MICRO FOCUS | SERENA

Dimensions CM

Process Configuration Guide

Product version: 14.4

Publication date: December 2017

Table of Contents

Part 1

Overview of Dimensions Process Configuration

Part 1: Overview of Dimensions Process Configuration contains the following chapters.

Chapter 1

Process Modeling Concepts

In this Chapter

What Are the Components of the Process Model?

The components of the Dimensions® CM process model can be broadly categorized into the main areas listed below. The rest of this chapter describes these component areas in more depth.

- **Product organization:** The design part breakdown structure for a product. See "About Product Organization" on page 16.
- **Object classes:** Items, requests, baselines, and so on, which together comprise the product. See "About Object Classes" on page 18.
- **Lifecycles:** Approval cycles that ensure objects are operated on by the correct role and in a controlled manner. See "About Lifecycles" on page 21.
- **Privileges:** Assignments for users and groups the functions that users can perform. See "About Privileges and Roles" on page 22.
- **Role responsibilities:** Role assignments that determine the product areas in which users can operate. See "About Privileges and Roles" on page 22.
- **Attributes:** System and user-defined information for each object.See "About Attributes" on page 23.
- **Relationships:** Associations between objects that determine how the objects may impact each other. See "About Relationships" on page 29.
- **Rules:** Change management controls that govern the progress and coordination of items with requests. See "About Change Management Rules" on page 32.

About Product Organization

You can view the organization of a product from several different management and technical perspectives. A management perspective is an organizational view that focuses on the team members, their areas of responsibility, and who they report to. A technical perspective is a logical view that focuses on how product components are organized and relate to each other.

To represent how a product is organized from both of these perspectives, Dimensions CM models the system as a structure of design parts.

About Design Parts

A *design part* is an object that represents some conceptual component of the system. Each design part owns a set of items associated with that part of the product, such as a functional specification document, a test specification, or source code files.

Design parts are organized into a design part structure by defining hierarchical relationships between them. Such relationships can be used to express the organization of your development project to Dimensions CM, providing a structure in which you can assign roles and responsibilities to specific users. Grouping items within a meaningful design part structure makes it easier for users to understand how things fit together and what role they have to play in the success of the development project.

Design Part Relationships

Dimensions CM has two predefined types of relationship between design parts, *breakdown* and *usage*.

Breakdown relationships describe ownership of a design part by another design part. Each design part has a parent that is defined when you create it. Dimensions CM uses breakdown relationships to determine the inherited responsibilities for the items or requests associated with a particular design part segment.

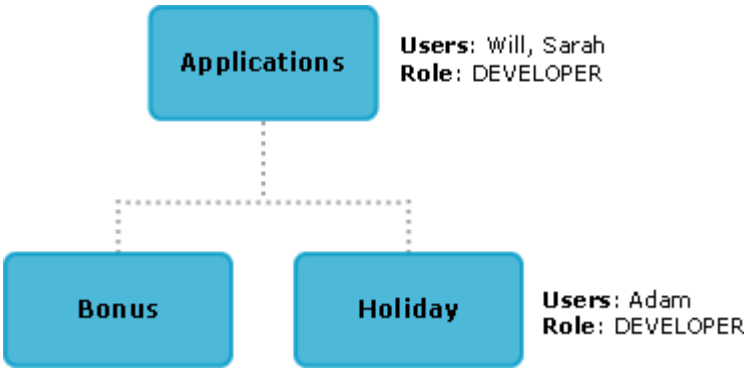
Usage relationships express dependencies between design parts, to show where design parts are reused in the design structure.

Design Part Lifecycle

Design parts follow a fixed system-defined lifecycle called LC_PART. You cannot assign user-defined lifecycles to design parts.

Example: PAYROLL Product

For example, the BONUS and HOLIDAY design parts are children of the APPLICATIONS design part. These design parts are in a breakdown relationship with APPLICATIONS. This means that team members with roles on APPLICATIONS will inherit the roles on BONUS and HOLIDAY, unless the same roles are assigned to other users on these lower level design parts.



In this example, Will and Sarah inherit the DEVELOPER role on BONUS but do not on HOLIDAY, since Adam is assigned the DEVELOPER role on that design part.

About Design Part Variants

A design part variant represents an alternative implementation of a design part. Typically a variant represents a specific market requirement or demand. For example, a team might develop a product that has two different markets, American and European. To represent those markets, the team creates a set of design part variants specific to each market.

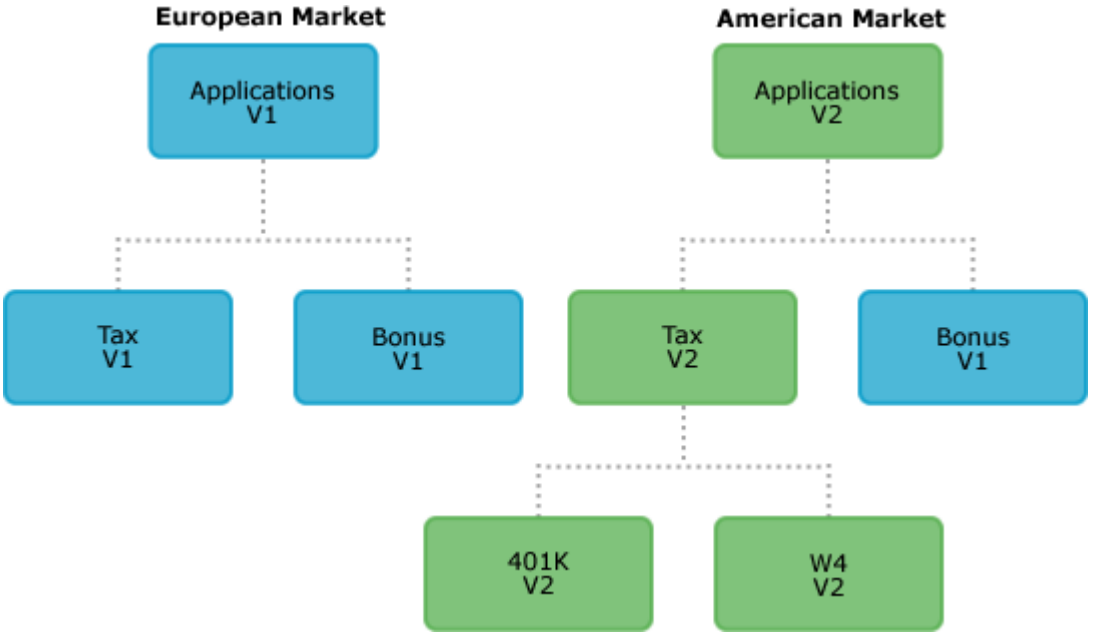
You can create variants at any level within the product structure. A variant will appear at the same level as its parent design part.

