you to define PDF templates to control the appearance of published documents when they are viewed or printed. PSS policies and templates are stored in the DCM repository.

Roles for configuring PDF Stamping Service (PSS)

A user must be in one of the following three administrator roles to access PSS configurations:

- Business Application Owner
- Document Class Owner
- Functional Area Supervisor

A user account which is capable of configuring PSS must be created with the following rights/permissions:

- Documentum Administrator role
- Have WRITE access to the following Documentum System cabinet folder and files:

  /System/Application/pss

About Banners

A banner allows for inclusion of additional information in the document's headers and footers. This information includes a combination of attribute and non-attribute data such as User Name, Print Date, and Time. It also allows for custom labeling of attribute fields.

About Watermarks

A watermark is a text object that is placed on a document when the document is printed. Configuration options include the content and location of the watermark. The banner and watermark information is accessed each time a user views/prints a document.

About Controlled Viewing

A document administrator can configure different combinations of document views and print views to users and groups. These different views may include additional text, graphics, barcodes, watermarks and properties from the repository.

For example, the same PDF rendition may be presented to one user with printing disabled and to another user with printing allowed. For the user who is allowed to print, a print time overlay can be applied that marks the document as an "Uncontrolled Print." This uncontrolled print can also have the user's name, any document properties, and the print time recorded on the printed copy.

Configuring PDF Stamping Service

PDF Stamping Service is used to apply FDA 21-CFR Part 11 compliant electronic signatures on PDF renditions of documents. E-signatures can be configured with any valid DCM lifecycle and can function without DCM or a lifecycle being attached to a piece of content. The Electronic Signature Module can operate within any supported Content Server (CS) environment. Signatures are stored in an audit trail record that also contain information regarding the electronic signature and the previous valid signature if present. This audit trail record is the official record of the electronic signatures.

Each new PSS Configuration object created according to this procedure defines a PDF template. You can create a variety of PDF templates as needed to control the appearance of published documents when they are viewed or printed. You can also associate a lifecycle to a PDF template if necessary when you follow the procedure used to customize a PDF template.

To create a PDF template:

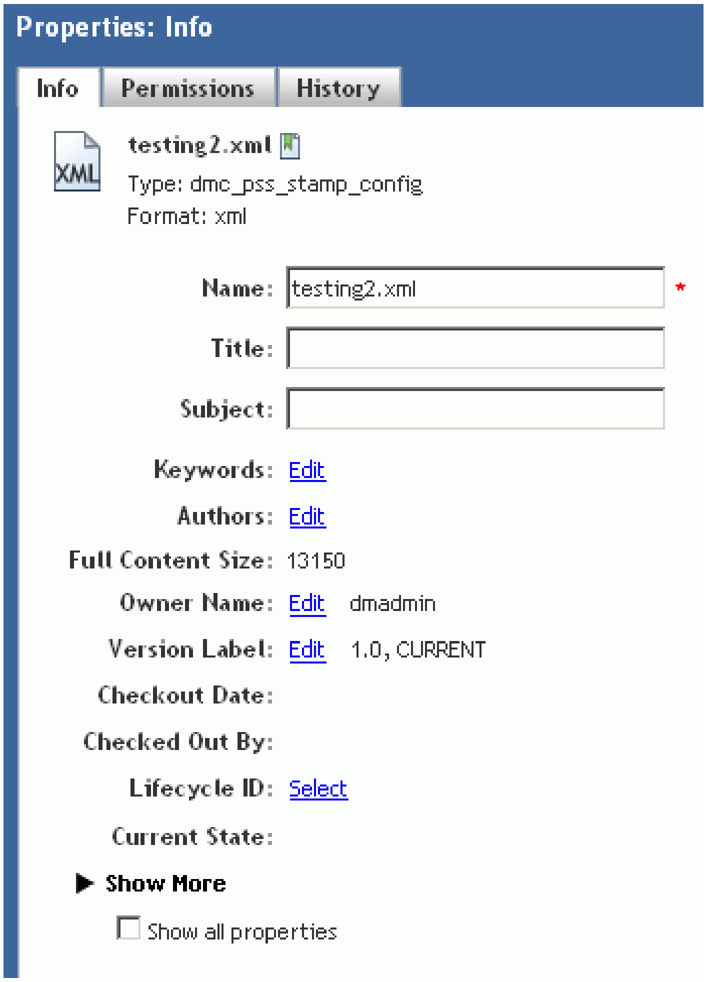
1. Navigate to **Administration > DCM > PDF Stamping Service**.
2. Select **File > New > PSS Configuration**.
3. Enter a unique value for the mandatory **Name** that is different from those PDF templates already listed in the content pane.
4. Select a value for the **Type** from the list box if the default value displayed is not desired.
5. Click **OK**.

To customize a PDF template (PSS Configuration object):

1. Navigate to **Administration > DCM > PDF Stamping Service**.

2. Select a PDF template listed in the content pane and click **View** > **Properties** > **Info**.

Figure 41. Customizing a PDF template



testing2.xml
Type: dmc_pss_stamp_config
Format: xml

Name: testing2.xml *
Title:
Subject:
Keywords: Edit
Authors: Edit
Full Content Size: 13150
Owner Name: Edit dmadmin
Version Label: Edit 1.0, CURRENT
Checkout Date:
Checked Out By:
Lifecycle ID: Select
Current State:

▶ Show More

□ Show all properties

3. Click **Ok** when you are done entering the preferred values on the **Info** tab and if necessary on the **Permissions** tab.

Choosing between Acrobat Form Based or Tagged Content Based templates

Liquent uses a tagged content based template to implement PDF watermarks. Tagged content is used to embed pre-defined tags along with the tag name into PDF template content. During the PDF watermark stamping, a runtime value is used to replace each tag and its name. The runtime values on the electronic signature page using the default sample template available out-of-the-box, is listed under "**/System/Applications/pss**".

For instance, a tag defined as "<attribute object_name>" is replaced as "SOP1" for an object with object_name as "SOP1" at the PDF template and this template is stamped on given PDF files.

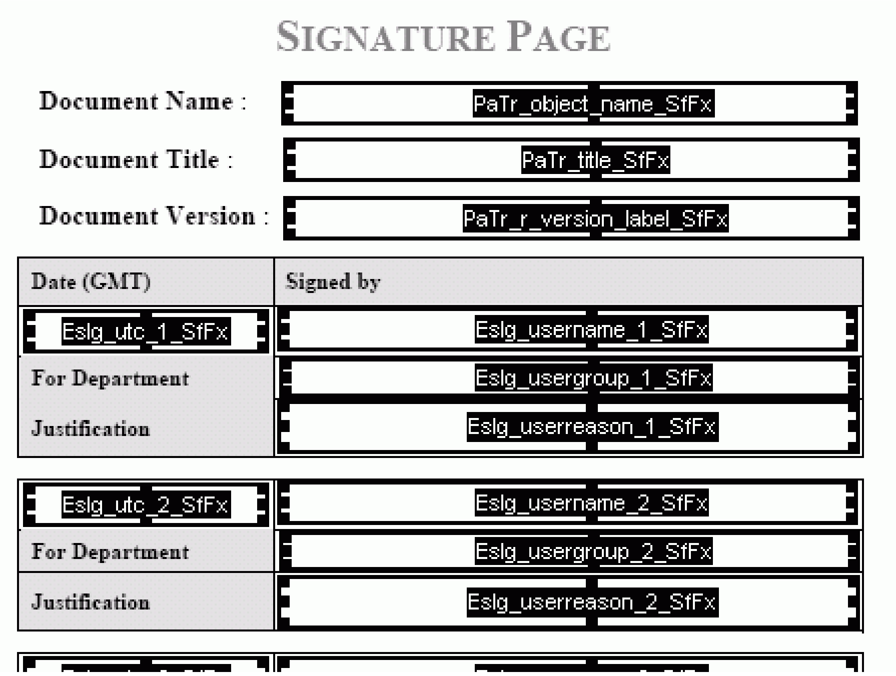
This approach is possible when PSS uses Acrobat SDK though is not recommended at all when PSS uses iText. This is because iText is not designed to edit PDF content since PDF format itself is not designed as a format for editing. Refer to "iText IN ACTION", page 578 for details. Using PSS with iText as the PDF library, Acrobat Form is used for dynamic value population.

To create an Acrobat Form Field within the template:

1. Convert a MS word template to PDF.
2. Add eSignature field into the PDF.

   On Acrobat Professional (using Acrobat 8.0 for example), either click on the Text Form icon or go to **Tools > Form Tools > Text Field Tool**.

Figure 42. Default_EsignPage.pdf template displayed on Adobe Acrobat Professional



SIGNATURE PAGE

Document Name : PaTr_object_name_SfFx

Document Title : PaTr_title_SfFx

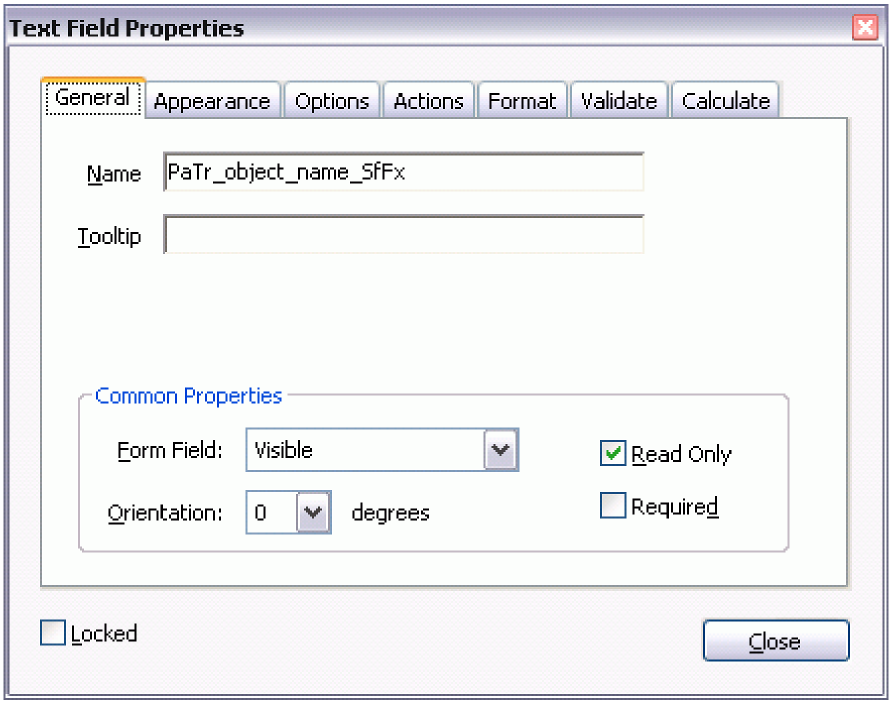
Document Version : PaTr_r_version_label_SfFx

Date (GMT)	Signed by
Eslg_utc_1_SfFx	Eslg_username_1_SfFx
For Department	Eslg_usergroup_1_SfFx
Justification	Eslg_userreason_1_SfFx

Eslg_utc_2_SfFx	Eslg_username_2_SfFx
For Department	Eslg_usergroup_2_SfFx
Justification	Eslg_userreason_2_SfFx

3. Double-click on the new added Text Form Field to edit and set the values for the properties you want to affect. Select the tab as displayed in the following screenshot examples to edit, and click **Close** when done.

Figure 43. Editing PaTr_object_name_SfFx, text field for Document Name



Name: PaTr_object_name_SfFx

Tooltip:

Form Field: Visible

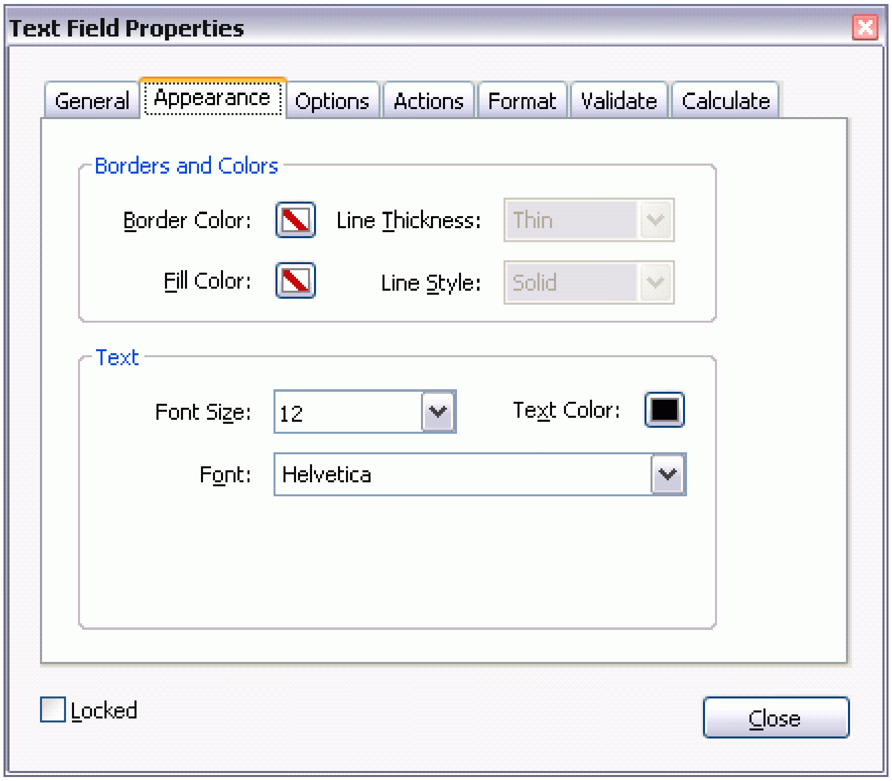
Orientation: 0 degrees

☑ Read Only

□ Required

□ Locked

Figure 44. Editing Against Appearance



Borders and Colors

Border Color:

Line Thickness: Thin

Fill Color:

Line Style: Solid

Text

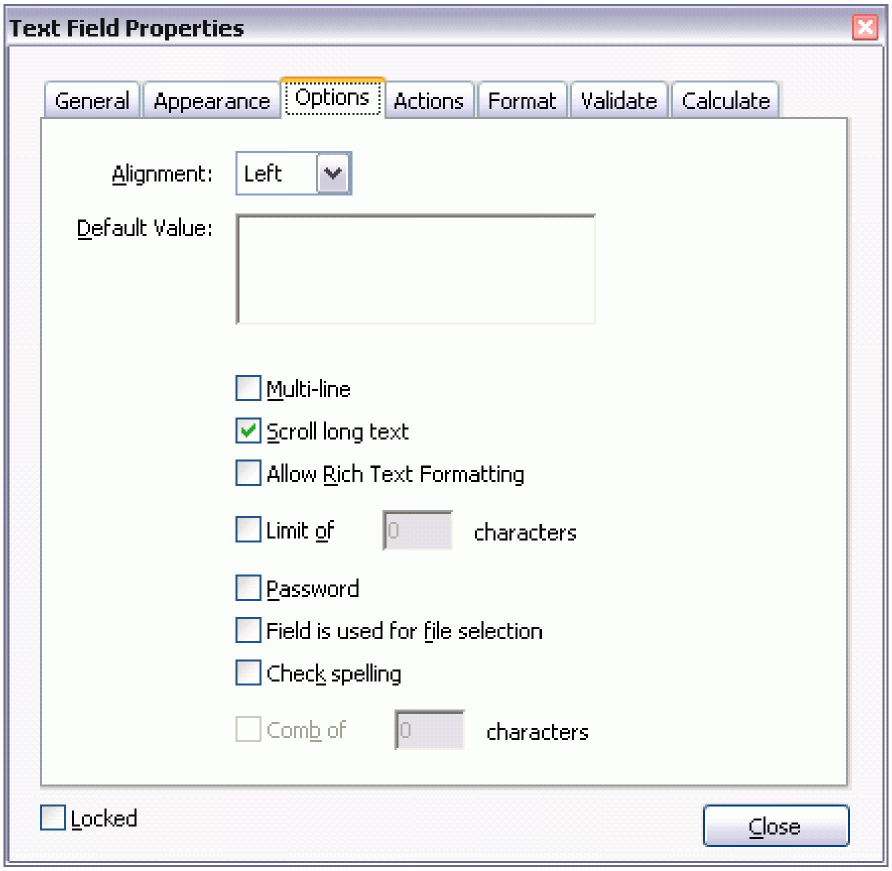
Font Size: 12

Text Color: ■

Font: Helvetica

□ Locked

Figure 45. Editing Against Options



Text Field Properties

Alignment: Left

Default Value:

□ Multi-line

☑ Scroll long text

□ Allow Rich Text Formatting

□ Limit of 0 characters

□ Password

□ Field is used for file selection

□ Check spelling

□ Comb of 0 characters

□ Locked

PSS eSign template configuration

PSS uses Documentum **dmc_pss_esign_config** objects to configure eSign.

Table 49. Attributes for object type "dmc_pss_esign_config" under its super type "dm_document"

Attribute	Type	Description
append_to_body	Boolean	True/False – prepend/ append eSiganture page.
document_type	String(40) Repeating	Supported object types.
max_signatures	Integer	Total maximum signatures allowed.

Attribute	Type	Description
max_signatures_per_page	Integer	Maximum signatures on one page.
pdf_version	String(8)	Used to specify a PDF version for signed PDF. *1.5; 1.6* and *1.7* are supported string values for this attribute. If any other string value is configured PSS keeps the PDF version of signed PDF as the same as the original PDF.
simple_date_format	String(128)	Specify a Java SimpleDateFormat pattern for the time information populated at the signature page.
runtime_ids	ID	Signing Object Id at runtime. PSS uses this attribute to prevent more than one user signing the same document concurrently.
reuse_pageno	Integer	For future support. Used to specify the page number that will be reused when the current signature page is full. This attribute will be used when PSS supports multiple pagestemplates. When configuring the rules, each configuration must: • • Has only one PDF rendition with empty page modifier, and this PDF must have only one page. PSS automatically re-use the same page when current signature page is full and the number

Attribute	Type	Description
		of total signatures is less than pre-defined total maximum signature. - Has supported object type(s) defined - Has total maximum signatures defined - Has maximum signature per page defined.
enable_compression	Boolean	True/False – Compress or not to compress the eSigned PDF. Compression can be enabled only if you select PDF 1.5 or higher as the signed PDF version. Earlier versions of PDF do not allow for full compression.
language_code(dm_ sysobject default attribute)	String(5)	This dm_sysobject default attribute is used to support multi-language. Refer to Documentum System Object Reference Appendix A for Language code. If a CJK(Chinese Japanese Korean) code is specified, the **pss.esign.default. font.XXX** and **pss.esign. default.encode.XXX** for the corresponding language specified in **pss.properties** will be used to manifest the fields on the signature page. **Note:** PSS with DCM 6.5 supports font and encoding provided by iText BASE FONT. Some special characters in certain font and encoding combinations may not be

Attribute	Type	Description
		displayed when changed to another font and encoding combination.