Example: Variants in the PAYROLL Product

The illustration below shows a simple set of design part variants that represent European (V1) and American (V2) markets. As shown, Dimensions CM allows you to:

- Create variants of the same design parts: APPLICATIONS, TAX

- Use common design parts: BONUS
- Create new design parts specific to a variant: 401K and W4



About Object Classes

Dimensions CM functionality includes a comprehensive set of well-defined object classes. These object classes allow you to place a diverse set of objects, such as files, projects, deployment areas, and bug reports, under Dimensions CM configuration control.

The Dimensions CM data model provides support for the following object classes:

- Design parts (discussed previously)
- Items
- Requests
- Projects/streams
- Baselines
- Releases
- Customers

Items

Items are the components by which a product is implemented and described. Typical items are source code files, executables, program resources such as graphics, and specification documents. Entire directories of files can also be items. Items are versioned objects with full lifecycle support.

Each item is owned by a single design part and may be used by one or more other design parts. Access to an item is controlled by the role assignments for the design part that owns it and/or the design parts above that.

Requests

When a problem, bug, or enhancement arises, you can create a *request* within Dimensions CM to represent that change or request. Requests help in the development lifecycle by automating the change process and reducing corrective cycles. Requests also have lifecycles.

Requests ordinarily affect one or more design parts and have relationships with other objects such as items and design parts for the purposes of change authorization, change tracking, and impact analysis.

Work Packages

Requests can be grouped into a *work package*, which is a high-level request for a set of interrelated changes or tasks. A work package can only be closed once all requests it contains are closed.

Projects and Streams

A *project* or *stream* is a dynamic collection of items relevant to a development activity. Examples include mainstream, customization, and maintenance. Projects and streams differ in the working practices that they are designed to support.

Projects:

- Follow a "lock, modify, unlock" mode of working
- Can contain parallel sets of changes, or branches, at the same time.
- Can support operations that apply to individual items, such as check in or check out.

Streams:

- Follow a "copy, modify, merge" mode of working
- Only contain one set of tip revisions, or branches, at any one time
- Only support operations that process multiple items or folders, such as update or deliver.

NOTE A stream is configured in the process model as a project of type "STREAM". References to projects in the Administration Console and in his book refer to projects generally, and therefore may apply to streams.

To support parallel development using projects, Dimensions CM provides interactive facilities for merging projects, as well as for identifying and resolving the conflicting files in the resulting merged project.

In the case of streams, you perform merge operations in the work area, using the Update function, and the merge tool where merging file content is necessary.

Global Project

The global project contains all of the projects/streams and items for the product. Item operations that are reflected in the global project include create, update, edit, check in, and delete.

Project/Stream Folders

Every project or stream has a folder structure associated with it that consists of subfolders and items. This folder structure may differ from one project/stream to another.

When a structural change is made to the project/stream directory, the change affects only that project/stream. For example, any new revision of an item appears only in the project or stream in which it is created.

Default Work Area

Users associate a work area to a project/stream, which is a specific setting to that user alone. A work area identifies a location outside the Dimensions CM repository for item files that are copied to and from the repository by functions such as deliver, update, check out, or check in. Items are placed in the work area relative to their path in the project/stream. A work area can also be a Dimensions CM-configured object that references a location on disk.

Baselines

A *baseline* is a snapshot of a design part segment at a given point in time. This snapshot is comprised of related design parts and item revisions selected from the current project/stream. There are three categories of baseline:

- **Release:** A frozen configuration that captures single versions of items in the design part segment. This configuration typically represents a development milestone within a product lifecycle and is only done once per project/stream. Item revisions in this configuration cannot be modified unless the baseline is deleted.
- **Design:** A configuration that captures all revisions of all items in the design part segment. This configuration is used for reporting purposes. Item revisions in this configuration can be modified.
- **Archive:** A special type of release baseline that captures item revisions using one or more *ALL rules in the baseline template. This configuration is used for archive and retrieval by Dimensions CM.

You can create baselines using a rich set of filtering criteria, including design part, item type, and status,. You can also create baselines based on requests. When you create a baseline, Dimensions CM records the status of all items included in it. All items in a baseline are fully preserved for future use, such as rebuilding the entire configuration or providing the basis for a new maintenance release.

Baselines are typically used to freeze a configuration within a project/stream for test, integration, build, or release purposes. They can also be used to meet the audit requirements for DOD-2167A, ISO9000, SEI Level accreditation, and contractual commitments to milestone payments.

Like other Dimensions CM objects, baselines have lifecycles. The status of a baseline reflects the status of the configuration as a whole, for example, PASSED TEST, FAILED TEST, or READY FOR RELEASE. It provides important management information on the progress of the project/stream.

Releases

A *release* is a set of items for the product to be issued to customers. When a baseline passes testing, it may be considered ready for release to customers using the Dimensions CM release management facilities. These facilities enable a full or delta configuration of the product to be copied from the protected Dimensions CM environment to a release directory. Dimensions CM provides templates that can be applied to the baseline in order to select the objects for release.

Customers

A *customer* is the recipient of a release. Dimensions CM can record details on customers who have received releases of a product and the specific releases which have been issued to them.

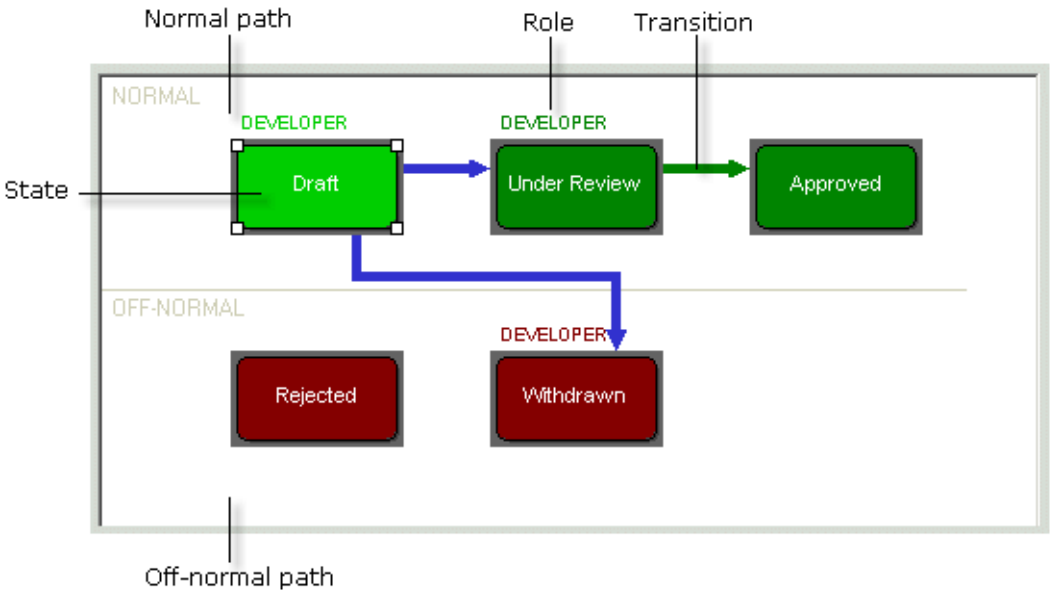
About Lifecycles

A *lifecycle* defines the approval process for object types in Dimensions CM. You can define a lifecycle independently and then associate it to an object type in the product's process model, or you can define a lifecycle from within the context of an object type. Each object will follow the lifecycle of its associated object type. The status of an object is its current state in its lifecycle.

You can define lifecycles for requests, items, and baselines.

A lifecycle comprises:

- **States:** Approval levels through which an object must pass. States typically reflect process states such as development, test, and release.
- **Transitions:** The permitted movement of an object between one state and another state. Transitions determine the sequence of states for a lifecycle.
- **Roles:** The role holders who can move an object from one particular state to another, which is known as actioning.
- **Normal and off normal paths**: Paths that indicate how an object is progressing through the lifecycle. The normal path is a linear succession between lifecycles states that represents a successful path, while an off normal path represents a failure or a need for rework.

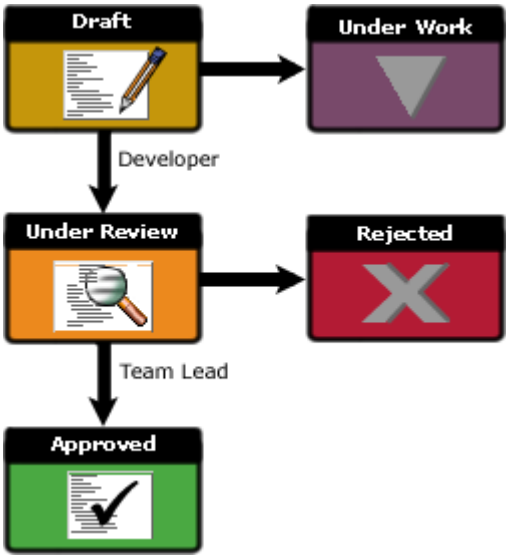


Lifecycles ensure that an organization's processes are followed and are auditable. Lifecycles are user-defined so that they reflect the way in which the organization works.

Example: Document Lifecycle

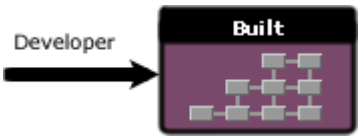
The following diagram illustrates a simple item lifecycle for a document. Only a user with the Developer role can action the object from DRAFT to UNDER REVIEW, while only a user with the Team Leader role can action it from UNDER REVIEW to APPROVED. Although not shown in this example, note that you can assign more than one role to a transition.

The DRAFT, UNDER REVIEW, and APPROVED states form the normal path for the lifecycle. WITHDRAWN and REJECTED are off normal.



Example: Single-State Lifecycle

For item types, you can create lifecycles that consist of only a single state. As in the example below, a single-state lifecycle may be used for intermediate derived files such as compiled output files, which do not normally follow an approval procedure.



A single-state lifecycle has one normal transition with the same begin and end state. An item assigned to a single-state lifecycle will not appear on any item inboxes.

Note that you cannot action an object with a single-state lifecycle.

About Privileges and Roles

The operations you can perform, and the object classes you can perform them on are defined in the Administration Console, Users and Roles function. What you can do is determined by the *privileges* and/or *role assignments* that have been defined for your user.

A privilege is the ability to perform a certain operation on a particular class of object, for example to create a project, or to action a request. It can also be the ability to perform a general administrative function, such as to manage lifecycles or privileges.

A role is a set of privileges that can be assigned to one or more users, enabling them to carry out activities required for a certain position within the organization, for example Developer, or Team Leader.

For details about privileges and roles and how to use them, see "Privileges and Roles" on page 39.

About Attributes

Items, requests, baselines, design parts, users, and products have a set of *attributes* that are specified in the process model. Attributes record and track important configuration information such as creation date, owner, status, description, and last updated. You can view, filter, and report on objects using these attributes.

Some attributes are inherent or system-defined and are always present for a particular type of object. You can also set up user-defined attributes specifically for your environment.

User-Defined Attributes

A user-defined attribute is a custom attribute that you set up for an object type. You can define user-defined attributes to suit your processes.

A user-defined attribute consists of:

- **Global properties**

  These are properties of the attribute that are independent of the assigned object type. Examples of global properties include the attribute name and maximum length.

- **Local properties**

  These are properties of the attribute relating to the specific object type. Examples of local properties include the user prompt and default value.

For each object class, you can define a number of user-defined attributes in a base database. For an Oracle or SQL Server database this can be up to 996. For greater flexibility, Dimensions CM allows you to reuse the same attribute with different local properties across object types. This enables you to derive more attributes from the maximum number allocated to each object class.

Example: Reusing Attributes Across Object Types

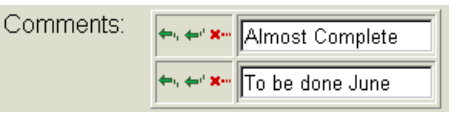
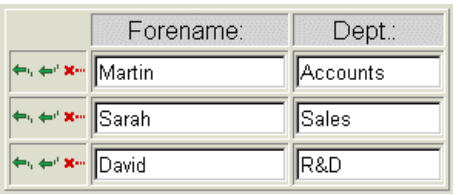
Stephanie, the administrator, wants to define an attribute named NOTES for the RMR and CR request types. For the RMR type, she wants to define a user prompt called "Review Notes" and for the CR type, she wants the user prompt "Inspection Notes." Stephanie creates the NOTES attribute for the RMR type with the desired user prompt, and then assigns NOTES to the CR type and changes the user prompt. In this way, Stephanie can modify the same attribute for use with two object types without creating two separate attributes.

Generally, each attribute definition includes:

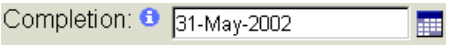
- The name of the attribute.
- The maximum length of the attribute value.
- The user prompt that appears as the field label for the attribute in the GUI.
- The data type for the attribute value, described below.
- The type of attribute, described below.
- The display length and height of the attribute.
- An optional valid set of values, described in "Valid Sets of Values" on page 24.
- Attribute update rules and role sections, described in "Attribute Update Rules" on page 25.

Attribute Types and Data Types

The type of attribute determines if the user can enter a single attribute value, a row of values, or a table of values.