PSS stamping configuration for view; export and print

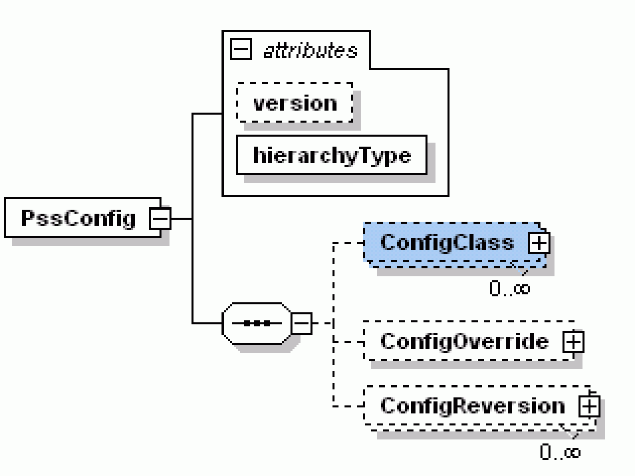
XML is used as the configuration file format. Each configuration file must be object type **dmc_pss_stamp_config**. The **dmc_pss_stamp_configobject** type is a subtype of **dm_document** without any custom attributes in the current release.

PSS XML configuration schema design

PssConfig

PssConfig is the top node of PSS stamping configuration.

Figure 46. PssConfig



Attribute

- **hierarchyType**: defines the hierarchy type of **ConfigClass**. PSS supports two hierarchy types - **DctmObjectType** which uses Documentum Object Type hierarchy, and **DcmDocClass** which uses both DCM Business Application and Document Class hierarchy.
- **version**: configuration version/reversion support reserved for future use.

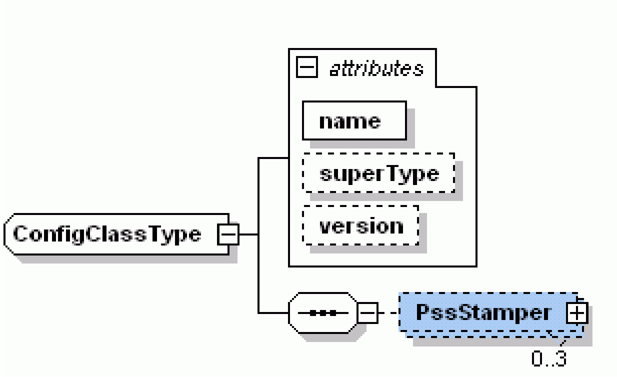
Sub-nodes

- **ConfigClass**: A ConfigClassType node.
- **ConfigOverride**: An OverridePluginType node.
- **ConfigReversion**: A ConfigRevisionType node.

ConfigClassType

ConfigClassType defines all configuration parameters for a specified **ConfigClass**, **dm_document** or **dm_sysobject** for **DctmObjectType** hierarchy type for example.

Figure 47. ConfigClassType



Attribute

- **name**: The name of **ConfigClass**. For **DctmObjectType** hierarchy type, It has to be a valid repository object type name, e.g. dm_document. For DcmDocClass, it has to be the name of an existing Business Application or Document Class.

- **superType**: The super type of current **ConfigClass**. The super type of **dm_document** is **dm_sysobject**.
- **version**: Configuration version/reversion support reserved for future use.

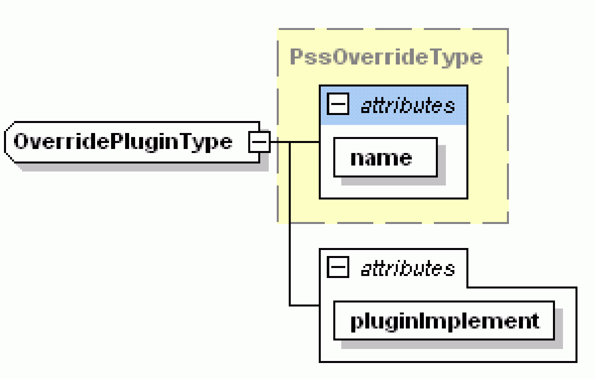
Sub-nodes

- **PssStamper**: A PssStamperType node

OverridePluginType

OverridePluginType contains **ConfigClass** override plug-in definition

Figure 48. OverridePluginType



Attribute

- **name**: The name of the plug-in.
- **pluginImplement**: The java implementation class of the plug-in. The plug-in java class must implement **IPssOverridePlugin** interface.

Sub-nodes

N/A

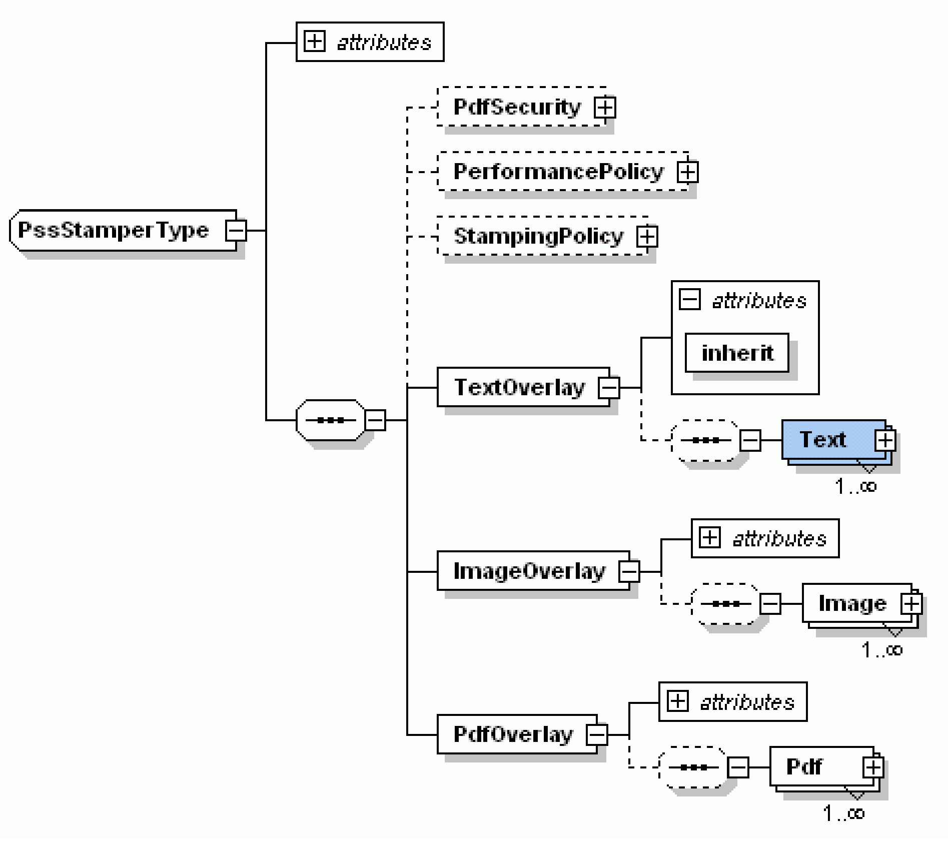
ConfigReversionType

Configuration version/reversion support reserved for future use.

PssStamperType

PssStamperType defines the configuration for an overlay stamper for a certain action. Supported actions are "View; Export and Print".

Figure 49. PssStamperType



Attribute

- **action**: the action of the current stamper. PSS supports three actions: "View, Export and Print".

- **pdfVersion**: The PDF version of stamped PDF file. PSS supports PDF version 1.5, 1.6 and 1.7.
- **version**: configuration version/reversion support reserved for future use.

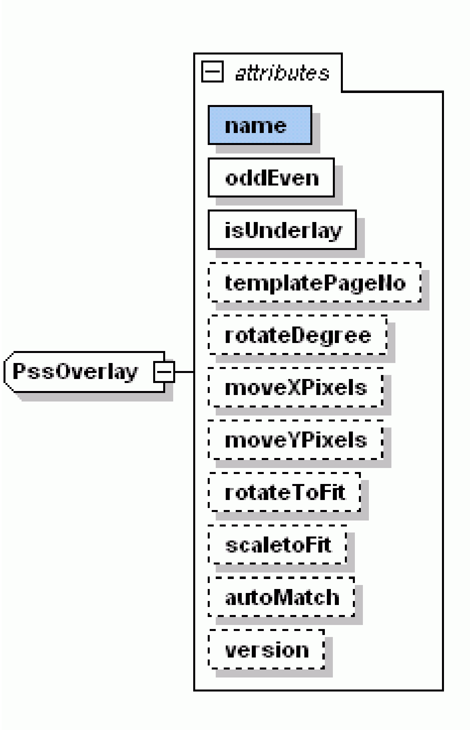
Sub-nodes

- **PdfSecurity**: a **PdfSecurityType** node
- **PerformancePolicy**: a **PerformancePolicy** node
- **StampingPolicy**: a **StampingPolicy** node
- **TextOverlay**: defines text overlay information for this stamper. The text overlay(s) could be inherited from super type, which controlled by a Boolean attribute "inherit", or override through "Text", which is a **TextOverlayType** node.
- **ImageOverlay**: defines image overlay information for this stamper. The image overlay(s) could be inherited from super type, which controlled by a Boolean attribute "inherit", or override through "Image", which is an **ImageOverlayType** node
- **PdfOverlay**: define PDF overlay information for this stamper. The PDF overlay(s) could be inherited from super type, which controlled by a Boolean attribute "inherit", or override through "Pdf", which is a **PdfOverlayType** node.

PssOverlay

PssOverlay defines all general options for an overlay. TextOverlayType, ImageOverlay and PdfOverlayType are its sub-type.

Figure 50. PssOverlay



Attribute

- **name**: name of the overlay.
- **oddEven**: An enumeration type with supported values as All, Odd, and Even to determine whether the overlay/watermark applies to odd, even, or all pages of the stamped PDF.
- **isUnderlay**: A Boolean type to indicate if the overlay/watermark is stamped over or under the original content.
- **templatePageNo**: only applies to PdfOverlay. If PDF overlay template is multi-pages, which page is used when stamping. If a zero page number has been specified, PSS uses the current page size of the stamped PDF file to find a close-match template page automatically.
- **rotateDegree**: the angle to rotate the Text Overlay or Image Overlay. Rotation of PDF Overlay is not supported.
- **moveXPixels**: x position in pixels from bottom left to apply the overlay.
- **moveYPixels**: y position in pixels from bottom left to apply the overlay.

- **rotatetoFit**: a Boolean to indicate if rotates the overlay to fit the stamped PDF. This attribute is reserved for future use.
- **scaletoFit**: a Boolean to indicate if scales the overlay to fit the stamped PDF. Currently this is only used for image form at PDF overlay.
- **autoMatch**: a Boolean to indicate if automatically matches the overlay for the stamped PDF. This attribute only applies to PDF overlay. This attribute is reserved for future use.
- version: configuration version/reversion support reserved for future use.

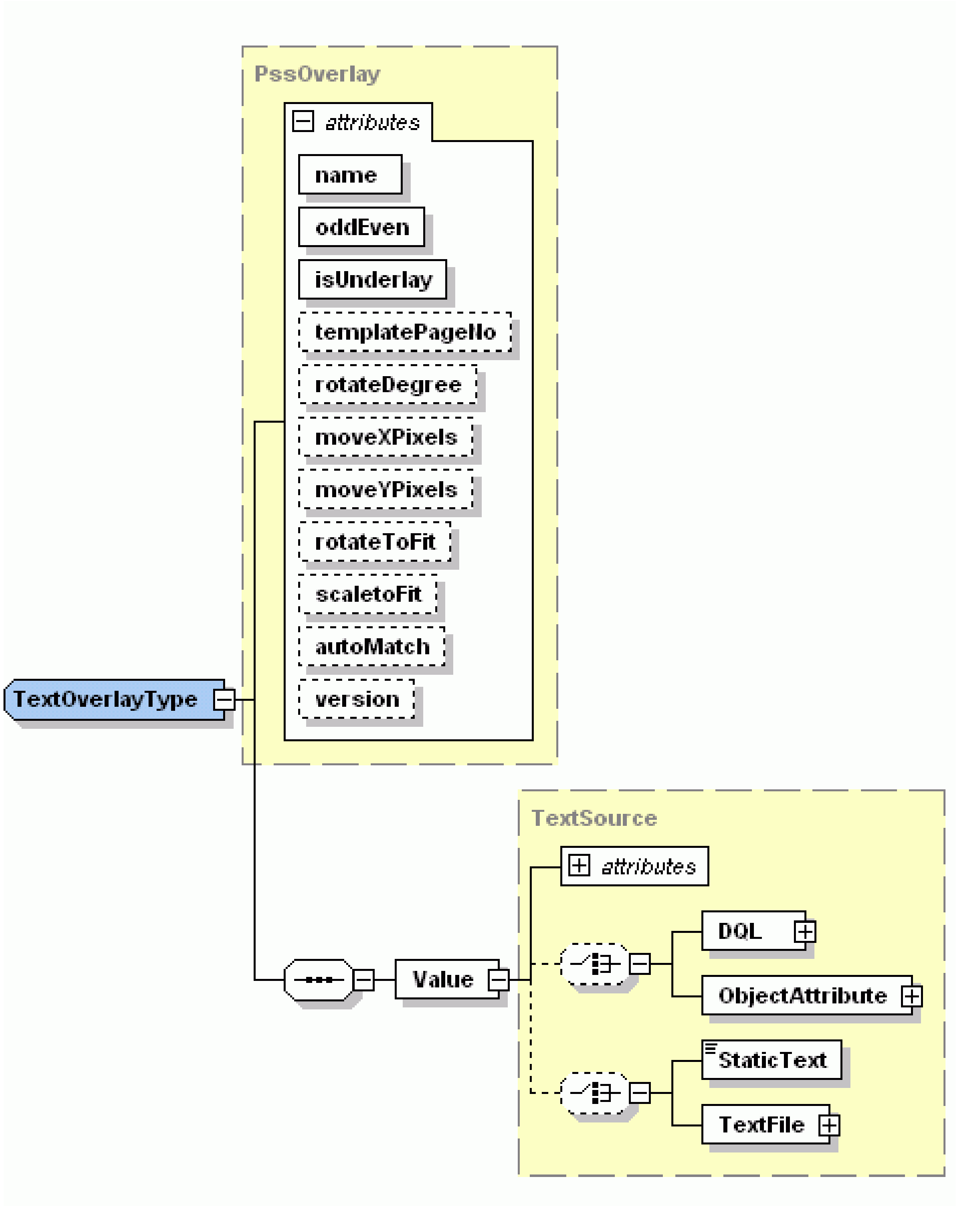
Sub-nodes

N/A

TextOverlayType

TextOverlayType defines the text overlay information, a sub-type of **PssOverlay**

Figure 51. TextOverlayType



Attribute

Defined in PssOverlay.

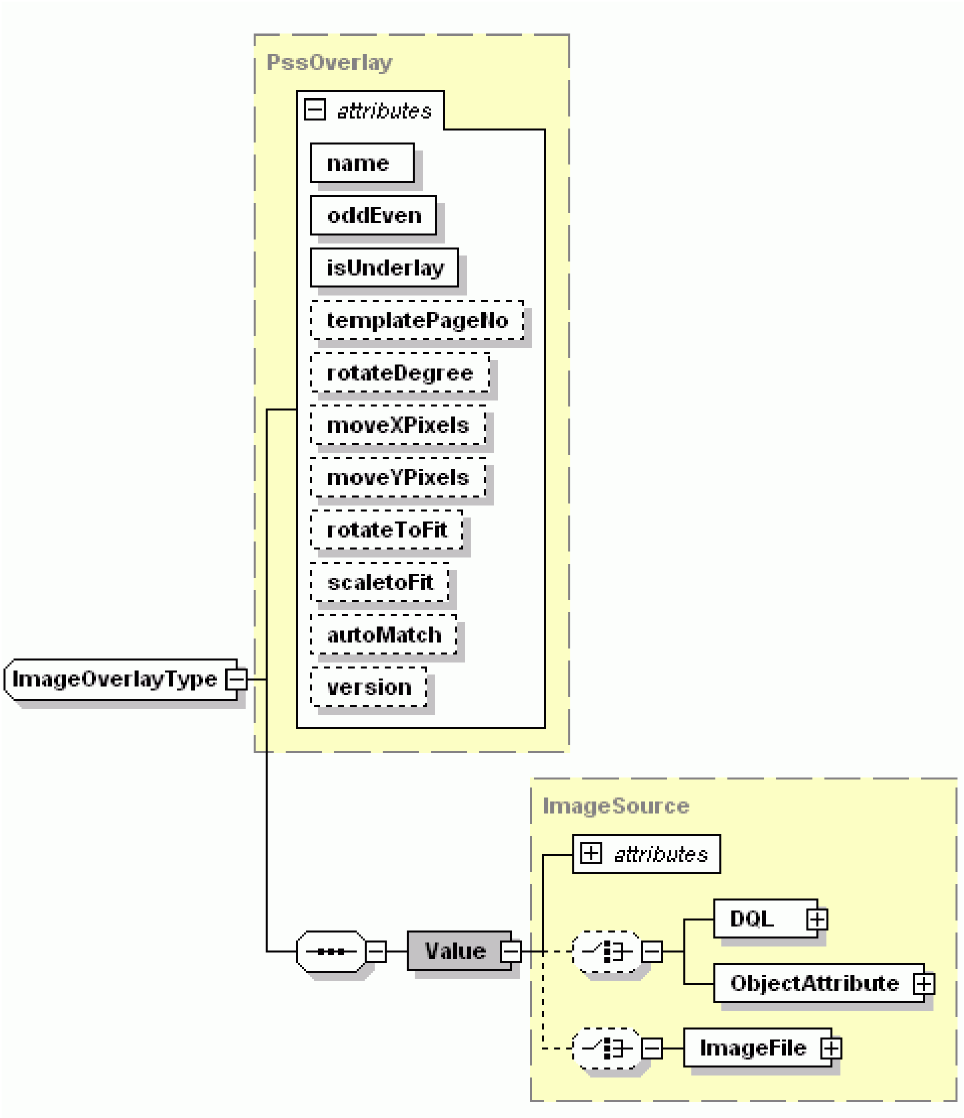
Sub-nodes

- **Value**: A TextSource node.