Type	Description	Example
Single-field, single-value	One value for the attribute.	
Single-field, multi-value	Multiple rows of values can be assigned to the attribute.	
Multi-field, multi-value	Multiple rows and columns can be assigned to the attribute. In the example, an attribute called Contact could consist of a table of Forename and Dept. values.	

The data type determines what kind of information the user can enter as the attribute value. Below are the valid data types.

Data Type	Description	Example
Character	One or more lines of text.	
Date	The date in format DD-MMM-YYYY. You can enter a date or choose one by clicking the calendar button.	
Number	A real number or integer.	

Valid Sets of Values

A valid set is a discrete list of values that can be assigned to a particular attribute. A valid set provides the permitted values that a user can select when specifying an attribute for

an item, request, baseline, design part, or user. Valid sets ensure that users only enter valid combinations of data into attributes.

A simple valid set contains a single column that lists all of the acceptable values for one attribute. For example, a valid set for the attribute Urgency could contain values High, Medium, and Low. A more complicated valid set could consist of up to eight columns, which can limit the combinations of values for up to eight different attributes. When the user completes a field belonging to the valid set Dimensions restricts the permissible values for the other fields in that valid set accordingly.

Example: Multi-Column Valid Set

Stephanie, the administrator, creates a valid set called CUSTOMER_NAME that contains the following columns and rows:

CUST_SITE	CUST_NAME	CUST_CONTACT
Azur SA	Marie Sancerre	10 46 92 21 56
High Tech Inc	Jane Doe	810 544 6771
Technik AG	Klaus Schmidt	89 5904 7112
Widgets Ltd 2	Joe White	01279 999001
Widgets Ltd 2	Fred Gray	01279 999001

Stephanie then assigns the first column of the valid set to the attribute CUST_SITE, the second column to CUST_NAME, and the third to CUST_CONTACT.

Will, a developer, selects the Widgets Ltd 2 value for CUST_SITE, and then chooses between the permitted values, Joe White and Fred Gray, for CUST_NAME. Sarah, another developer, selects High Tech Inc for CUST_SITE and the values for the other attributes are filled in automatically.

NOTE The order in which the fields are evaluated is determined by the setting of the `DM_SMART_ATTR_VALIDATION` symbol in the `dm.cfg` file. If this option is set, the evaluation of fields in the valid set only takes place from left to right, so that values to the left of a field are not restricted by what is entered, only values to the right. See the *System Administration Guide* for more details on the `DM_SMART_ATTR_VALIDATION` symbol.

Attribute Update Rules

Users can generally update attributes if they have the required update privilege. In addition, Update rules may specify whether a particular role can view and update a specific attribute and whether an attribute value is required. Each rule can apply throughout the object's lifecycle or only when an object is in a particular lifecycle state.

For example, for requests of type Problem Report, you could specify that the SEVERITY attribute is updateable only by a user with the Reviewer role during the DEFINED state.

The following list summarizes update rule applicability:

- Attributes cannot be updated unless it is in the user's inbox.
- If there are no rules for an attribute, then the attribute will appear in the Dimensions CM GUIs as writable and not required.

- If there are attribute rules defined for a particular state, then by default all attributes are not writable and not required at that state.

  However, if a rule exists for an attribute with the **Writable at the From State** setting enabled for a particular state, then this rule will take precedence and the attribute will be writable.

- If a rule is defined for an attribute as **Required when actioned to the To state**=yes for a given **From State** and **To State** pair, then the attribute will be displayed as required in the Dimensions CM Action dialog box when making that transition.

  NOTE The Attributes tab in the Edit Attributes dialog box shows required attributes based on the next lifecycle state (normal or off normal).

- Change Managers see all requests as writable and not required, and Product Manager see all items and baselines as writable and not required.

Example: Defining an Update Rule for Actioning

Stephanie wants to create a rule so that a particular attribute is required during a transition between the Approved and Implemented states. She chooses the attribute name and role, and then sets the following:

- **From State**: Approved
- **To State**: Implemented
- **Display in this role section**: set
- **Writable at the From state**: set
- **Required when actioned to the To state**: set

As a result, a user with the specified role must define a value for the attribute in the Action dialog box when actioning the object to the Implemented state. The user could also define the value in the Edit dialog box before actioning.

Ordering of Attribute Rules

In the current version of Dimensions CM, by default, attribute rules are ordered by:

1. `From` state, then,
2. `To` state, then,
3. `Rules` (not Required ahead of Required), and finally,
4. `Role`.

In the example below the attributes rules relating to the attribute PLAN_FINISH are ordered as follows:

Attribute	From	To	Role	Rules
PLAN_FINISH	REVIEW	WORK	CRB	Required, Writeable
PLAN_FINISH	REVIEW	WORK	IMPLEMENTOR	Writeable
PLAN_FINISH			TESTER	

NOTE In versions of Dimensions up to, and including 10.1.3, the attribute rules were ordered by role; and rules that were non-mandatory took precedence over mandatory rules. This meant that an attribute rule with no `From` and `To` states would take precedence over one that had specified `From` and `To` states.

You can revert to this behavior, by adding the following symbol to `dm.cfg` on the Dimensions CM server machine:

DM_ORDER_ATTR_RULES_BY_ROLE

For details see the System Administration Guide.

Attribute Role Sections

In the attribute update rules, you can also specify if you want an attribute to appear in a role section. A role section is a list of attributes that are pertinent to the holders of a particular role in an object's lifecycle. When viewing and updating attributes, users can choose one of these sections in order to reduce the list to what they want to see.

For example, a user with the role Developer will normally be interested in a certain subset of the user-defined attributes of an item revision, whereas a user with the role Quality Assurance will be interested in a different subset.

Every attribute defined for an object type can be included in, or excluded from, any role section. Note that role sections simply classify attributes according to roles; they do not specify permissions. Any authorized users are permitted to view any of the role sections, regardless of whether they hold the corresponding roles.

Example: Defining a Role Section

Stephanie wants to create a rule to set a role section for a particular attribute in the Allocated state. She chooses the attribute name and role, and then sets the following:

- **From State:** *Allocated*
- **Display in this role section:** *set*
- **Writable at the From state:** *set*
- **Required when actioned to the To state:** *unset*

As a result, users with the specified role can view the attribute in their role section and update it when the object is in the Allocated state.

The option that determines how attributes are set between different revisions of the same item.

Copying Attributes between Item Revisions

There is a setting for an item attribute that determines whether, and how the attribute is copied when a new revision of an item is created. The options are:

Stored per revision, default as specified: when a new revision of an item is created, Dimensions CM does not copy the attribute values from the previous revision but initially sets them to the specified default value according to the Default Value above. You can subsequently override them for a specific revision.