ImageOverlayType

ImageOverlayType defines the image overlay information. It is a sub-type of PssOverlay

Figure 52. ImageOverlayType



Attribute

Defined in PssOverlay.

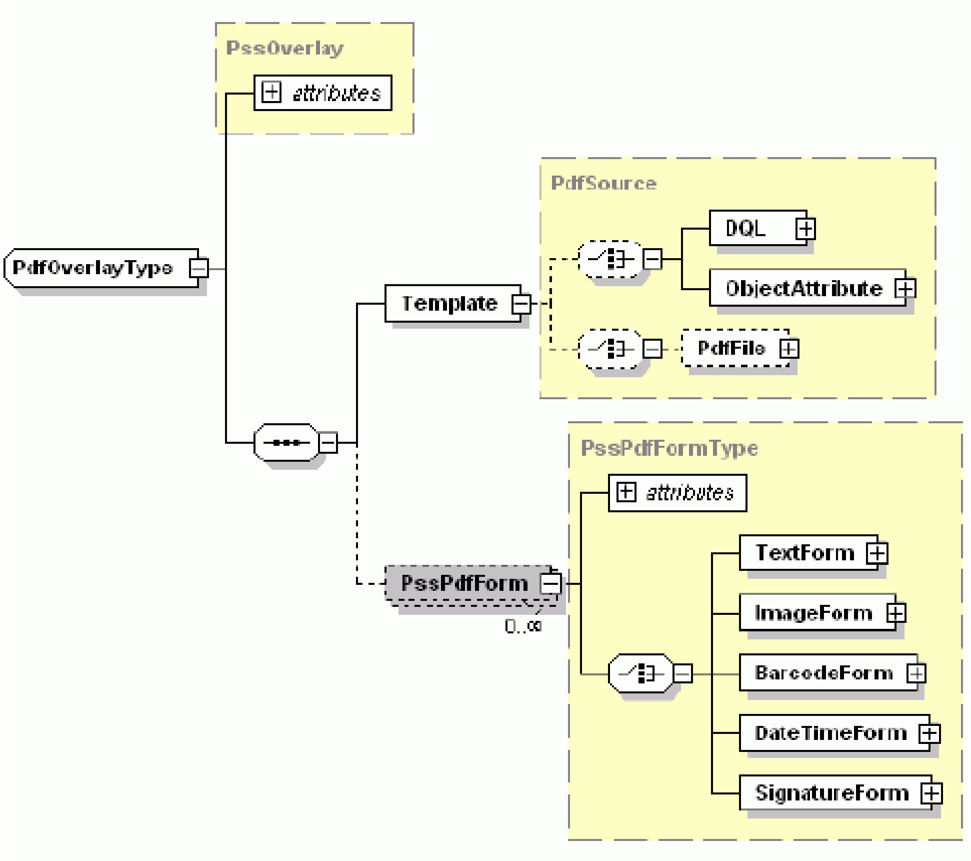
Sub-nodes

- **Value**: A ImageSource node

PdfOverlayType

PdfOverlayType defines the PDF overlay information, a sub-type of **PssOverlay**.

Figure 53. PdfOverlayType



Attribute

Defined in PssOverlay.

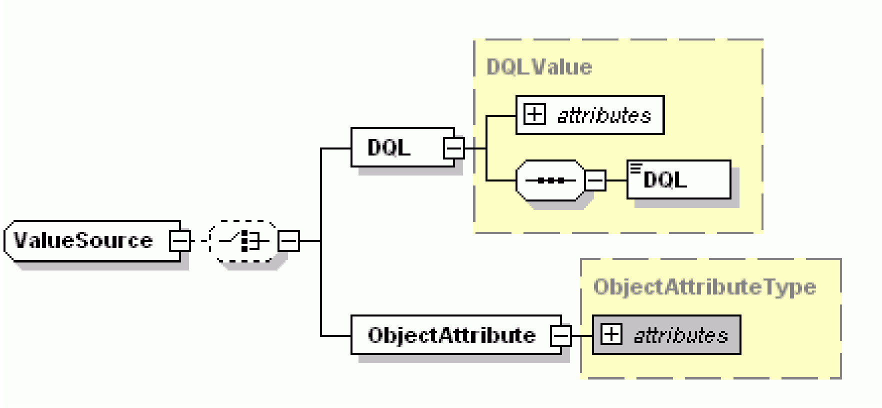
Sub-nodes

- **Template**: PDF overlay template, which is a PdfSource node.
- **PssPdfForm**: A PssPdfFormType node.

ValueSource

ValueSource defines either a DQL or a Docbase Object Attribute data source.

Figure 54. ValueSource



Attribute

N/A

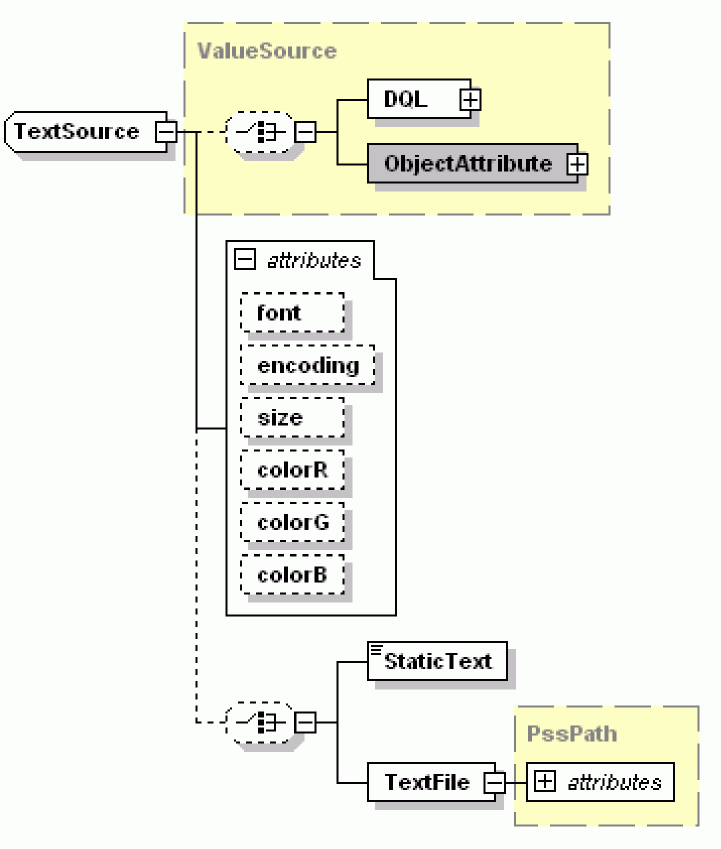
Sub-nodes

- **DQL**: A DQLValue node.
- **ObjectAttribute**: An ObjectAttributeType node.

TextSource

TextSource defines the options for text data source, a sub-type of **ValueSource**. It has four data source types: "DQL" and "ObjectAttribute", which are inherited from ValueSource, as well as "StaticText" and "TextFile".

Figure 55. TextSource



Attribute

- **Font**: font name of the text.
- **Encoding**: encode type of text.
- **Size**: text size.
- **colorR**: text color value of Red.
- **colorG**: text color value of Green.
- **colorB**: text color value of Blue.

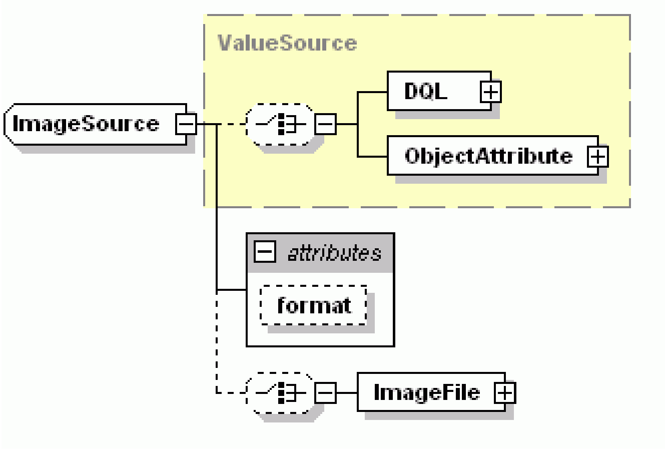
Sub-nodes

- **DQL**: inherit from ValueSource.
- **ObjectAttribute**: inherit from ValueSource.
- **StaticText**: a text xml node that define static overlay text.
- **TextFile**: a PssPathType node.

ImageSource

ImageSource defines the options for image data source, a sub-type of **ValueSource**. It has three data source types: "DQL" and "ObjectAttribute", which inherited from ValueSource, "ImageFile".

Figure 56. ImageSource



Attribute

- **format**: image format, jpg and gif for example.

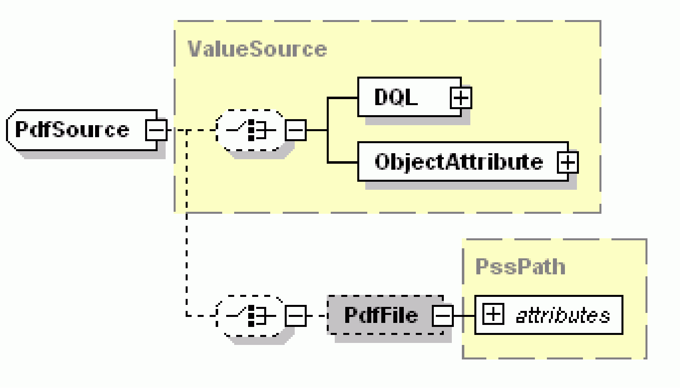
Sub-nodes

- **DQL**: inherit from ValueSource.
- **ObjectAttribute**: inherit from ValueSource.
- **ImageFile**: a PssPath node.

PdfSource

PdfSource defines the options for PDF data source, a sub-type of **ValueSource**. It has 3 kinds of data source: DQL and ObjectAttribute, which inherited from ValueSource, PdfFile.

Figure 57. PdfSource



Attribute

N/A

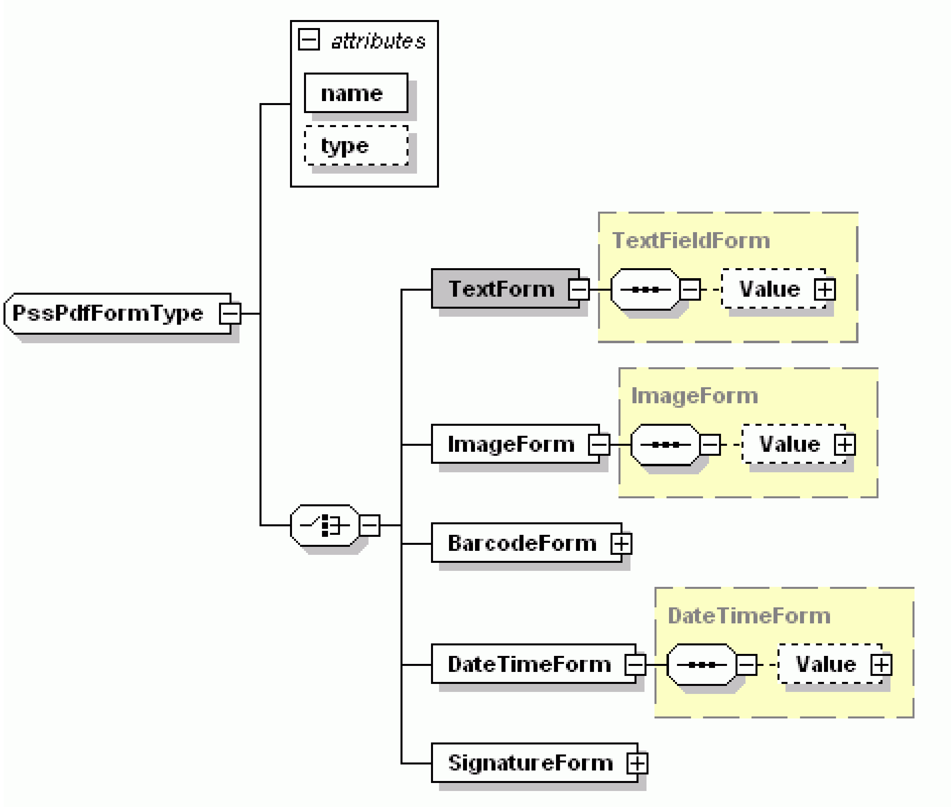
Sub-nodes

- **DQL**: inherit from ValueSource
- **ObjectAttribute**: inherit from ValueSource
- **PdfFile**: a PssPathType node

PssPdfFormType

PssPdfFormType define a sequence of PDF forms for one overlay template. PSS fill these forms at run-time and apply the overlay template to the overlaid document.

Figure 58. PssPdfFormType



Attribute

- **name**: name of the PDF form.
- **type**: only for JAXB customization.

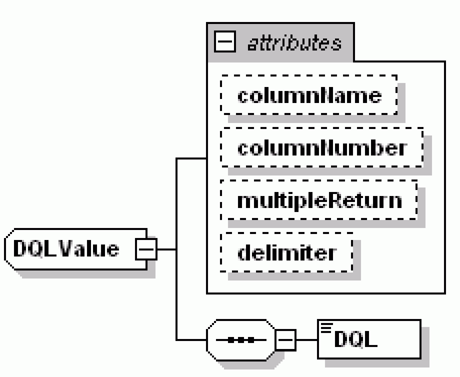
Sub-nodes

- **TextForm**: A TextFieldForm node.
- **ImageForm**: A ImageForm node.
- **DateTimeForm**: A DateTimeForm node.
- **BarcodeFrom**: for future support.
- **SignatureForm**: for future support.

DQLValue

DQLValue defines the options for the DQL data source and how to populate DQL execution results.

Figure 59. DQLValue



Attribute

- **columnName**: the column name in DQL that used to populate the value.
- **columnNumber**: the column number (start with 0) in DQL that used to populate the value. If columnName and columnNumber are set together, columnName has the higher priority to populate the data. If none of them set, column 0 will be picked up.
- **multipleReturn**: a Boolean value to specify if there is more than one rows in DQL Result set. If not set or set wrong, the first row will be picked up.
- **delimiter**: only apply when multipleReturn is set. If set and more than one rows return, use this delimiter to merge the result.

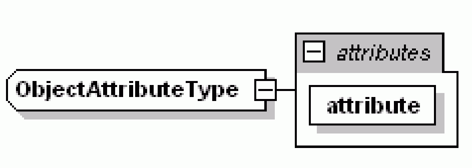
Sub-nodes

- **DQL**: a text node to set DQL statement.

ObjectAttributeType

ObjectAttributeType define the options to populate an attribute value against overlaid Docbase object.

Figure 60. ObjectAttributeType



Attribute

- **attribute**: the attribute name of the over laid object.

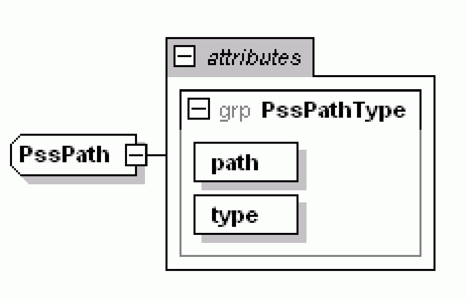
Sub-nodes

N/A

PssPath

PssPath defines all supported file paths.

Figure 61. PssPath



Attribute

- **path**: string of path.

- **type**: the type of the path. PSS supports 3 types:
  - OS: file system file path
  - Documentum: Docbase file path
  - URI: URI fie path

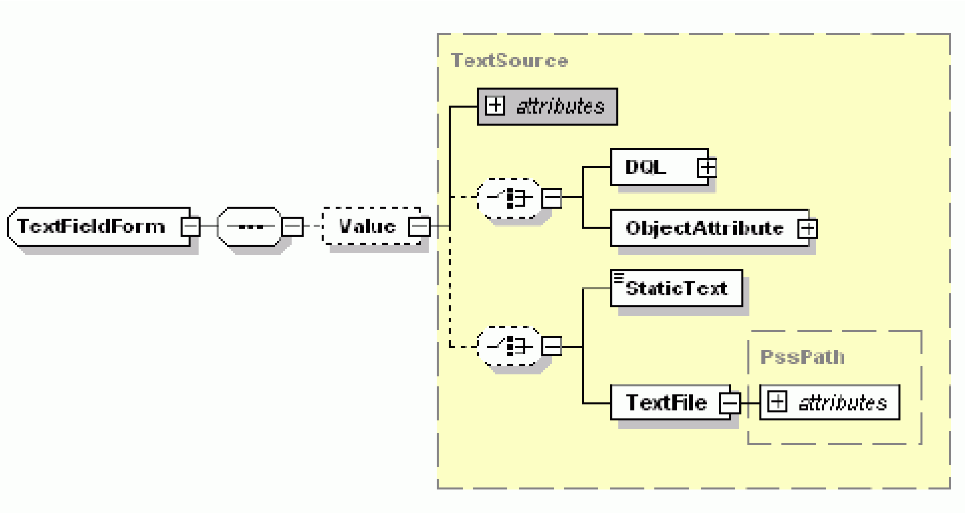
Sub-nodes

N/A

TextFieldForm

TextFieldForm defines the parameters for all text forms in an overlay PDF template that needs to be filled at runtime.

Figure 62. TextFieldForm



Attribute

N/A

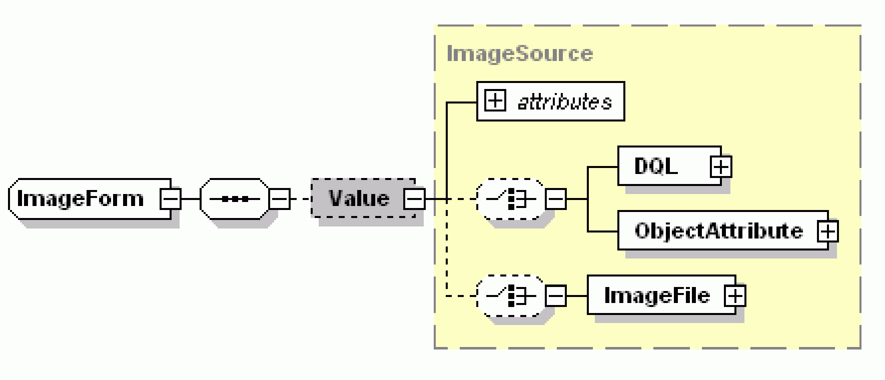
Sub-nodes

- **Value**: a TextSource node.

ImageForm

ImageForm defines the parameters for all image forms in an overlay PDF template that need to be filled at run-time.

Figure 63. ImageForm



Attribute

N/A

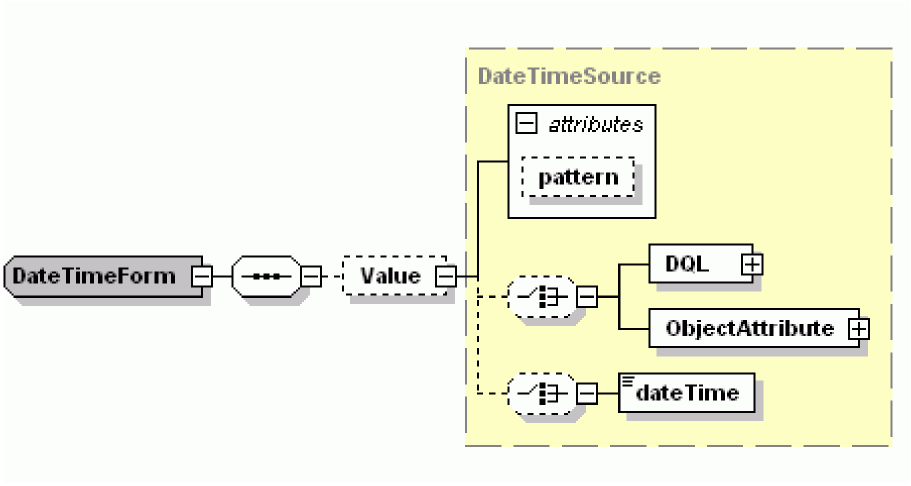
Sub-nodes

- **Value**: an ImageSource node.

DateTimeForm

DateTimeForm defines the parameters for all Date/Time forms in an overlay PDF template that needs to be filled at runtime.

Figure 64. DateTimeForm



Attributes

N/A

Sub-nodes

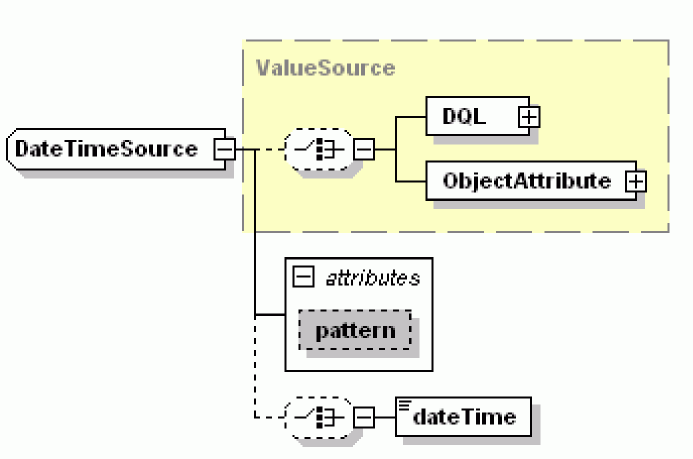
- **Value**: an DateTimeSource node.

Note: Important: **iText** does not support the "Data/Time" form in a PDF template. Please use the "Text Field" form instead of the "Date/Time" form if you are using the iText implementation of PSS.

DateTimeSource

DateTimeSource defines the options for Date/Time data source. It is a sub-type of ValueSource. It has 3 kinds of data source: DQL and ObjectAttribute, which inherited from ValueSource, and StaticDateTime.

Figure 65. DateTimeSource



Attribute

- **pattern**: the pattern of Data/Time displayed. PSS supports all pattern definitions described in "java.text.SimpleDateFormat".

Sub-nodes

- **DQL**: inherited from ValueSource.
- **ObjectAttribute**: inherited from ValueSource.
- **dateTime**: An ISO8601 Date/Time in string.

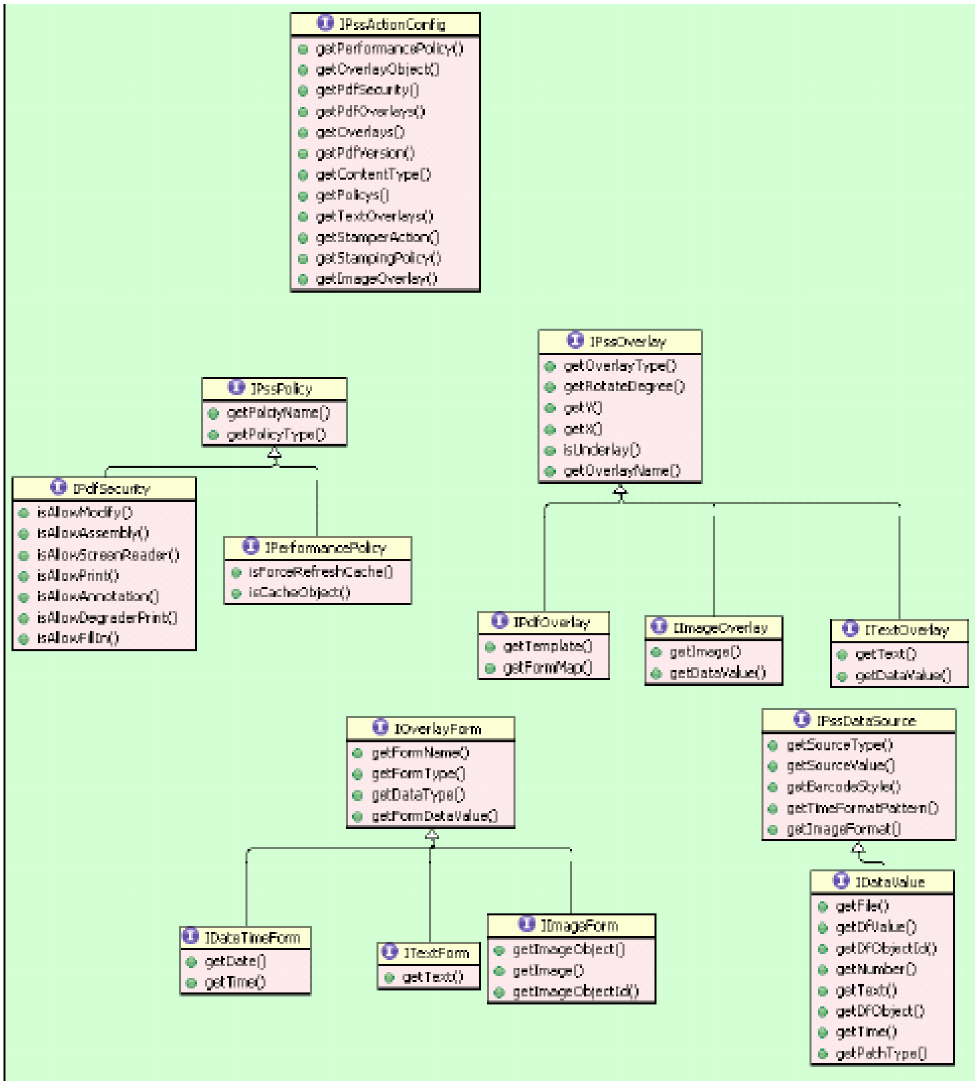
PSS XML configuration API design

Java API object model

Overview

The PSS API object model has four layers: stamper action, policy/overlay, PDFForm and Data Source/Value.

Figure 66. PSS API Object Model

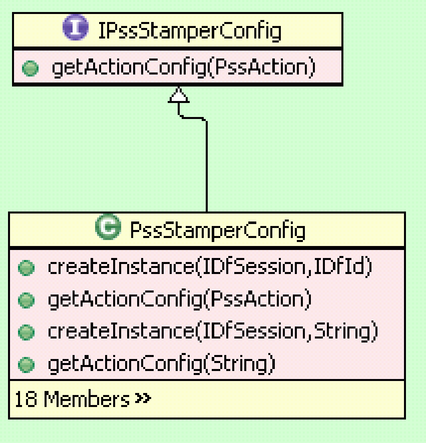


Stamper Action

Interface IPssStamperConfig and its implementation PssStamperCofig are the entry of PSS configuration API.

Relationship

Figure 67. Related

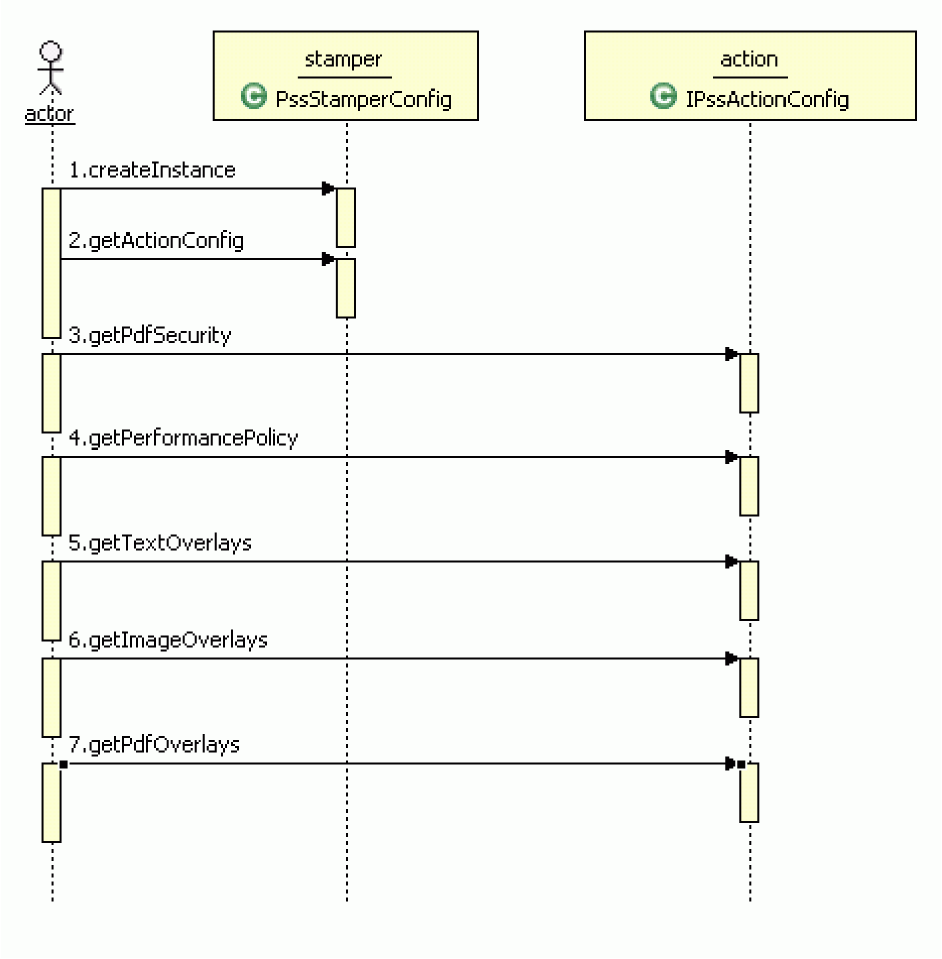

IPssStamperConfig
getActionConfig(PssAction)
PssStamperConfig
createInstance(IDfSession,IDfId)
getActionConfig(PssAction)
createInstance(IDfSession,String)
getActionConfig(String)
18 Members »

Interaction - creation

PSS runtime gets an instance of IPssStamperCnfig/PssStamperConfig through its static creation method: createInstance(). Then PSS runtime can get Stamper's Action configuration: IPssActionConfig, which is ConfigClass in PSS xml file, through getActionConfig() method.

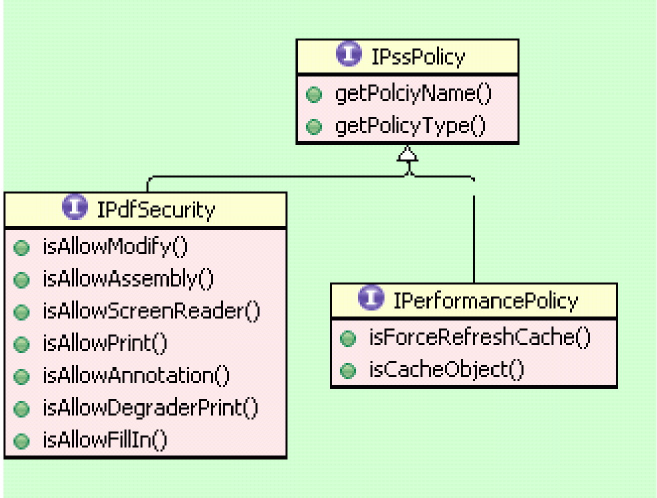
The interface IPssActionConfig has all the information of Policy and Overlay. The sequence of actions are detailed in the diagram below:

Figure 68. Interface Action Configuration Sequence



Policy

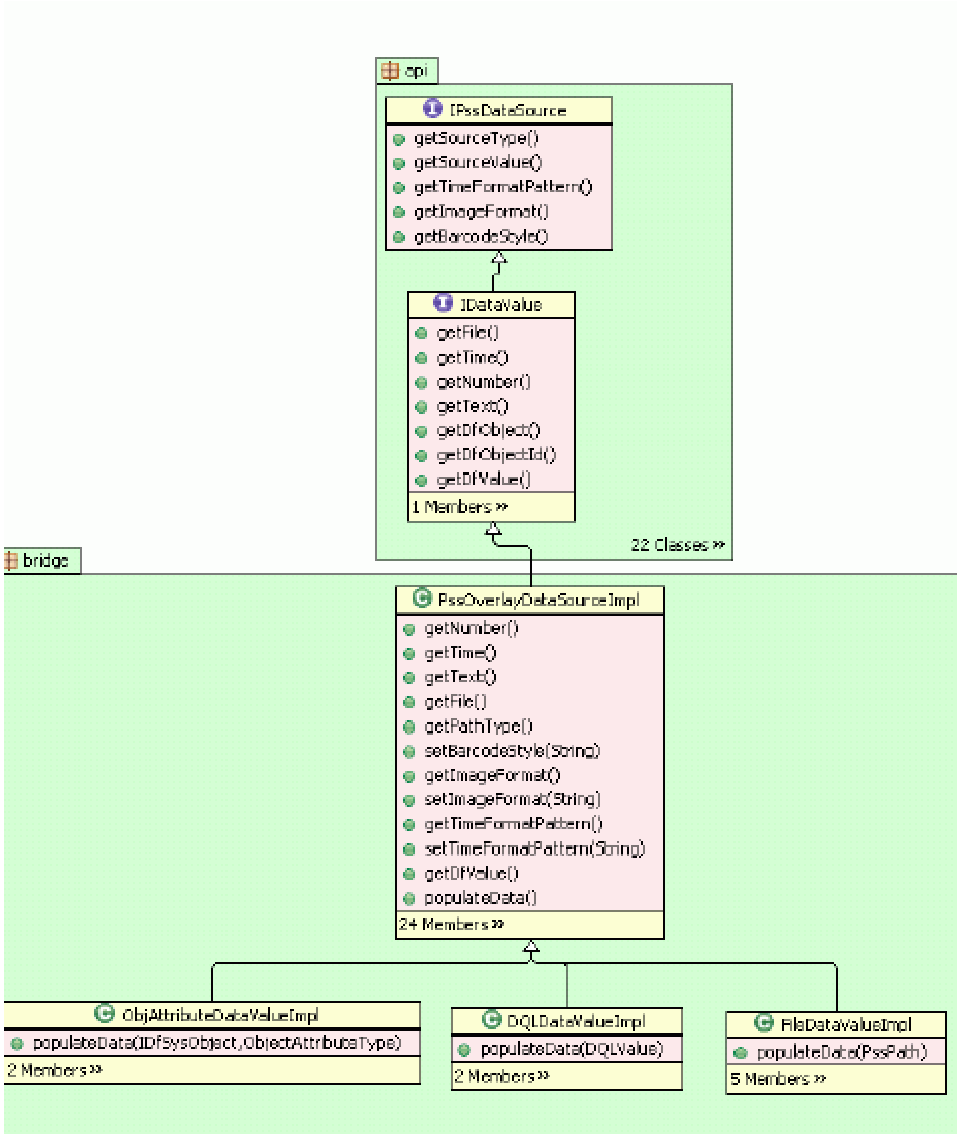
Figure 69. IPssPolicy



Data Source/Value and data population

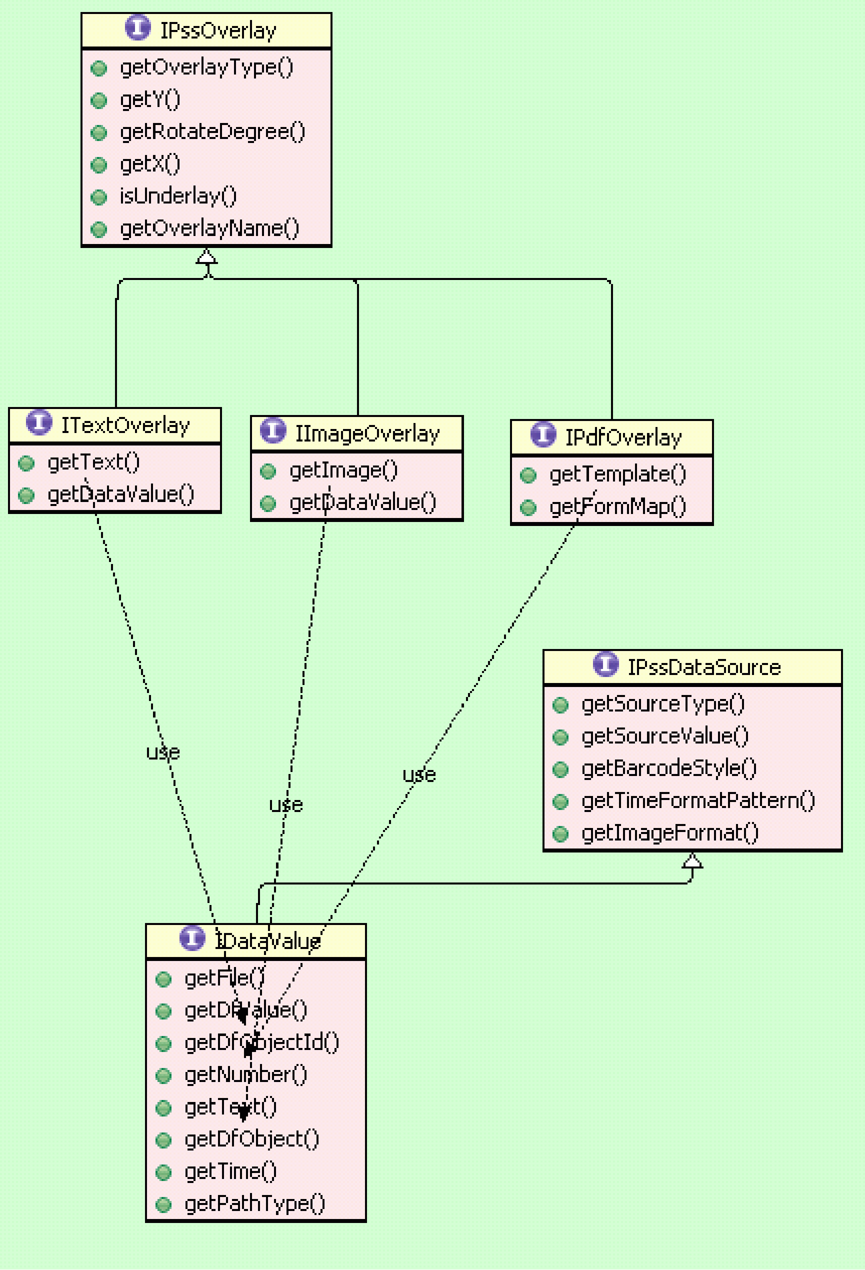
Both "Overlay" and "PDF" forms have data source definitions. There are three dynamic data source types: "DQL", "Docbase Object Attribute" and "Docbase Object/File". PSS runtime concerns about data value instead of data source. So the PSS configuration API introduces a DataValue mechanism to populate the overlay parameter from the data source. DataValue and DataSource are implemented as an inheritance relationship for data population. The methods **populateXXX()** in PssOverlayDataSourceImpl, FileDataValueImpl, DQLDataValueImpl and ObjAttributeDataValueImpl implement the data population logic. The details show below.