Stored per revision, default to previous revision: when a new revision of an item is created, Dimensions CM initially sets the attribute values to those from the previous revision, but you can subsequently override them for a specific revision.

Same for all revisions: Dimensions CM sets the attribute values for all revisions of the same item to the same value. When you update the value of an attribute for any revision, Dimensions CM will also update that attribute for all other revisions.

This is determined when an attribute is assigned to an item type. See "Assigning an Attribute to an Object Type" on page 146 for details of how to do this.

About Authentication Points

Dimensions CM provides the ability within the process model to define lifecycle states or attributes as *sensitive*. This means that when a user attempts to action an item or request to or from a sensitive lifecycle state, or attempts to update a sensitive attribute, this results in an authentication point, requiring an electronic signature. The user is presented with an Authentication Point dialog box, requiring them to re-enter their Dimensions CM password in order to complete the update. Dimensions CM maintains an audit trail of all failed and successful authentication attempts.

Authentication points are generated only for requests and items, and occur when you:

- Action an item or request to a sensitive lifecycle state, unless the sensitive state is the first normal lifecycle state and you are creating the object.
- Action an item or request from a sensitive lifecycle state unless the sensitive state is the first normal lifecycle state and you are creating the object.
- Modify the value of a sensitive attribute unless the modification occurs as part of creating the object.
- Delete an item or request when it is at a sensitive lifecycle state.

Authentication points are not generated for held requests.

Authentication points are also generated in the Administration Console when you:

- Change a lifecycle state's sensitivity, unless this is done as part of creating the lifecycle state.
- Change an attribute's sensitivity, unless this is done as part of creating the attribute definition.
- Delete a sensitive lifecycle state, including automatic deletions that occur when you delete the last transition involving the state or delete the entire lifecycle.
- Delete a sensitive attribute definition.

See "Assigning an Attribute to an Object Type" on page 146 for details of how to set attributes as sensitive.

See "Setting the State Properties for an Object Type" on page 192. for details of how to set lifecycle states as sensitive.

About Relationships

Relationships provide traceability between items, design parts, and requests. You can use relationships to record whether an item is affected by an issue reported in a request or to specify other items which are derived from, or which are used to derive, an item. You can also unrelate items, if necessary, depending on the state the item is in.

Relationship types are system-defined or user-defined. If you are the Product Manager, you can set up user-defined relationship types in the process model.

Item-Design Part Relationships

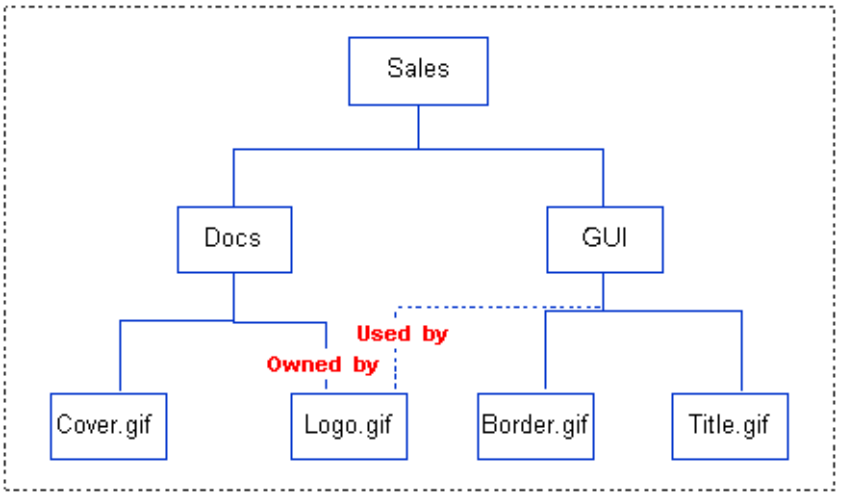
Items are related to design parts to establish where an item belongs and where it is used within the product. An item is either owned or used by a design part. Only one design part can own an item, but several design parts can use it. Item-design part relationships apply to all revisions of the item.

Item-design part relationships are:

- **Owned.** A single design part can own any number of items. This relationship indicates the part of the product in which the item belongs.
- **Used.** An item can be used by any number of design parts. This relationship indicates that the item is reused in different parts of the product.

Example: A Reused Item

The item *Logo.gif*, which is the company logo bitmap in the product Sales, is owned by the design part Docs and is also used by design part GUI.



Item-Item Relationships

Use item-item relationships to signify a dependency or connection between an item and other items. Items can be related to one another by system-defined or user-defined relationships. The system-defined relationships are determined by your process model and cannot be changed.

System-defined item-item relationships are:

- **Made Of.** An item is made of those item revisions from which it was created by a build process.
- **Made Into.** An item is made into those item revisions that are built from it.
- **Created From**. An item is created from those earlier item revision(s) from which it made by merging or editing.

User-defined relationships can also be defined to identify relationships that are specific to your organization or product.

Notes on Relating Items

For two items to be related to each other:

1. The item types must be allowed to be in a valid relationship. This relationship is a parent-child one.
2. The child item must be open and the parent item must be held or open.

Example: A Common Item-Item Relationship

At Program Utilities Inc., executable programs, such as their flagship product Process Utility, are usually built from a number of object files, such as Setup, task1, task2, and Finish. The system-defined relationship Made Of is used to relate these types of items.

Request-Request Relationships

Use request-request relationships to signify a dependency or connection between a request and other requests. Requests can be related to one another by system-defined or user-defined relationships. The system-defined relationships are determined by your process model and cannot be changed.

System-defined request-request relationships are:

- **Dependent**. A request can be dependent on another request for closure.
- **Information**. A request can have an informational relationship to one or more requests. This means that the requests are related for information purposes only.

User-defined relationships can also be defined to identify relationships that are specific to your organization or product. These relationships are based on the system-defined **Dependent** or **Information** types.

Notes on Relating Requests

For two requests to be related to each other:

1. The request types must be allowed to be in a valid relationship. This relationship is a parent-child one and must be defined for both Dependent and Information relationship types.
2. For both Dependent and Information types, the child request must be open and the parent request must be held or open.
3. For the Dependent type, the request types must both have rules enabled or must both have rules disabled. See "About Change Management Rules" on page 32 for information.
4. For the Dependent type, if rules are enabled, then the relationship can only be established (or dissolved) if the parent request is in the ANALYSIS or An+Work phase, as described later.

Example: Dependent Requests

In-Car Software Inc. has a product set up in Dimensions CM called NAVIGATION. A bug has been found such that the variable direction is sometimes being set to the wrong value. A request is raised for this.

It is found on examination that both the animation and the sound software will need to be changed. Two new requests are created for these functional areas, and they are related to the initial request as Dependent.

Item-Request Relationships

You can relate items and requests in order to track which items need to be changed and to see whether the items have been modified or approved. Items and requests can only be related to one another by system-defined relationships.

NOTE Valid relationships are not effective if CM Rules have not been set up for a particular request type. Therefor any request, whose type rules are disabled, may have any product item revision related to it, provided its item type rules are also disabled.

Item-request relationships are:

- **Affected By**. An item (or item revision) can be affected by one or more requests. This means that the item(s) need to be modified to implement the changes required by the request(s).
- **In Response To**. An item (or item revision) can be in response to one or more requests. This means that these versions of the items are intended to satisfy the required changes as requested in the request.
- **Information.** An item (or item revision) can have an informational relationship to one or more requests. This means that the change is related for information purposes only.

Notes on Relating Requests and Items

For a request and item to be related to each other:

1. If CM Rules are switched on, the item and request types must be allowed to be in a valid relationship. This relationship is a parent-child one, with the request type always the parent of the item type.

   You can relate any item to any request if CM rules are off.

2. For all relationships types, the item and request must both be open.

3. For the Affected By and In Response To types, the request type and the item type must both have rules enabled or must both have rules disabled. See "About Change Management Rules" on page 32 for information.

4. For the Affected By type, if rules are enabled, then the relationship can only be established (or dissolved) if the request is in the CREATE, ANALYSIS, or AN+WORK phase.

5. For the In Response To type, if rules are enabled, then the relationship can only be established (or dissolved) if the request is in the WORK or AN+WORK phase.

Example: Relating a Request to Item Revisions

In-Car Inc. has decided to change its web site to American spelling. Sally was assigned the task of combing through the pages and submitting requests for instances of different spellings. One of the resulting requests is a request to change "catalogue" to "catalog".

The request passed to Jane, the web site manager, who identified the page as *Product_List.html*. She relates the existing version, revision 3 of this item to the request as Affected By.

The changes are assigned back to Sally. When she checks out this item to create revision 4 and update the file, this new revision is automatically related to the request as In Response To.

About Change Management Rules

You can enable change management (CM) rules in the process model in order to gain tighter control over planning, implementing, and closing changes. One set of rules, defined for a request type, organizes a request's lifecycle states into phases. These phases determine the relationships that can be established in each state, as well as which other tasks can be performed. The rules also determine how parent-child requests affect each other.

The other set of rules, defined for an item type, connects a request lifecycle and an item lifecycle. These rules require item revisions to be related to requests at various states in its lifecycle. In this way, CM rules bring version management and change management closer together, ensuring that related items and requests progress through their lifecycles in tandem.

You enable rules for specific item types and request types in a product. This means that the same lifecycle can be reused within the same product or across products with different rules applied per object type.

There is a per-project override you can apply that can cause item operations taking place within that project to act in one of the following ways:

- To take effect according to whether the CM rules are enabled on the individual item and request types or not
- To behave as if CM rules were enabled for all item and request types
- To behave as if CM rules were disabled for all item and request types.

These overrides do not apply to streams.

Most rules are related to states on the normal path through the relevant lifecycle.

Valid Relationships

Note that rules can only be applied to object types that are allowed to be in valid relationships. For items, you can specify relationships between two item types. For requests, you can specify relationships between two requests types or a request type and an item type.

Rules for Request Lifecycles

When you apply rules to a request type, you must group its lifecycle states into phases. A phase is a system-defined stage that determines what kind of processing can take place while a request is at that state.

The phases are defined below.

Phase	Description	How Set?	Permitted Tasks
HELD	Specifies the stage in which an originator prepares and holds a request. A held request is not visible to any other users. It appears in the originator's inbox with the status $TO_BE_DEFINED. This phase does not correspond to an actual lifecycle state.	Automatically applies to held requests.	■ Relate design parts ■ Update attributes
CREATE	Specifies the normal lifecycle states (if any) which precede the ANALYSIS phase.	Automatically applies to new requests.	■ Relate design parts ■ Relate items as Affected ■ Update attributes ■ Add action descriptions ■ Action
ANALYSIS	Specifies the normal lifecycle states in which you analyze and plan the implementation of the change. Good analysis during this phase ensures that you understand the change and that you identify the dependencies and items that require changing.	Select the normal lifecycle state that signifies the start of the ANALYSIS phase. If you select the first state in the lifecycle, then the CREATE phase does not exist; its activities are merged with those in the ANALYSIS phase.	■ Relate design parts ■ Relate items as Affected ■ Relate requests ■ Update attributes ■ Add action descriptions ■ Action
WORK	Specifies the normal lifecycle states in which you implement the change. You can work on items that have been related as **Affected** in this phase. On check in, these items are automatically related as **In Response To**. If item rules require that a request is specified on check out, and an item has not yet been related as **Affected**, then Change Manager must relate the item as **In Response To** in this phase.	Select the normal lifecycle state that signifies the start of the WORK phase. You cannot select a state that precedes the state specified for the ANALYSIS phase.	■ Relate items In Response To ■ Update attributes ■ Add action descriptions ■ Action

Phase	Description	How Set?	Permitted Tasks
FROZEN	Specifies the normal lifecycle states (if any) which follow the work phase but may precede the final normal state. This phase includes the states in which testing, baselining, and qualification of the software may take place. Consequently, no changes are permitted to any of the relationships, nor to any of the related item revisions. The related item revisions are also frozen by this stage, through baselining and/or actioning. **Note:** The restrictions on updating relationships do not apply if the user has the CHANGE MANAGER role.	Select the normal lifecycle state that signifies the start of the FROZEN phase. If you select the final normal state, then the FROZEN phase does not exist. The request becomes frozen only when it is closed. You cannot select a state that precedes the state specified for the WORK phase.	■ Update attributes ■ Add action descriptions ■ Action
CLOSED	Specifies the final normal state. No further changes are permitted.	Automatically applies to the final normal state.	None
REJECTED	Specifies all final states except the final normal state. No further changes are permitted.	Automatically applies to any off normal final states.	None
OFF NORM	Specifies all other states that are not in the normal lifecycle and are not final states.	Automatically applies to all off normal, non-final states.	■ Update attributes ■ Add action descriptions ■ Action
AN+WORK	A composite phase that specifies all normal lifecycle states between the CREATE phase and the FROZEN phase, as an alternative to assigning separate ANALYSIS and WORK phases. Using this phase works best on small teams, where some implementation may be allowed during analysis. For larger teams or projects, separate ANALYSIS and WORK phases are recommended for more control.	Select the same state for the WORK and ANALYSIS phases.	■ Relate design parts ■ Relate items as Affected ■ Relate items in Response to ■ Relate requests ■ Update attributes ■ Add action descriptions ■ Action

NOTE A request is described as open when it is in any phase except HELD, CLOSED, or REJECTED.