Figure 70. IPssDataSource



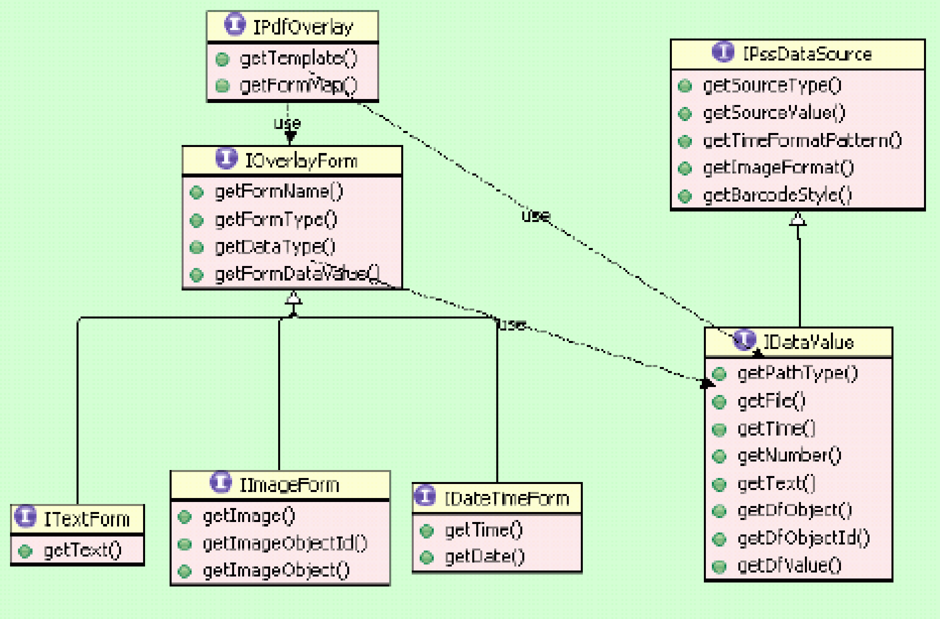
Overlay

Figure 71. IPssOverlay



PDF forms

Figure 72. IPdfOvrelay



JAXB customization and code generation

JAXB uses xml annotation (xs:annotation) to customize code generation. PSS has the following customizations.

Global binding customization

```
<xs:annotation>
 <xs:appinfo>
  <jxb:globalBindings choiceContentProperty="true" generateIsSetMethod=
       "true">
   <jxb:serializable uid="1"/>
   <jxb:javaType name="java.util.Calendar" xmlType="xs:dateTime"
         parseMethod="javax.xml.bind.DatatypeConverter.parseDate"
         printMethod="javax.xml.bind.DatatypeConverter.printDate"/>
  </jxb:globalBindings>
 </xs:appinfo>
</xs:annotation>
```

Attibutes

- **choiceContentProperty** : to handle choice content efficiently.
- **generateIsSetMethod** : to handle optional attribute/element efficiently.

Elements

- **Serializable**: all generated classes implement serializable interface. This make deep java object copy efficiently.
- **javaType**: this option control the cod generation that map xs:dataTime to java.util.Calendar.

typesafeEnumClass customizations

The purpose of **typesafeEnumClass** customization is to facilitate "WDK DropdownList UI Control" programming. The Enum value is the value of xml properties and the Enum name is the key of the localization string.

Note: In DCM 6.5, we only support font and encoding provided by iText BASE FONT. Some special characters under a certain font and encoding combination may not be displayed when changed to another font/encoding combination.

- **EnumHierarcyType**

```
<xs:element name="PssConfig">
 <xs:complexType>
  <xs:attribute name="hierarchyType" use="required">
   <xs:simpleType>
    <xs:annotation>
     <xs:appinfo>
      <jxb:typesafeEnumClass name="EnumHierarcyType">
       <jxb:typesafeEnumMember
        name="HIERARCY_DCTMOBJECTTYPE"
        value="DctmObjectType"/>
       <jxb:typesafeEnumMember
        name="HIERARCY_DCMDOCUMENTCLASS"
        value="DcmDocClass"/>
      </jxb:typesafeEnumClass>
     </xs:appinfo>
    </xs:annotation>
    <xs:restriction base="xs:string">
     <xs:enumeration value="DctmObjectType"/>
     <xs:enumeration value="DcmDocClass"/>
    </xs:restriction>
   </xs:simpleType>
  </xs:attribute>
 </xs:complexType>
</xs:element>
```

- **OddEvenType**

```
<xs:simpleType name="OddEventEnum" final="list">
 <xs:annotation>
  <xs:appinfo>
   <jxb:typesafeEnumClass name="OddEvenType">
    <jxb:typesafeEnumMember name="OVERLAY_ALL" value="All"/>
    <jxb:typesafeEnumMember name="OVERLAY_EVEN" value="Odd"/>
    <jxb:typesafeEnumMember name="OVERLAY_ODD" value="Even"/>
   </jxb:typesafeEnumClass>
  </xs:appinfo>
 </xs:annotation>
 <xs:restriction base="xs:string">
  <xs:enumeration value="All"/>
  <xs:enumeration value="Odd"/>
  <xs:enumeration value="Even"/>
 </xs:restriction>
</xs:simpleType>
```

- **EnumStamperActionType**

```
<xs:complexType name="PssStamperType">
 <xs:attribute name="action" use="required">
  <xs:simpleType>
   <xs:annotation>
    <xs:appinfo>
     <jxb:typesafeEnumClass name="EnumStamperActionType">
      <jxb:typesafeEnumMember name="ACTION_VIEW" value="View"/>
      <jxb:typesafeEnumMember name="ACTION_EXPORT" value="Export"/>
      <jxb:typesafeEnumMember name="ACTION_PRINT" value="Print"/>
     </jxb:typesafeEnumClass>
    </xs:appinfo>
   </xs:annotation>
   <xs:restriction base="xs:string">
    <xs:enumeration value="View"/>
    <xs:enumeration value="Export"/>
    <xs:enumeration value="Print"/>
   </xs:restriction>
  </xs:simpleType>
 </xs:attribute>
</xs:complexType>
```

- **EnumFormType**

```
<xs:complexType name="PssPdfFormType">
 <xs:attribute name="type">
  <xs:simpleType>
   <xs:annotation>
    <xs:appinfo>
     <jxb:typesafeEnumClass name="EnumFormType">
      <jxb:typesafeEnumMember name="MSG_TEXTFORM" value="com.emc.
documentum.pssdatabinding.stamper.TextFieldForm"/>
      <jxb:typesafeEnumMember name="MSG_BARCODEFORM" value="com.
emc.documentum.pssdatabinding.stamper.BarcodeForm"/>
      <jxb:typesafeEnumMember name="MSG_IMAGEFORM" value="com.
emc.documentum.pssdatabinding.stamper.ImageForm"/>
      <jxb:typesafeEnumMember name="MSG_DATETIMEFORM" value="com.
emc.documentum.pssdatabinding.stamper.DateTimeForm"/>
     </jxb:typesafeEnumClass>
    </xs:appinfo>
   </xs:annotation>
   <xs:restriction base="xs:string">
    <xs:enumeration value="com.emc.documentum.pssdatabinding.stamper.
TextFieldForm"/>
    <xs:enumeration value="com.emc.documentum.pssdatabinding.stamper.
BarcodeForm"/>
    <xs:enumeration value="com.emc.documentum.pssdatabinding.stamper.
ImageForm"/>
    <xs:enumeration value="com.emc.documentum.pssdatabinding.stamper.
DateTimeForm"/>
   </xs:restriction>
  </xs:simpleType>
 </xs:attribute>
</xs:complexType>
```

- **EnumPathType**

```
<xs:attributeGroup name="PssPathType">
 <xs:attribute name="type" use="required">
  <xs:simpleType>
   <xs:annotation>
    <xs:appinfo>
     <jxb:typesafeEnumClass name="EnumPathType">
      <jxb:typesafeEnumMember name="PATH_OS" value="OS"/>
      <jxb:typesafeEnumMember name="PATH_DM" value="Documentum"/>
      <jxb:typesafeEnumMember name="PATH_URI" value="URI"/>
     </jxb:typesafeEnumClass>
    </xs:appinfo>
   </xs:annotation>
   <xs:restriction base="xs:string">
    <xs:enumeration value="OS"/>
    <xs:enumeration value="Documentum"/>
    <xs:enumeration value="URI"/>
   </xs:restriction>
  </xs:simpleType>
 </xs:attribute>
</xs:attributeGroup>
```

Stamper actions and attributes for view and export

All stamper actions for each of the various overlays are configured from the **View** and the **Export** tabs displayed when you select and **Edit** a **Doc Type** from the **Document Types** screen.

Although the attributes for both View and Export are the same, their settings do not have to be the same. A user can export the PDF document to their local drive or elsewhere with the same PDF Security settings, PDF Properties, and overlays based on the settings of the Export tab. A user who decides/needs to export a copy of the PDF document to their local drive for example, will obtain a copy without the overlays, security settings, and property settings if no entries are provided for the Export tab. The PDF document in the repository when opened contains all the watermarks, security, and property settings as defined according to the View Tab. The copy in the repository containing the watermarks can be exported with or without the watermarks, security settings, and properties settings depending on the entries defined according to the Export tab. All attribute settings can be inherited or only specific settings can be inherited from the Super Type under the Document Types added/listed. Inheritance of an attribute setting searches the Super Types listed, if there any listed below the default_root, until the default_root is reached at the top of the hierarchy. There are no checkboxes for inheritance when the default-root is selected for editing. Any document type added/listed below the default_root however displayes the checkboxes for inheritance.

Administrators define the attribute settings on the Export tab such that a user who exports the PDF document obtains it according to the settings defined on the Export tab.

Attributes for the security settings are exposed when you click **PDF Security** on either tab for **View** and **Export**.

Attributes for the property settings are exposed when you click **PDF Properties** on either tab for **View** and **Export**.

Attributes for adding the various overlays are always exposed.

Overlay editing of attributes for text, image, and PDF

The screen displayed for text, image, and PDF overlays, when you click **Add** in the **Text Overlay List**, the **Image Overlay List**, and the **PDF Overlay List** under the **View** tab or the **Export** tab, contains the attribute settings used to configure watermarks.

The following list defines the attributes on the tabs displayed when you **Add** an overlay:

- This attribute is defined on the **Create** tab for all overlays added/created:
  - **Name**

    Each or any overlay added/created must have a unique name. This attribute is mandatory, on the **Create** tab, regardless of the overlay added.
- These attributes are defined on the **Attributes** tab for all overlays added/created:
  - **Underlay**

    When selected, the PDF document is applied on top of the overlay. If deselected, the overlay is applied on top of the PDF document.
  - **Odd/Even Overlay**

    The value selected for this attribute determines whether the overlay is applied to odd pages, even pages, or all pages of the PDF document.
  - **Rotate Degree**

    The value or number typed for this attribute determines the angle at which the watermark is applied to the PDF document.
  - **Position X(Pixels)**

    The value entered for this attribute in combination with the value provided for the "Y" position determines where the watermark is to be placed on the page.
  - **Position Y(Pixels)**

    The value entered for this attribute in combination with the value provided for the "X" position determines where the watermark is to be placed on the page.

- These attributes are defined on the **Text Overlay** tab for text overlays only:
  - **Font Name**

    The value selected for this attribute determines what font the watermark is to use, Courier or Helvetica for example.
  - **Encoding**

    The options/values that populate the list box for this attribute are determined based on the Font Name selected. The default value displayed is typically accepted.
  - **Font Size**

    The value can be typed and changed from the default value, 8.0, to any integer between 8.0 and 72.0.
  - **Font Color**

    The default value, black, can be changed to the desired color from the color pad.
  - **Text Source**

    The content that populates the text overlay is obtained and retrieved from one of the specified sources based on the selected radio button.

    Object Attribute

    The content is retrieved from one attribute of the specified document for the watermark.

    DQL

    The content for the text form is retrieved from the DQL query results based on the specified attribute name for the Attribute field. The value provided for Delimiter field separates the results returned in the DQL query if the specified Attribute is a repeating attribute.

    Docbase Object(file)

    The content for the text form is retrieved from a file stored in the repository.

    Static

    The watermark is hard coded based on the text entered.

- These attributes are defined on the **Image Overlay** tab for image overlays only:
  - **Image Source**

    The content that populates the image overlay is obtained and retrieved from one of the specified sources based on the selected radio button.

    Object Attribute

    The image is retrieved from one attribute of the specified document for the watermark.

    DQL

    The image is retrieved from the DQL query results based on the specified attribute name for the Attribute field. The value provided for Delimiter field separates the results returned in the DQL query if the specified Attribute is a repeating attribute.

    Docbase Object(file)

    The image is retrieved from a file stored in the repository.
- These attributes are defined on the **Overlay Template** tab for PDF overlays only:
  - **Template Source**

    The content that populates the PDF overlay is obtained and retrieved from one of the specified sources based on the selected radio button.

    Object Attribute

    The template is retrieved from one attribute of the specified document for the watermark.

    DQL

    The template is retrieved from the DQL query results based on the specified attribute name for the Attribute field. The value provided for Delimiter field separates the results returned in the DQL query if the specified Attribute is a repeating attribute.

    Docbase Object(file)

    The template is retrieved from a file stored in the repository.

- These attributes are defined on the **PDF Forms** tab for PDF overlays only, though the attributes described here exposed only when you **Add** a form using a **Text Form**, **Image Form**, or **Date Form** as the intended source for the watermark:
  - **Text Form** attributes:
    - **Name**

      The value is mandatory and must be unique among the Text Forms created if PDF Forms are utilized.
    - **Form Name**

      The value typed for the Name on the Create tab when the Text Form is selected.
    - **Form Type**

      This attribute is read-only to identify the form being added.
    - **Font Name**

      The value selected for this attribute determines what font the watermark is to use, Courier or Helvetica for example.
    - **Encoding**

      The options/values that populate the list box for this attribute are determined based on the Font Name selected. The default value displayed is typically accepted.
    - **Font Size**

      The value can be typed and changed from the default value, 8.0, to any integer between 8.0 and 72.0.
    - **Font Color**

      The default value, black, can be changed to the desired color from the color pad.
    - **Source Types**

      The content that populates the text form is obtained and retrieved from one of the specified sources based on the selected radio button.

      Object Attribute

      The content is retrieved from one attribute of the specified document for the watermark.

      DQL

      The content for the text form is retrieved from the DQL query results based on the specified attribute name for the Attribute field. The value provided for Delimiter field separates the results returned in the DQL query if the specified Attribute is a repeating attribute.

      Docbase Object(file)

      The content for the text form is retrieved from a file stored in the repository.

      Static

      The watermark is hard coded based on the text entered.
  - **Image Form** attributes:
    - **Form Name**

PSS system configuration

PSS uses the Java properties file called pss.properties to store the PSS system configuration. The list below shows some examples of possible configurations. A complete list and explanation of each configuration is provided within pss.properties.

- pss.pdf.manipulation.type = itext
- pss.esign.config.type = dmc_pss_esign_template
- pss.stamp.config.type = dmc_pss_stamp_config
- pss.esign.EventName = dm_addesignature
- pss.stamp.Config.HierarchyType = DctmObjectType
- pss.esign.ServiceType = Java Method Server method
- pss.extension.StampedPdf = STD
- pss.extension.MergedPdf = MGD
- pss.extension.ESignedPdf = ESD
- pss.extension.OriginalPdf = ORG
- pss.extension.FormfilledPdf = FLD
- pss.esign.support.format = pdf
- pss.LoggerName = pss
- pss.stamp.config.path = /System/Applications/pss
- pss.stamp.default.ownerpassword = ownerPassword
- pss.stamp.default.userpassword = userPassword
- pss.stamp.default.encryption.type = ENCRYPTION_RC4_40
- pss.stamp.default.time.format = yyyy-MM-dd HH:mm:ss
- pss.stamp.default.pdf.version = 1.7
- pss.stamp.default.font.type = Courier
- pss.stamp.default.encode = Cp1250
- pss.stamp.default.font.size = 8.0
- pss.esign.default.pdf.version = 1.4
- pss.esign.default.font.western = Courier
- pss.esign.default.encode.western = Cp1250
- pss.esign.default.font.chinese = STSong-Light
- pss.esign.default.encode.chinese = UniGB-UCS2-H
- pss.esign.default.font.japanese = HeiseiMin-W3
- pss.esign.default.encode.japanese = UniJIS-UCS2-H
- pss.esign.default.font.korean = HYGoThic-Medium

- pss.esign.default.encode.korean = UniKS-UCS2-H

Chapter 13

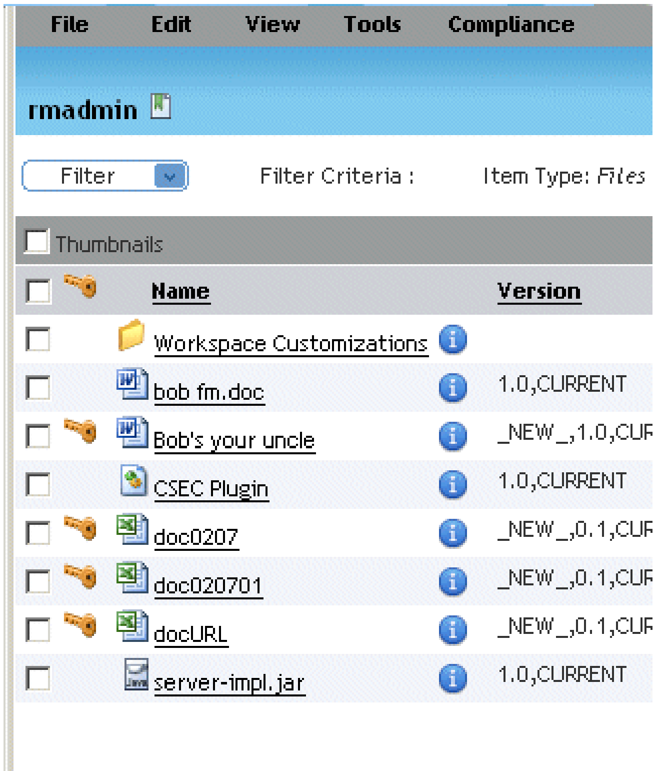
Enabling Controlled Document Filtering

The controlled document **Filter** feature is not visible on the user interface unless it is enabled. To enable the document list **Filter**, the DCM Application Server administrator must change the default setting from **false** to **true** in **showextendedfilter** attribute of the /dcm/**app.xml** file on the Application Server.

```
<!--define filtering controlled documents-->
 <filtersettings>
  <showallversions>true</showallversions>
  <showextendedfilter>true</showextendedfilter>
 </showextendedfilter>
```

Restart the DCM Application Server for the changes to take affect. The user can then login to DCM and see the feature enabled.

Figure 73. Controlled document Filter feature



Filter Criteria : Item Type: *Files*

□		Name		Version
□		Workspace Customizations	ⓘ	
□		bob fm.doc	ⓘ	1.0,CURRENT
□		Bob's your uncle	ⓘ	_NEW_,1.0,CUR
□		CSEC Plugin	ⓘ	1.0,CURRENT
□		doc0207	ⓘ	_NEW_,0.1,CUR
□		doc020701	ⓘ	_NEW_,0.1,CUR
□		docURL	ⓘ	_NEW_,0.1,CUR
□		server-impl.jar	ⓘ	1.0,CURRENT

User choices from the **Filter** options in the drop down list are as follows:

Note: Choices the user makes, from the drop down list, are displayed horizontally next to the **Filter Criteria** by: **Item Type**, **Lifecycle States**, and **Version**.

- **Item Type**
  - **Files and Folders**
  - **Files**
  - **Folders**
  - **All Object Types**
- **Version**
  - **All Versions**
  - **Current Version**
  - **Most Recent Version**

- **Lifecycle State**
  - **All States**
  - **In Progress**
  - **Review**
  - **In Approval**
  - **Approved**
  - **Customer**
  - **Effective**
  - **Retired**
  - **Obsolete**

Appendix A

The DCM Data Model and Object Types

This appendix describes the data model for DCM. All DCM-specific repository object types are described, as well as any extensions to existing Documentum object types.

The DCM Data Model

The following figures show the data model for DCM. The object types and attributes shown in italics are additions to base Documentum object types and attributes. All additional object types and attributes are defined in the following sections.

Figure 74. The DCM Data Model — dm_sysobject extensions

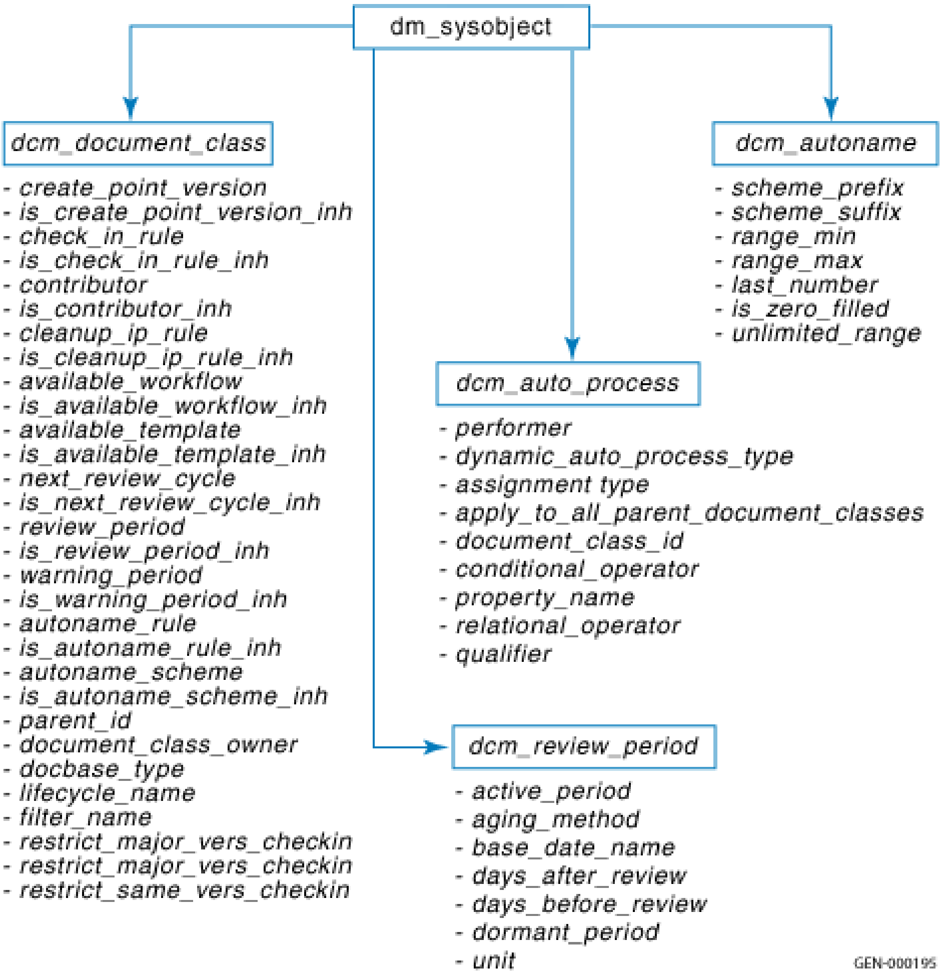


Figure 75. The DCM Data Model — dm_relation extensions

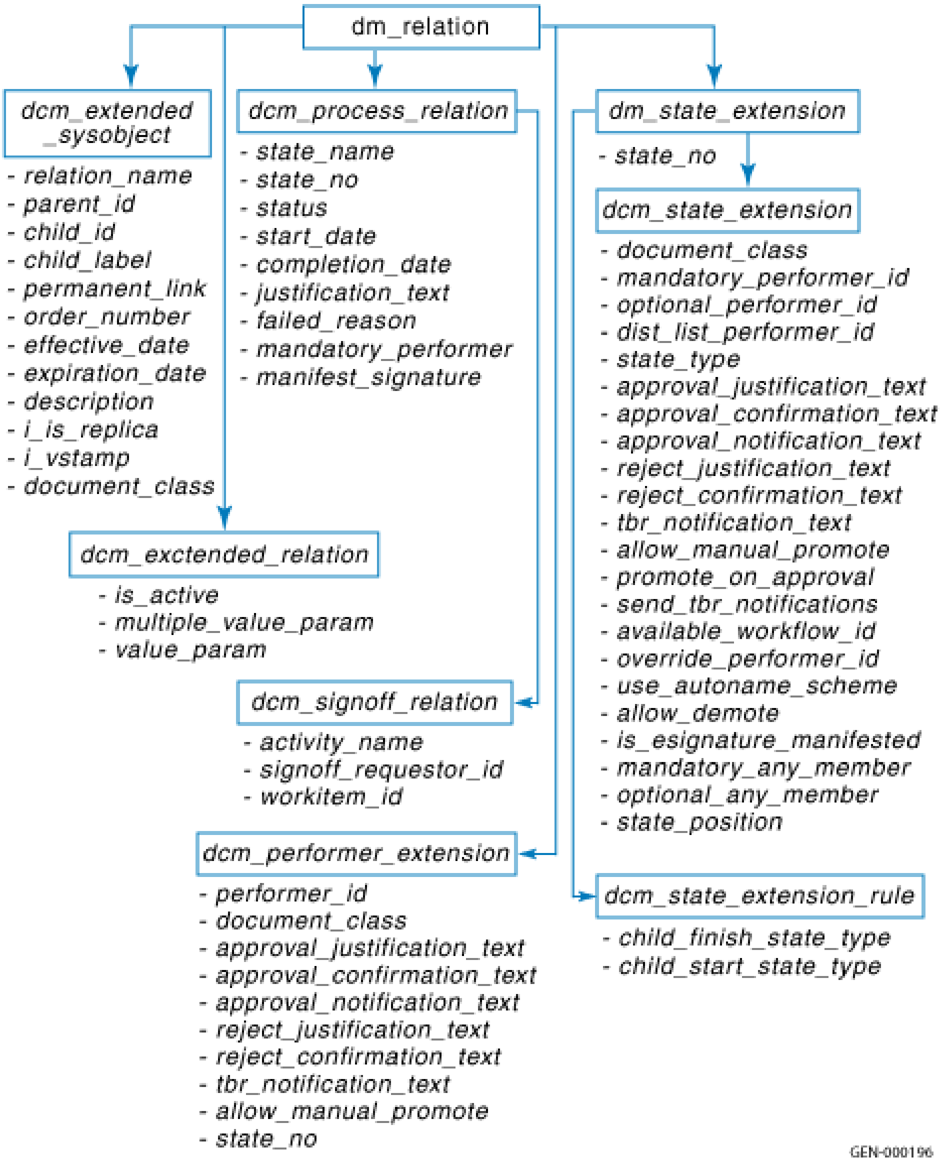


Figure 76. The DCM Data Model — dm_relation_type extensions

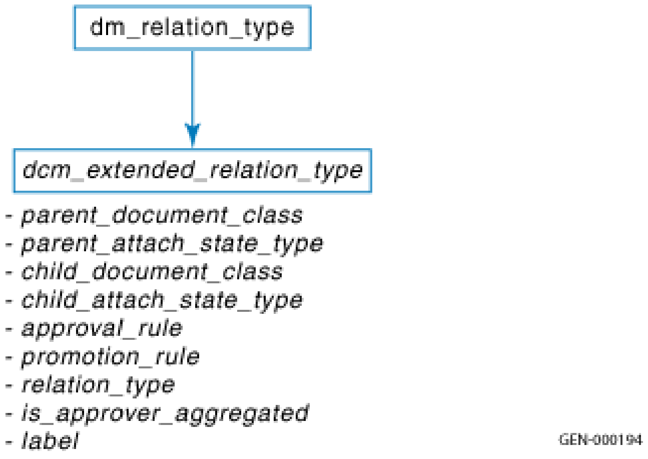
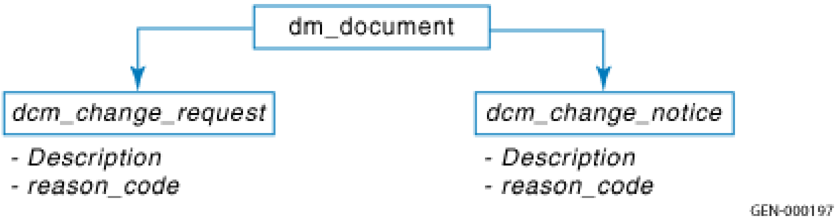


Figure 77. The DCM Data Model — dm_document extensions



DCM object types and properties

This section describes all of the DCM-specific object types, their purposes, and their properties. This section discusses only the DCM-specific extensions to the existing Documentum object model.

For information about the base Documentum object model and the object types recognized by Content Server, see the *Content Server Object Reference Manual*.

dm_sysobject extensions

This section describes the DCM extensions to the dm_sysobject type.

dcm_document_class

The dcm_document_class object type extends dm_sysobject to represent the configuration data for a DCM business application and its associated document classes.

Supertype: dm_sysobject

The following table lists the attributes defined for the document class type.

Table 50. Attributes defined for the document class type

Attribute	Datatype	Single/ Repeating	Description
alias_set_name			
autoname_scheme	ID	S	The ID of the autoname scheme for a business application or document class Optional field
autoname_rule	char(32)	S	This attribute controls whether the autoname scheme, if specified, is applied on create of an object, or on a lifecycle state. If placed on a lifecycle state, the validation process must enforce that the autoname process appears on only one state. Valid values: • **on_object_create_ only** • **on_lifecycle_state_ transition** This field is mandatory if there is an entry for autoname_scheme

Attribute	Datatype	Single/ Repeating	Description
available_template	ID	R	Contains the list of available templates for a business application or document class. If there is only one value, this value becomes the default template. Mandatory field; must have at least one entry
available_workflow	ID	R	The list of available workflows per business application or document class. If there is only one value, this is taken to be the default workflow. Mandatory field; must have at least one entry
cd_alias_set_name			
cleanup_ip_rule	char(32)	S	This rule determines the behavior when a new in progress version of a document is created. The rules are: • **prune_on_checkin** : When a new version of a document is created, delete all existing in-progress versions on checkin • **prune_on_ promote**: When an existing version of a document

Attribute	Datatype	Single/ Repeating	Description
			is promoted, delete all existing in-progress versions • **prune_on_ effective**: Delete on promote to effective all previous in-progress versions including all states prior to Effective. Pruning is restricted to a single branch in the version tree. • **prune_on_ effective_ including_ annotations** Same as prune_on_ effective, but also deletes annotations on all pre-Effective documents - including annotations on the effective version. • **do_nothing**: Do not delete in-progress versions Default value: do_nothing. Mandatory field

Attribute	Datatype	Single/ Repeating	Description
contributor	char(32)	R	A list of valid users, groups, or roles allowed to work on the business application or document class. Mandatory field; must have at least one entry
coordinator	char(32)	R	
create_point_version	Boolean	S	Defines whether DCM starts object version numbering with 0.1 instead of 1.0. Default value is 1.0. Mandatory field
docbase_type	char(32)	S	This is the repository object type that the document class maps to. Mandatory field. Only a single value permitted.
enforce_major_ effective			
enforce_unique_in_ba			
filter_name	char(32)	S	This attribute stores whether a document class is used for a controlled document, change notice, or change request type. Valid values: • controlled document types • change request types • change notice types

Attribute	Datatype	Single/ Repeating	Description
			Mandatory field. Only a single value permitted.
is_autoname_rule_inh	Boolean	S	Indicates whether the value of the autoname_rule attribute is inherited from parent. Default value: True
is_autoname_scheme_inh	Boolean	S	Indicates whether the value of the autoname_scheme attribute is inherited from parent. Default value: True
is_available_template_inh	Boolean	S	Indicates whether the value of the available_template attribute is inherited from parent. Default value: True
is_available_workflow_inh	Boolean	S	Indicates whether the value of the available_workflow attribute is inherited from parent. Default value: True

Attribute	Datatype	Single/ Repeating	Description
is_check_in_rule_inh	Boolean	S	Indicates whether the value of the following set of attributes is inherited from the parent: • restrict_major_ vers_checkin • restrict_minor_ vers_checkin • restrict_same_ vers_checkin Default value: True
is_cleanup_ip_rule_ inh	Boolean	S	Indicates whether the value of the cleanup_ip_rule attribute is inherited from parent. Default value: True
is_contributor_inh	Boolean	S	Indicates whether value of the contributor attribute is inherited from parent. Default value is: True
is_coordinator_inh	Boolean	S	Indicates whether value of the coordinator attribute is inherited from parent. Default value is: True
is_create_point_ version_inh	Boolean	S	Indicates whether the value of the create_point_ version attribute is inherited from parent Default value is: True

Attribute	Datatype	Single/ Repeating	Description
is_document_review_ inh			
lifecycle_name	char(32)	S	The lifecycle name associated with a document class. Mandatory field. Only a single value permitted.
next_review_cycle	integer	S	Specifies a time period after the effective date. This value is used to determine the next review cycle for a document. Optional field
parent_id	ID	S	This attribute stores the ID of the parent (business application) object that the document class belongs to. This value tracks the parent relationship and it is used to easily obtain the configuration for the parent business application. The value of this field is set by the application.
restrict_major_vers_ checkin	Boolean	S	Do not allow major version checkin. Mandatory field. Valid values are TRUE or FALSE

Attribute	Datatype	Single/ Repeating	Description
restrict_same_vers_checkin	Boolean	S	Do not allow same version checkin. Mandatory field. Valid values are TRUE or FALSE
review_period	integer	S	The Document Review Notification job uses the value of this attribute to determine the number of days required to start the notification. Optional field
warning_period	integer	S	The Document Review Notification job uses the value of this attribute to determine the number of days required to send a notification if the business owner has not acted on the notification. Optional field

dcm_auto_process

The dcm_auto_process object type extends dm_sysobject to represent user list rules (also known as dynamic signatory processes). These objects contain queries, and can be assigned to a particular business application or document class. A user list rule performs one of the following functions:

- adds mandatory or optional signatories to a signoff on a state change
- adds mandatory or optional signatories to a workflow
- adds users to a notification list
- adds users to a to-be-read list

The conditional_operator, property_name, relational_operator, and qualifier attributes are correlated repeating attributes that are used to both store and reconstruct the dynamic query.

The description dm_relation attribute is used to store the reason for the user list rule. For example: "Over $1M contract."

Supertype: dm_sysobject

The following table lists the attributes defined for the dynamic signatory process type.