Parent-Child Rule

In addition to the phases outlined above, you can define a rule regarding parent-child requests. This rule specifies the minimum state that the request (the child) must reach so that the parent request can close. Additionally, the parent request can close if the child document is actioned to the REJECTED phase.

Roles and Permitted Tasks

The tasks permitted in each phase can only be performed by the Dimensions CM users who hold a role on the current state. In the case of the HELD phase, the only authorized

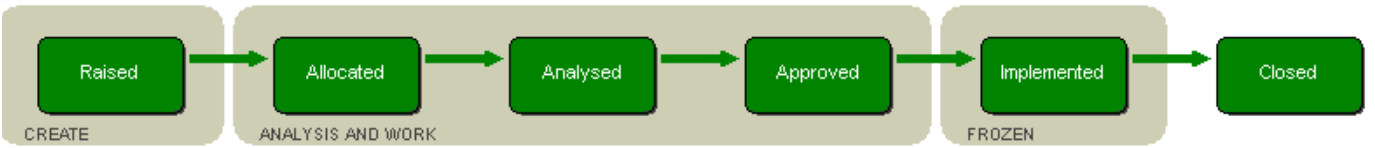
user is the originator. If the leader responsibility is enforced, then only the users with the leader responsibility can action the request and update its attributes.

Change Managers have additional permissions. They can relate and unrelate objects at any lifecycle state except the final state, as well as action any request to any of its lifecycle states. They can also reactivate a closed or rejected request by actioning it to a non-final state.

Example: Change Request Phases

The following diagram shows how the normal lifecycle states in a change request (CR) type are mapped to phases:

- Analysis phase starts with state: Allocated
- Work phase starts with state: Allocated
- Frozen phase starts with state: Implemented



Rules for Item Lifecycles

For item types, you can enable CM rules so that the development of items is subject to the required level of change control and authorization. These rules provide the flexibility to introduce change control on items at whatever stage in the process is most applicable to them.

The rules for item types and their normal lifecycle states are described below. Note that rules are only applicable when the related request is on a normal lifecycle path.

Rule	Description	Sample Scenario
Creation	Requires a request to be specified when creating a new item. When the item is created, it will be related to the request as **In Response To**.	You need to enforce change control right from the beginning of a project.
New Revision	Specifies the state in the item's lifecycle in which all new item revisions based on the revision at the specified state must be related to a request as **In Response To**. If you specify the first lifecycle state, this requires every new revision to be related to a request.	You want to enforce change control after a beta release of the project.

Rule	Description	Sample Scenario
Action	Specifies the state in the item's lifecycle in which an item cannot be actioned to the next normal state unless it is related to a request as **In Response To**.	You want to make sure that only the appropriate revisions progress far enough in their lifecycles to be baselined. You set this rule to catch item revisions from progressing without approval.
Closure	Specifies the state in the item's lifecycle that an item must reach in order for a related request to close.	You do not want to allow a request to be closed while any related item revision has not reached a quality state in its lifecycle.

NOTE For the New Revision and Action rules, a request must be related In Response To, which means that a previous revision of the same item must first be related as Affected. If you cannot fulfill this latter condition, then the Change Manager is allowed to override it when establishing the In Response To relationship.

Examples of Rule Usage

The following examples show how two development teams apply CM rules for item and request types to meet their needs.

Example: A Highly Controlled and Visible Project

Team A is working on a highly controlled and visible project that requires a high level of control and visibility. From the start of the project, the team needs to use CM rules to ensure that no new revision is created unless it is related to a request and approved.

For item types, the team sets these rules:

- **Creation rule:** Set so that any new item requires a request.
- **New Revision rule:** Set to the initial state so all subsequent revisions require requests.
- **Action rule**: Set to a state just prior to a test state so that only controlled item revisions can be included in release baselines. This rule would also be used to catch any item revisions that might become unrelated from a request for any reason.
- **Closure rule:** Set to a quality state in the item's lifecycle so a request can only close when the item reaches this state.

For request types, the team:

- Specifies different normal lifecycle states for the ANALYSIS, WORK, and FROZEN phases for full control.
- Specifies that a child request must reach a quality state before the parent request can close.
- Enforces the Leader role responsibility to maximize control.

Example: Prototype Project

Team B, a small group of designers, must produce a prototype of a system in a short amount of time. The goal is to demonstrate capability and win a contract. After winning the contract, the team will expand to a larger development group, and will need to impose more product controls to ensure consistent and repeatable overall functionality.

During the initial stages of the prototype project, the team disables CM rules for items and request types. This allows revisions to be freely created and actioned.

Just before delivery, the team enables CM rules to ensure that items in the release baseline cannot be further developed without request authorization.

For item types, the team sets these rules:

- **Creation rule:** Disabled so that the team can still create completely new items without requests.
- **New Revision rule:** Set to the initial state so that it is immediately effective. This means that all subsequent revisions of an item require requests.
- **Action rule:** Set to a state just prior to the state specified in the baseline template. Item revisions that existed prior to the rules being enabled are prevented from progressing beyond a certain state without a request, ensuring that they do not get included in any future baseline without this control. This rule would also be used to catch any item revisions that might become unrelated from a request for any reason.

  NOTE Any item revisions that have already progressed beyond the state specified in the Action rule can still be actioned without the need for a request. The team must manage these items outside the scope of the rules specifications.
- **Closure rule:** Set to a quality state in the item's lifecycle so a request can only close when the item reaches this state.

For request types, the team:

- Combines the ANALYSIS and WORK phases.
- Specifies that a child request must reach a quality state before the parent request can close.

Following the delivery of the prototype, the team imposes further controls to prevent any unauthorized changes being made.

Chapter 2

Privileges and Roles

In this Chapter

About Privileges and Roles

The operations you can perform, and the object classes you can perform them on, are determined by the privileges and/or role assignments that have been defined for your user.

The purpose of this chapter is to explain how privileges and roles work, the differences between them, and how you can use them to control which users can perform the various tasks in your applications and processes.

Managing privileges and roles using the Administration Console is described in "Users and Roles" on page 77. There is a complete list of the privileges and their associated rules in Appendix E, "Dimensions CM Privileges" on page 361.

A *privilege* is a function or action that a user or group can perform, such as editing the content of items or managing lifecycles. There are a set of *privilege rules* that you can specify in the Administration Console for each privilege that determines which users can perform that function, and under what conditions. For example, the ability to update a design part can be restricted to the user who originally created the design part or any member of the ADMIN group.

A *role* consists a collection of privileges that can be assigned to a user or group. For example, if you are assigned the role of PRODUCT-MANAGER you can perform all the functions permitted for that role, for example deleting or renaming a product.