Table 51. Attributes defined for the dynamic signatory process type

Attribute	Datatype	Single/ Repeating	Description
any_member			
apply_to_parent_doc_class	Boolean	S	This setting is used to specify if the user list rule should apply to all parent document class objects. Default value of this attribute is False.
assignment_type	char(32)	S	This is a subcategory of the dynamic_auto_process_type. It applies specifically to the optional value in the dynamic_auto_process_type attribute. The options are: • **workflow_assignment**: Present optional users in a workflow for parallel review task • **contributor_assignment**: Signoff process will check this setting to determine if these performers will be added to existing performers • **workflow_contributor_assignment**: Represents both checkboxes checked

Attribute	Datatype	Single/ Repeating	Description
			• **no_assignment**: No value assigned This field is mandatory.
conditional_ operator	char(32)	R	This is the conditional operator between the property name and the qualifier. Valid values are: • AND • OR This field is optional.
document_class_id	ID	R	The value of this attributes determines to which document classes the user list rule applies. The user can also select **All** to indicate that this rule applies to all document classes in a particular business application. If the user selects **All** for both the business application and the document class, the DCM system applies the user list rule to all document classes. The user can multi-select several DCs. This field is mandatory if the apply_to_parent_doc_ classes attribute is set to False.

Attribute	Datatype	Single/ Repeating	Description
dynamic_auto_ process_type	char(32)	S	This setting will be used to determine the type of use list rule. The options are: • mandatory • optional • to_be_read_required • notification This field is mandatory.
performer_id	char(32)	R	This attribute specifies who is in the user list rule. DCM can be configured based on group, role, or user. The users contained in the listed groups and roles will constitute the list of users that are added to the user list rule. This field is mandatory.
property_name	char(32)	R	This attribute contains a list of the properties that the user may choose. This will be displayed as a drop-down list of attribute names for the object type selected. This attribute correlates with condition and property values attributes. This field is optional.
qualifier	char(32)	R	Contains a qualifier value that is used to construct that part of the query that follows the relational_operator. Optional field

Attribute	Datatype	Single/ Repeating	Description
relational_operator	char(32)	R	One of the following query operators: • greater than • equals • less than • any • contains Optional field
state_type			

dcm_autoname

The dcm_autoname object type extends dm_sysobject to represent the autonaming schemes for DCM controlled document types. These autoname schemes can then be assigned to business applications or document classes. A single autoname scheme can be assigned to multiple business applications or document classes, but the numbering sequence applies across all those business applications or document classes.

Supertype: dm_sysobject

The following table lists the attributes defined for the autoname type.

Table 52. Attributes defined for the autoname type

Attribute	Datatype	Single/ Repeating	Description
last_value	double	S	The last generated auto name number for this autoname scheme. Mandatory: set by application

Attribute	Datatype	Single/ Repeating	Description
is_zero_filled	Boolean	S	Specifies whether the generated number portion of the autoname is filled with leading zeros. A value of **True** will add leading zeros. A value of **False** will return just the number. Cannot have zero filled if no range_max is specified. Optional
range_max	double	S	The highest, or ending number for this autoname scheme. Optional
range_min	double	S	The lowest, or starting number for this autoname scheme. Optional
scheme_prefix	char(32)	S	Any alphanumeric value that is added to the beginning of all the generated autoname schemes of this type. Optional

Attribute	Datatype	Single/ Repeating	Description
scheme_suffix	char(32)	S	Any alphanumeric value that is appended to all generated autoname schemes of this type. Optional
unlimited_range	Boolean	S	Specifies whether an autoname scheme can use an unlimited range of numbers. Mandatory: must be set to either True or False.

dcm_review_period

Though reserved for future use, this object type and its attributes are included on the DAR file used for installing DCM. You are able to see it, using Documentum Application Builder (DAB), but not use it.

Supertype: dm_sysobject

The following table lists the attributes defined for the review period type.

Table 53. Attributes defined for the review period type

Attribute	Datatype	Single/ Repeating
active_period	char(32)	S
aging_method	char(32)	S
base_date_name	char(32)	S
days_after_review	char(32)	S
days_before_review	char(32)	S
dormant_period	char(32)	S
Unit	char(6)	S

dm_document extensions

This section describes the DCM extensions to the dm_document type.

dcm_change_notice

The dcm_change_notice object type extends dm_document to allow the creation of change notice documents, which have special processing requirements.

Supertype: dm_document

The following table lists the attributes defined for the change notice type.

Table 54. Attributes defined for the change notice type

Attribute	Datatype	Single/ Repeating	Description
reason_code	char(255)	S	A reason for the change notice; this value is selected from a pre-defined list. Mandatory field.
description	char(255)	S	A description entry used to describe the change request. Optional field.

dcm_change_request

The dcm_change_request object type extends dm_document to allow the creation of change request documents, which have special processing requirements.

Supertype: dm_document

The following table lists the attributes defined for the change request type.

Table 55. Attributes defined for the change request type

Attribute	Datatype	Single/ Repeating	Description
reason_code	char(255)	S	A reason for the change request; this value is selected from a pre-defined list. Mandatory field.
description	char(255)	S	A description entry used to describe the change request. Optional field.

dm_relation extensions

This section describes the DCM extensions to the dm_relation type.

dcm_extended_relation

The dcm_extended_relation represents the relation between a Controlled Document and its attachment, an attachment to a change request or change notice for example.

dcm_extended_sysobject

The dcm_extended_sysobject represents the relation between a Controlled Document and its Document Class.

The following table lists the attributes defined for the extended sysobject type.

Table 56. Attributes defined for the extended relation type

Attribute	Datatype	Single/ Repeating	Description
child_id	ID	S	Identifies the object that is the child in the relationship.

Attribute	Datatype	Single/ Repeating	Description
child_label	char(32)	S	Version label of the specified child_id. This is optional. If set, the child_id must be the chronicle ID for the child.
effective_date	date	S	Not system defined. Provided for the user's convenience.
expiration_date	date	S	Not system defined. Provided for the user's convenience.
description	char(255)	S	Not system defined. Provided for the user's convenience.
document_class	char(64)	S	Name of the document class associated with the dcm_extended_sysobject object.
i_is_replica	integer	S	Indicates whether the object is a local replica of an object in a remote repository.
i_vstamp	integer	S	Contains a count of the number of committed transactions that have changed this object. This value is used internally to support locking and versioning.
order_number	integer	S	Not system defined. This is provided for the user's convenience. For example, this could be used to order a set of relationships.
parent_id	ID	S	Identifies the object which is the parent in the relationship.

Attribute	Datatype	Single/ Repeating	Description
permanent_link	Boolean	S	Indicates if you want to maintain the relationship across versions of the parent object.
relation_name	char(32)	S	Identifies a valid relation type object, which defines the type of relationship existing between the two objects.

dcm_performer_extension

The dcm_performer_extension object type extends dm_relation to provide configuration override information tied to a specific performer. In DCM, a performer can be a group, role, or user ID. This type allows configuration choices to be based on a particular performer, such as defining a lifecycle state and performer combination.

Supertype: dm_relation

The following table lists the attributes defined for the DCM performer extension type.

Table 57. Attributes defined for the DCM performer extension type

Attribute	Datatype	Single/ Repeating	Description
allow_manual_ promote	Boolean	S	Allow the promote operation to be done manually through the user interface. Default value is False.
document_class	char(64)	S	Document class name. Mandatory: populated by the DCM application

Attribute	Datatype	Single/ Repeating	Description
performer_id	ID	S	The r_object_id of the role, group, or user object. This value is used as part of the key to look up performer configuration information. Mandatory: populated by the DCM application
reject_confirmation_ text	char(255)	S	The text for the rejection confirmation. Mandatory
reject_justification_ text	char(255)	R	The rejection text list that a user selects from when performing a reject operation. Mandatory: must have at least one value
signoff_confirmation_ text	char(255)	S	The text for the signoff confirmation. Optional
signoff_justification_ text	char(255)	R	The justification (also known as a *reason code text*) that a user must use when performing a signoff or promote process. Optional
signoff_notification_ text	char(255)	S	The text used in the signoff notification process. Optional

Attribute	Datatype	Single/ Repeating	Description
state_no	integer	S	The state number for the dcm_state_ extension_performer dm_relation_type.
tbr_notification_text	char(255)	S	Text used in the TBR notification process. Mandatory

dcm_process_relation

The dcm_process_relation object type extends dm_relation to define document-specific processes or requirements that must be satisfied prior to promotion or another application event such as:

- Required signoffs and signatures
- Required TBR confirmations
- Required change request signoff
- Close a change request when document is made effective
- Required change notice signoff
- Auto-review notifications

A dcm_process_relation type tracks the signoff information for each user. The dcm_process_relation objects are used by the signoff processes to determine if a user has acted on a particular document, and to track various information on these processes by user.

Supertype: dm_relation

The following table lists the attributes defined for the process relation type.

Table 58. Attributes defined for the process relation type

Attribute	Datatype	Single/ Repeating	Description
completion_date	date	S	Contains a Date/Time stamp when this process was completed. Mandatory: application entry

Attribute	Datatype	Single/ Repeating	Description
failed_reason	char(255)	S	Contains the error message recorded when an operation fails. Server methods populate this field for failure conditions since this information cannot be returned to the calling client program. This field is populated on failed status from server methods.
justification_text	char(255)	S	Contains the justification, or reason, for the process. Optional field
mandatory_ performer	Boolean	S	Indicates if performer for this process is mandatory Mandatory field: value is set to True for mandatory users
signature_manifest	Boolean	S	The value of this field indicates whether the signature should be manifested on the document. Default value: False
start_date	date	S	Contains a Date/Time stamp when this process was created. Mandatory: application entry
state_name	char32)	S	The lifecycle state name. Mandatory field: application entry

Attribute	Datatype	Single/ Repeating	Description
state_no	integer	S	The lifecycle state number. This value is part of the key that uniquely identifies the process relation for a particular state. The other part of the key is the dm_relation parent_id. Mandatory: application entry
status	char(32)	S	Status of the operation (for example, Pending, Approved, Promoted, Rejected, or Failed). These values will be defined in properties files for different processes (such as signoff, promote, to-be-read, etc.) so that they can be localized in the user interface. Mandatory: application entry

dcm_state_extension

The dcm_state_extension relation object type extends dm_relation to store attributes that define the mandatory signatories, optional signatories, and to-be-read groups, roles, and users values per state number for a particular document class, lifecycle, and lifecycle state. For each state in a document class/lifecycle combination, there is a distinct dcm_state_extension object instance.

Supertype: dm_relation

The following table lists the attributes defined for the state extension type.

Table 59. Attributes defined for the state extension type

Attribute	Datatype	Single/ Repeating	Description
allow_manual_ promote	Boolean	S	Specifies whether to allow the promote to done manually via user interface. Default value is False.
available_ workflow_id	ID	R	Specifies the workflow that will be used at a particular state. This will appear in a UI when a user starts a workflow for a given object in a particular state. If multiple workflows are specified, the user must choose which workflow to use. Optional
dist_list_ performer_id	ID	R	The ID of each dm_group or dm_user object included in the to-be-read list. This value is used when processing the to-be-read notices. Optional.
document_class	char(64)	S	Name of the document class associated with this dcm_state_extension object. This value in conjunction with the lifecycle and state_no attributes make the unique key to retrieve a dcm_state_extension object. Mandatory: populated by the DCM application

Attribute	Datatype	Single/ Repeating	Description
mandatory_ performer_id	ID	R	The ID of each dm_group or dm_user object included in the mandatory performer list. This value is used with the signoff and promote processes. Optional
optional_ performer_id	ID	R	The ID of each dm_group or dm_user object included in the optional performer list. This value is used with the signoff and promote processes. Optional.
override_ performer_id	ID	R	The r_object_id of the performer (dm_group or dm_user) object. This value, in conjunction with the dm_policy r_object_id value, make up the unique key to store and retrieve a dcm_ state_extension_performer relation object. This object stores the actual configuration for each performer. Optional.
promote_on_ signoff	Boolean	S	If set to True, specifies whether to promote the document on signoff. Default value is False.
reject_ confirmation_text	char(255)	S	The text for the rejection confirmation. Mandatory

Attribute	Datatype	Single/ Repeating	Description
reject_justification_text	char(255)	R	The rejection text that a user must use when performing a reject operation. Mandatory
send_tbr_notification	Boolean	S	Specifies whether to send both notification and To Be Read notifications. Default value is False.
signoff_confirmation_text	char(255)	S	The text for the signoff confirmation. Mandatory
signoff_justification_text	char(255)	R	The justification (also know as a *reason code text*), that a user must use when performing a signoff or promote process. Mandatory
signoff_notification_text	char(255)	S	The text for the signoff notification Mandatory
state_type	char(32)	S	This value specifies the current state_type of the document instance. This value is set by the DCM application
tbr_notification_text	char(255)	S	Text used in the TNR notification process. Mandatory
use_autoname_scheme	Boolean	S	Specifies if an autoname scheme is to be used on a lifecycle state. Mandatory. Default value is False.

dcm_state_process_rule

The dcm_state_process_rule relation object type extends dm_relation to

Supertype: dm_relation

The following table lists the attributes defined for the state process rule type.

Table 60. Attributes defined for the state process rule

Attribute	Datatype	Single/ Repeating	Description
child_finish_state_ type	char(32)	R	Child finish state type for group promotion. Optional
child_start_state_ type	char(32)	R	Child start state type for group promotion. Optional

dm_relation_type extensions

This section describes the DCM extensions to the dm_relation_type object type.

dcm_extended_relation_type

The dcm_extended_relation_type object extends the dm_relation_type object with additional attributes for defining the lifecycle attachment requirements and for defining any relationship processing rules. This allows business application owners to define custom relationships between document classes, and to define the relational behavior for application processing into rules that can be set as desired in the Create Relationship Types process. The rules are represented in the form of attributes.

Supertype: dm_relation_type

The following table lists the attributes defined for the extended relation type.

Table 61. Attributes defined for the extended relation type

Attribute	Datatype	Single/ Repeating	Description
child_attach_state_ type	char(32)	R	This attribute indicates the specific state types of the child documents that may be attached using this relation. Only documents in these states may be attached to the parent using this relation. The possible values are: • **Any**: Documents of all state types will appear in the attach list and may be attached to the parent document. • **state_type**: This value indicates the state type of child documents that will appear in the list of documents to be attached. Mandatory: Value of this attribute must be either **All** or one or more lifecycle state types.
child_document_ class	char(255)	R	Defines the child document class types. Mandatory: Value of this attribute must be either **All** or specific document classes

Attribute	Datatype	Single/ Repeating	Description
parent_attach_ state_type	char(32)	R	This attribute indicates the specific state types of the parent document this relation is allowed to be created with. Only parent documents in these states will be able to use this relation. The possible values are: • **Any**: Documents may be attached to the parent document at any state type. • **State Type**: This value indicates the state type value at which the parent document can have documents attached to it. Mandatory: Value of this attribute must be either **All** or one or more lifecycle state types.
parent_document_ class	char(255)	S	Defines the parent document class type. Mandatory: Specific document class
promotion_rule	char(32)	S	This attribute specifies the rule for the promotion_rule. Possible values are: • **parent_only**: Only the parent will be promoted • **parent_and_child**: The children will be promoted when the parent is promoted Mandatory

Attribute	Datatype	Single/ Repeating	Description
relation_type	char(32)	S	This attribute stores the value of the relation type. Current possible values are: • closes • process • reference • supporting Mandatory
signoff_rule	char(32)	S	This attribute specifies the rule for the signoff_rule. Possible values are: • **parent_only**: Only the parent document is approved • **child_only**: Only the children are approved • **children_with_parent**: The children are approved automatically when the parent is approved • **children_first**: The children must be approved before the parent can be approved or forwarded Mandatory

Appendix B

Troubleshooting

This appendix provides information for troubleshooting a DCM configuration, including how to use the DCM system validation tool. The following topics are discussed:

- Validating Your DCM System , page 295
- Frequently-Encountered Issues, page 296

Validating Your DCM System

The Validate System tool helps you ensure that there are no system configuration conflicts. Run this tool each time you create a new DCM configuration or modify an existing configuration.

The following DCM users can run this tool:

- The administrator can view all DCM configurations.
- The business application owner can only access the business application system configurations that they own.
- The document class owner can only access the document class configurations that they own.

For regulated companies that have to comply with the FDA regulation 21CFR, Part 11, the report generated by the Validate System tool can be processed for signoff before it is implemented. To assist the process of validating configuration changes, you can also view the changes between different version of the reports, thus identifying any parameters that have been modified.

To run the Validate System tool:

1. Connect to the DCM repository as an administrator, business application owner, or document class owner.
2. Select **Tools > Validate System**.
3. Select the parameters for your report:

Table 62. Validate System Fields

Field Name	Value
Report Type	Choose one of the following options from the pull-down menu: • **All** • **Warning** • **Error** • **Warnings and Errors**
Select the type of system validation:	Choose one of the following options from the pull-down menu: • **All**: Reports on all document class and business application objects in your system. • **Business Applications**: Reports only on the business application objects in your system. • **Document Classes**: Reports only on the document class objects in your system.

4. When you have finished selecting your report options, click **OK** to run the report.

   The report results are displayed, listing test subject, primary object, related object and status of the object:

Frequently-Encountered Issues

This section describes some frequently-encountered issues after installing and configuring DCM.

DCM Administration Node Not Displayed in Navigation Pane

Symptoms: After installing, the installation owner logs on to DCM. The DCM Administration node does not appear in the left pane.

Resolution: To see the DCM Administration module, you must add the user to the Administration role. (To create a business application or document class, they must also be added to the appropriate role.)

After making these changes, you must log out of DCM and restart the Application Server.

Newly-Created Document Classes Don't Appear in Drop-down List

Symptoms: When attempting to create a controlled document, a newly-created document class does not appear in the drop-down list.

Resolution: To display the newly-created document class in a controlled document drop-down list, you must add your user account to the list of Authors/Contributors configured for the document class. Newly-created document classes also require that you log out and log in to the DCM Webtop application.

A Newly-Created Controlled Document Vanishes After Check-in

Symptoms: After creating a controlled document and checking it into the DCM system, it simply vanishes.

Resolution: A controlled document when checked into the system for the first time will be migrated to the appropriate Controlled folder. This is dictated by the lifecycle that has been attached to the document. If you use the default and standard lifecycles provided with DCM, the document will end up in the **Controlled Docs→In Progress** folder.

A Newly-Created Controlled Document Does Not Appear in the In Progress folder.

Symptoms: After creating a controlled document and checking it into the DCM system, it does not appear in the **In Progress** folder.

Resolution: It is possible that you are using an Auto-name scheme to rename your document upon migration. To verify if this is the case click on the **View All Messages** in the status bar and DCM will list what the new document name has been changed to.

Optional Signatories for a Lifecycle Do Not Appear on a Controlled Document’s Properties Page

Symptoms: You have added a mandatory and an optional signatory in the Lifecycle and have created a controlled document. But when you view the Lifecycle’s **Properties** page, it only shows the mandatory signatories.

Resolution: When you add a signatory to a lifecycle state, you are adding either mandatory signatories or optional signatories. The optional signatories are really a list of signatories that you can select from when viewing the signatories on a controlled document. When you add a signatory from this list they become required signatories. Mandatory signatories set in the lifecycle are always required signatories.

Create Controlled Document Option is Unavailable

Symptoms: When you try to select **File→New→Controlled Document**, the option is grayed out or unavailable.

Resolution: To create a controlled document, the user must be:

- defined as an Author/Contributor in either a Business Application or a Document Class.
- in their Home Cabinet.

Cannot View Contents of Controlled Documents Folders

Symptoms: When you click on any of the Controlled Documents folders on the navigation bar, you receive an error message.

Resolution: If you have updated or upgraded your DCM software, delete the contents of the C:\Program Files\Apache Group\Tomcat 4.1\work\ folder. This folder contains the old compiled DCM application pages and can prevent the new pages from taking effect.

The Lifecycles Content Pane does not Display any Lifecycles

Symptoms: Under the DCM Administration module, you clicked on Lifecycles and the content pane displays no existing lifecycles.

Resolution: From the pull-down menu in the upper right-hand corner of the content pane, select **All Lifecycles** rather than the default display option, which is **Show Current User's Lifecycles**.

Manual Promotion Option on Menu is Grayed Out

Symptoms: When trying to promote a controlled document manually, the **Document→Lifecycle→Promote** menu item is disabled.

Resolution: Manual promotion must be defined in the lifecycle for each specific state that you want to promote to.

Cannot Create a Relationship Type

Symptoms: You receive an error when attempting to create a relationship type.

Resolution: You must have Superuser or Sysadmin privileges to create a relationship type.

Cannot Modify a Document Class

Symptoms: You modified a document class but the changes did not appear to take effect.

Resolution: There are two possible resolutions:

- To modify a document class, you must be the owner of that particular document class.
- Any changes made to a document class will not appear until the document class owner logs out of DCM.

Cannot Create a New Change Notice

Symptoms

: The **File→New→Change Notice** option is disabled.

Resolution: Verify the following things:

- Is there a Change Notice document class (either the default Change Notice document class installed with DCM, or a custom-configured document class)?

- Has the user been added on the **Author/Contributor** tab for the change notice document class?
- After the administrator created or modified the Change Notice document class, did the user log out and log in again to the DCM system? (Changes to document classes only go into effect for a new DCM session.)

Cannot Create a New Change Request

Symptoms: The **File→New→Change Request** option is disabled.

Resolution: Verify the following things:

- Is there a Change Request document class (either the default Change Request document class installed with DCM, or a custom-configured document class)?
- Has the user been added on the **Author/Contributor** tab for the change request document class?
- After the administrator created or modified the Change Request document class, did the user log out and log in again to the DCM system? (Changes to document classes only go into effect for a new DCM session.)

Cannot Create a Link Between a Change Notice and an Associated Change Request

Symptoms: The change request has been promoted to the Approved state, but the user is unable to set up the relationship linking the change request to a change notice.

Resolution: Verify that the State Type has been correctly defined in the Lifecycle Editor.

Electronic Signature Does Not Appear in Rendered PDF Documents

Symptoms: Electronic signatures do not appear in your rendered PDF documents

Resolution: Login to the database (where the DCM repository is created) and verify that the “a_esignature_required” field in dmi_registry_s table is set to 1 for both dcm_signoff and dcm_reject events.

If it is not already set to 1, you must set it to 1.

How To Display Attributes for Custom Document Types?

Symptoms: Cannot get attributes for custom document types to display on DCM screens.

Resolution: You must edit the attributes_dcm_controlled_doc_docbaseattributelist.xml file. For detailed instructions on selective display of custom attributes in a WDK-based application such as DCM, see *Chapter 4, Hiding Attributes from a Data Dictionary Display*, in the *Web Development Kit and Applications Tutorial*.

When Creating Document Classes, No Lifecycles Are Displayed

Symptoms: In the **Document Class: Info** page, no lifecycle names appear in the **Lifecycle** pull-down menu for a custom document type, such as aqua_regulated.

Resolution: To associate a lifecycle with a custom document type, you must use Documentum Application Builder (DAB) to modify the DCM DAR. Performing this task will ensure that the appropriate lifecycle is displayed as a configuration choice when configuring document classes.

For more information on creating, viewing, or modifying DARs using Headless Composer, refer to *EMC Documentum Composer User Guide*, version 6.5.

Sending Notifications is Very Slow

Symptom: Sending notifications to multiple users takes a long time.

Resolution: Use Documentum Administrator to set the dm_event_sender method to run asynchronously.

To set the dm_event_sender method to run asynchronously:

1. Connect to Documentum Administrator.
2. Open the **Job Management** node.
3. Choose the **Methods** node.
4. On the **Methods** page, choose the dm_event_sender method.
5. Select the associated **Launch Async** checkbox.

Stale Information Appears on DCM Pages

Symptom: Outdated or wrong information appears on DCM **Properties** pages or menus even after the information has been updated.

Resolution: The Web browser is using stale cached information. Ensure that your Web browser settings specify that to check for newer versions of stored pages on every visit to the page.

For example, in Internet Explorer, choose **Tools→Internet Options→Settings**, and click the radio button labeled **Every visit to the page**.

Unsuccessful Attempt to Check In or Convert Documents to Controlled Causes Application Server Errors

Symptom: An unsuccessful attempt to check in a controlled document, or to convert documents to controlled documents fails, resulting in a number of documents in a state where they are controlled but have no dm_policy object. This results in stack dumps from the Application Server.

Resolution: Delete all of the semi-converted controlled documents, and try the conversion or import operation again.

Deleting a Controlled Document Temporarily Disables the File Menu

Symptom: After deleting a controlled document, when you view the File menu, all of the options are grayed-out.

Resolution: To work around this problem, perform one of the following actions: click on the node to refresh the screen again; hit F5 to refresh; or click away to another node and return. Bug 120198 (SN 77840).

Appendix C

Adding or Modifying Reason Codes, Confirmation Codes, or Locale-Specific Value Assistance

DCM comes with a default set of reason codes, rejection/confirmation codes, and other codes, but you may want to customize these to fit your organization's requirements, or to make the text locale-specific. This appendix explains how use Application Builder to modify DCM DAR attributes. This appendix discusses the following topics:

- About the DCM DAR, page 303
- Adding or Modifying Value Assistance in the DCM DAR, page 303
- Changing the Locale of the DCM DAR, page 308

About the DCM DAR

To create or modify DARs, refer to *EMC Documentum Composer User Guide*, version 6.5.

The Dcm5Master DAR is located in the following repository folder: /System/Applications/Dcm5Master.

Adding or Modifying Value Assistance in the DCM DAR

This section provides information about modifying DAR properties to add or change value assistance. For example, you may want to modify the default set of reason codes or confirmation/rejection codes included with DCM.

DCM value assistance properties are stored in the /dcm/strings/valuassistance.properties file.

For more information on creating, viewing, or modifying DARs, refer to *EMC Documentum Composer User Guide*, version 6.5.

Specifying value assistance for an object type attribute

Value assistance is a list of values that a client program (such as Desktop) displays at runtime for an object attribute. A user can select a value from this list (or, if allowed, add a new one to it). There are two kinds of value assistance:

- Default value assistance − Values displayed when no value is selected; for example, when a new object is created.
- Conditional value assistance − Values displayed when a specific condition is satisfied; for example, when the values displayed in one attribute depend on the value selected in another attribute.

Value assistance for an attribute takes the form:

```
Condition 1: List of Values 1
...
Condition n: List of Values n
Default list of values (required)
```

Each condition is a Boolean Docbasic expression involving attributes of the type.

Note:

- Value assistance is not valid for attributes of the Boolean data type.
- Because of an underlying Docbasic limitation, you can only specify up to thirty-five conditional value assistance statements. If you require more than thirty-five conditional value assistance statements, you can use either a query or $value.

To specify value assistance for an object type attribute:

1. Double-click the attribute's name in the DAR explorer to open the attribute editor, and select the Value Assistance tab.
2. Click Add Default to enter the required default list of values.

   See Specifying conditional value assistance, page 305.
3. Click Add Conditional to add conditions and their associated lists.

   See Specifying conditional value assistance, page 305.
4. Use the U and D buttons to arrange the conditional elements.

Specifying conditional value assistance

You arrive here to specify either the default value assistance list or a conditional value assistance list.

To specify conditional value assistance:

1. For a conditional list, specify the condition as a Docbasic expression that resolves to true or false.

   Note: For repeating value attributes, you must use repeating value attribute keywords. For more information about repeating value attribute keywords, see Repeating value attribute keywords, page 307.

2. Select a radio button to specify the source of the list:
   - If Fixed List, enter the list into the text box, one value per line. Go to step 5.
   - If Query, Enter the query in the Query textbox. You can use the $value keyword to resolve attribute values at runtime. For more information about the $value keyword, see $value Keyword, page 305.
3. Enter the query attribute that provides the list values in the Query Attribute textbox (optional).
4. If you specified a default value, make sure that you have entered it in the fixed list or it is returned as part of the query.
5. Use the checkbox to specify whether or not cached queries are acceptable.
6. Use the checkbox to specify whether users can enter values that are not in the list.

$value Keyword

You use the $value keyword to resolve values of single-value attributes in conditional value assistance queries at runtime.

Note: If you use this keyword with a repeating-value attribute, the value in the first index position is always returned.

The syntax is:

```
$value(attribute)
```

where:

attribute is the name of the attribute.

There cannot be any spaces before the opening parenthesis.

At runtime, the $value(*attribute*) phrase is replaced with the specified attribute's current value (for example, the value selected for the attribute in the document's Properties dialog box). In some cases, you can use the $value keyword instead of multiple conditional value assistance statements. In addition, using $value dynamically allocates values instead of using hard-coded values in the conditional value assistance statements.

Example C-1. Using the $value keyword

A dm_document subtype, dozen_roses, has stem_length, color, and company attributes. If you want to display the names of companies (florists) depending on whether they have roses of the desired color and stem length, use this query (against a registered table) for the value assistance:

```
select florist_name from florists where total_length = '$value(stem_length)'
and true_color = '$value(color)'
```

If the registered table, florists, has these columns and values:

total_length	true_color	florist_name
80	Red	Fragrant Florists
80	Red	Lyrical Lilies
60	White	Flower Power Florists

and you select stem_length to be 80 centimeters and the color to be Red in a dozen_roses document's Properties dialog box, then this value assistance is displayed in the company attribute's field:

Fragrant Florists
Lyrical Lilies

If, instead, you select stem_length to be 60 centimeters and the color to be White, then this value assistance is displayed in the company attribute's field:

Flower Power Florists

Because the $value keyword dynamically allocates values, you do not need to change the query when more rows with new values are added to the registered table. For example, if two rows are added to the registered table so that it has these values:

total_length	true_color	florist_name
80	Red	Fragrant Florists
80	Red	Lyrical Lilies
60	White	Flower Power Florists
80	Red	Flower Power Florists
70	Hot Pink	Fragrant Florists

then the same query displays this value assistance when you select stem_length to be 70 centimeters and the color to be Hot Pink:

Fragrant Florists

To create the value assistance that corresponds to the previous example without using the $value keyword, you would need to create:

Value Assistance Statement	Condition	Query
Statement 1	length = 60 and color = ’White’	select florist_name from florists where total_length = 60 and true_color = ’White’
Statement 2	length = 70 and color = ’Hot Pink’	select florist_name from florists where total_length = 70 and true_color = ’Hot Pink’
Statement 3	length = 80 and color = ’Red’	select florist_name from florists where total_length = 80 and true_color = ’Red’
Default Value Assistance	N/A	select florist_name from florists

Notice that if you do not use the $value keyword and more rows with new values are added to the florists registered table, you need to add more value assistance statements to query for the new values.

Repeating value attribute keywords

You use repeating value attribute keywords to specify which runtime values in an object’s repeating value attribute to use in constraint or conditional value assistance expressions.

The syntax is:

```
attribute_name (keyword) operator {attribute_name | value}

{attribute_name | value} operator attribute_name (keyword)
```

where:

attribute_name is the name of the attribute. *keyword* is one of the repeating value attribute keywords—ANY, ALL, FIRST, LAST. *operator* is a valid Docbasic operator. *value* is a valid value for the attribute.

Keyword	Description
ANY	Uses any runtime value of an object’s attribute in the expression.
ALL	Uses all of the runtime values of an object’s attribute in the expression.

Keyword	Description
FIRST	Uses the runtime value of the first array position (that is, array position zero) in the expression.
LAST	Uses the runtime value of the last array position in the expression.

For example, the following condition and corresponding value assistance for the books attribute of a custom type called literature:

Condition	Value Assistance
literature_authors (ANY) = "Basho"	Narrow Road to the Deep North The Poetry of Matsuo Basho

displays the choices, "Narrow Road to the Deep North" and "The Poetry of Matsuo Basho," if at least one of the runtime values of a literature object's literature_authors attribute is the Haiku poet's name, Basho.

Changing the Locale of the DCM DAR

This section introduces locales, and explains how to modify a DAR to add or change a locale.

Changing the locale

When connecting to a repository, Application Builder, by default, uses the locale defined in the dmcl.ini file's client_locale setting; otherwise, it uses the locale from your computer's Regional Settings in the Control Panel.

If the locale is not published in the repository's data dictionary when connecting to a repository or changing locales, you are prompted to choose one that is published. When you connect to a 4.2.x or 4.4.x repository, the nls_key attribute's values of the dm_domain object determines the locales that are published. When you connect to a 5.x repository, the dd_locales attribute's value in the dm_repository_config object determines the locales that are published.

Changing your DAR's locale changes the locale for some object type attribute values and attribute values (most of which can be displayed to a user through a client interface to your DAR). You enter these attribute values in the specified locale. You can specify as many different locale values for a single attribute as you want—all of an attribute's locale values are saved to the repository. When you check in an object type, these attribute values are saved to the repository with their locale-specific text. When you change to another locale, you enter attribute values in that locale and save them to the repository—without overriding the previous locale's attribute values.

To change the locale:

1. Check in any modified object types.
2. Select View > Locale and one of the locales.

By default, English, French, German, Japanese, Korean, Italian, and Spanish are displayed. Use DQL to add your own locales—see the *Content Server Fundamentals Guide* for information about the data dictionary and localized text. To be able to select a locale, the locale-specific information must be published in the data dictionary. Use the Docbasic script dd_populate.ebs and server API publish_dd method to publish the locale-specific information in the data dictionary.

Note: The current locale for the DAR is shown with a check mark next to it in the Locale submenu as well as in the bottom right-corner of the status bar.

If you specified locale-specific information for a type and that locale is not enabled (specified in the dm_repository_config object's dd_locales attribute for a version 4.2.x or 4.4.x repository and in the nls_key attribute's values of the dm_domain object for a 5.x repository) on the target repository, then the type is installed, but without the locale-specific information. You will also be given a choice to continue or abort the DAR installation.

Locale

A locale represents a specific geographic region or linguistic group; for example, a national language such as Japanese.

Appendix D

Icons

This section illustrates the icons available in DCM:

- Icons Used to Designate Nodes, page 311
- Icons Common to Documentum Applications, page 311
- Icons Specific to DCM, page 313

Icons Used to Designate Nodes

The following icons designate nodes that are commonly used in Documentum web applications:

- : Inbox. Displays the tasks and notifications sent to you
- : Subscriptions are files and folders you want quick access to.
- ★: My Files. This node gives you quick access to files you have recently created, edited or checked out.
- : Categories. This node gives you access to files organized by you.
- : Administration. This node gives administrators access to system settings and administrative functions.

Icons Common to Documentum Applications

The following icons are commonly used in Documentum web applications.

Accessing Help:

- ?

Cabinets and Files:

- : A repository.
- : A cabinet.
- : Your home cabinet.
- : A folder.
- ℹ: Displays an item's properties when clicked.
- : Checks out selected files when clicked.
- : Edits selected files when clicked. If a file is not checked out, this also checks out the file.
- : Checks in selected files when clicked.
- : Cancels checkout of selected files when clicked.
- : Adds selected files to your clipboard when clicked.
- : Indicates that the object is checked out and that you own the lock.
- : Indicates that the object is checked out and that another user owns the lock.
- : Dragging this icon to your local computer creates a link to an item in the repository.

Selection Lists:

- ⊕: Clicking this adds an item.
- ⊖: Clicking this removes an item.
- : Clicking this edits an item.

Categories and Taxonomies:

- : A taxonomy.
- : A category.

Discussions

- : A discussion.
- : You have read all the object's discussion comments.
- : The object has discussion comments that you have not read.

Renditions and Transformations:

- : Displays additional file renditions when clicked. (This icon does not appear if a file's only other rendition is the thumbnail.)

Virtual Documents:

- : A virtual document.
- : A snapshot.
- : A frozen snapshot.

Workflows and Inboxes:

- : A notification.
- or : A task.
- : A high priority item.
- : A low priority item.
- : A workflow.
- : A workflow Template.
- : A package. A container for attaching a file. Indicates that the workflow you are sending has no attached files. You click this icon to attach files. Text adjacent to the icon tells you whether attached files are mandatory or optional.
- ▶: A currently running workflow.
- ⏸: A paused workflow.
- ■: A stopped workflow.

Administration:

- : A server.
- : A federation.
- : A user.
- : A user group.
- : A role.
- : A template for a user type.
- : A user session.
- : A job.
- : A method.
- : An alias set.
- : A format.
- : A type.
- : Storage.
- : Site publishing.
- : Content Intelligence Services administration.

Icons Specific to DCM

The following are the more commonly used icons in Documentum Compliance Manager:

- : My Signoffs. This node is where documents that require signoff reside.

- : DCM Administration. This node gives DCM administrators access to system settings and DCM administrative functions.
- : Business Applications. This node gives DCM administrators the ability to create or modify business applications.
- : Document Classes. This node gives DCM administrators the ability to create or modify document classes.
- : Auto-naming Schemes. This node gives DCM administrators the ability to create file naming schemes.
- : Business Rules. This node gives DCM administrators access to user list rules.
- : User List Rules. Same as Business Rules icon. This node gives DCM administrators the ability to create user list rules.
- : Relationship Types. This node gives DCM administrators the ability to create relationship types, and to delete existing relationship types.
- : Lifecycle Extensions. This node gives DCM administrators the ability to create signatories for a document.

Index