The fundamental differences between roles and privileges are:

- You can assign the same role to different users for different design parts or subordinate design parts.
- You can assign different roles to different transitions in the lifecycle of an object type, thus allowing different people to be responsible for the various stages in the approval process within that lifecycle.
- You can control who can perform a specific task more precisely using privileges. For example, you could explicitly allow only a particular user to edit lifecycle definitions.

Managing Privileges

A privilege is either a function that a user can perform on an object class, such as creating a baseline, or it is an administrative task such as managing users and groups. Each privilege has a set of rules that can be set at various levels to suit your organization's ways of working. Unlike roles, privileges cannot be restricted to particular design parts, but generally apply to all objects in a particular class.

You can have a privilege assigned to:

- Your user
- A group to which your user belongs
- A Dimensions CM role to which you have been assigned.

Privileges are classified into two areas:

Administration Privileges, privileges to perform various administrative functions within the Dimensions CM database, for example to assign roles to users, or update the properties of a deployment area. The settings for these privileges apply to all products in the base database. They are subdivided into:

- Process Management
- Area Management
- Other Administration.

Product Level Privileges, privileges to perform operations on classes of Dimensions CM objects within a specified product, for example to promote an item or create a baseline. The settings you make in the Administration Console for these privileges only apply to your current product.

The object classes are:

- Product
- Design Part
- Request
- Item
- Baseline
- Project/stream
- Release

About Privilege Rules

A privilege rule specifies the conditions under which a user has a given privilege. Privilege rules fall into the following categories:

- **General Grant Rules**, privilege rules that do not designate specific users or groups. The privilege applies to any user who satisfies the rule at that time. For example, a user may have the privilege to action an item if it is in their inbox.
- **Explicit Grant/Deny Rules**, privileges that can be specifically granted or denied to one or more users or groups. For example, a member of the group TEAMLEADS can create a new project, or a member of the group QA cannot update the contents of an item. These rules take precedence over other rules that grant the same privilege. For example, a General Grant Rule may allow a user to update the attributes of a project/stream if it is in their inbox. If an explicit deny rule is set for the group QA to prevent them from updating project/stream attributes, then a user belonging to group QA would be prevented from updating the attributes of a project/stream even if it was in their inbox.

  Certain privileges relating to promotion and demotion can be specified for specific projects/streams and deployment stages, and privileges for deployment and rollback can also be specified for specific areas.
- **Other Grant Rules**, privilege rules that specify a role. For example, a user may be allowed to rename a project if they have the role PROJECT_MANAGER for any product, or a user can delete an item if they have the role DEVELOPER on the design part that owns the item.

Example of Privilege Rules

In the example below, the Revise Item Content privilege has the following rules set in the Administration Console:

General Grant Rules

Rule	Enabled?	Description
Object is in the user's inbox or user has current role	✔	The privilege will be granted if the object is in the user's inbox or the user has a role on a state tran
User is the originator of the object	✖	The privilege will be granted if the current user is the originator/creator of the object
User has any role on the initial lifecycle state transition	✔	The privilege will be granted if the user holds any role on the initial normal lifecycle state transition c
User has any role on object lifecycle	✖	The privilege will be granted if the user holds any role on any transition in the whole lifecycle of the
User holds any role on the design part owning the object	✖	The privilege will be granted if the current user has any role on the design part that owns the objec
User holds any role on the product owning the object	✖	The privilege will be granted if the current user has any role on the product owning the object
User holds any role on any product	✖	The privilege will be granted if the current user has any role on any product in the base database
Grant to all users	✖	The privilege will be granted to everyone

Explicit Grant/Deny Rules

User/Group	Grant/Deny
ADMIN	✔

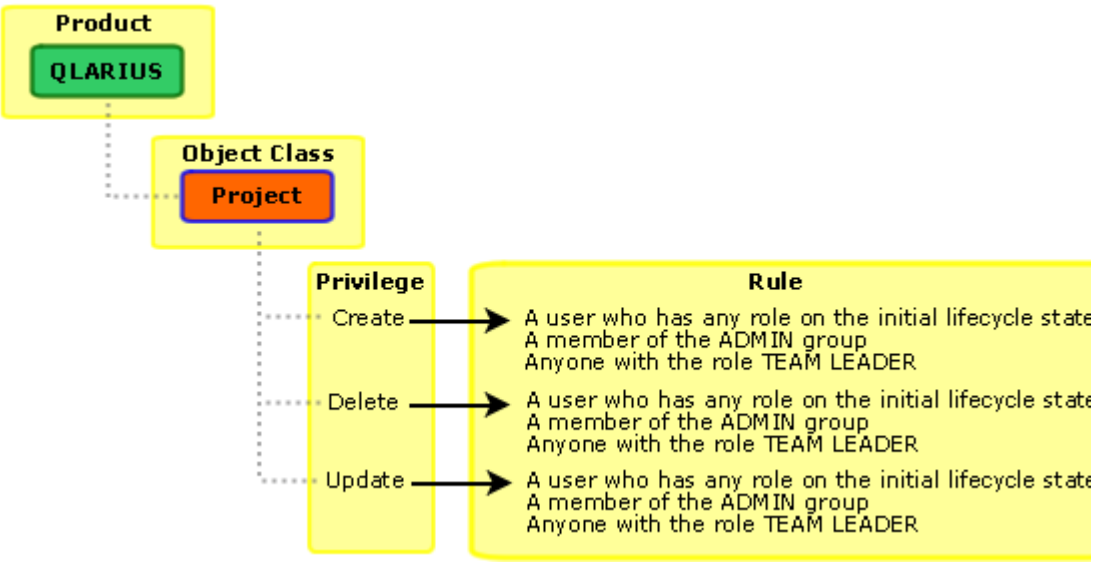
Other Grant Rules

Rule	Role name
User has one or more of the following roles on the design part	No roles have been assigned.
User has one or more of the following roles on the product	DEVELOPER PRODUCT-MANAGER
User has one or more of the following roles on any product	No roles have been assigned.

- General Grant Rules
  - *Object is in the user's inbox or user has current role*. This means that a user with the necessary role to action an item from its current lifecycle state to the next one is able to change its contents.
  - *User has any role on the initial lifecycle state transition*. This means, for example, that a user with the necessary role to action an item of type Document from the Draft state (the first state in its lifecycle) is able to change its contents (even when it is not currently in the Draft state).
- Explicit Grant/Deny Rules
  - ADMIN: A member of the ADMIN user group can change the contents of an item.
- Other Grant Rules
  - *User has one of the following roles on the product*: DEVELOPER, PRODUCT MANAGER. This means, for example, that the user assigned the PRODUCT-MANAGER role for the product owning the item can change its contents.

Privilege Rules Scenario

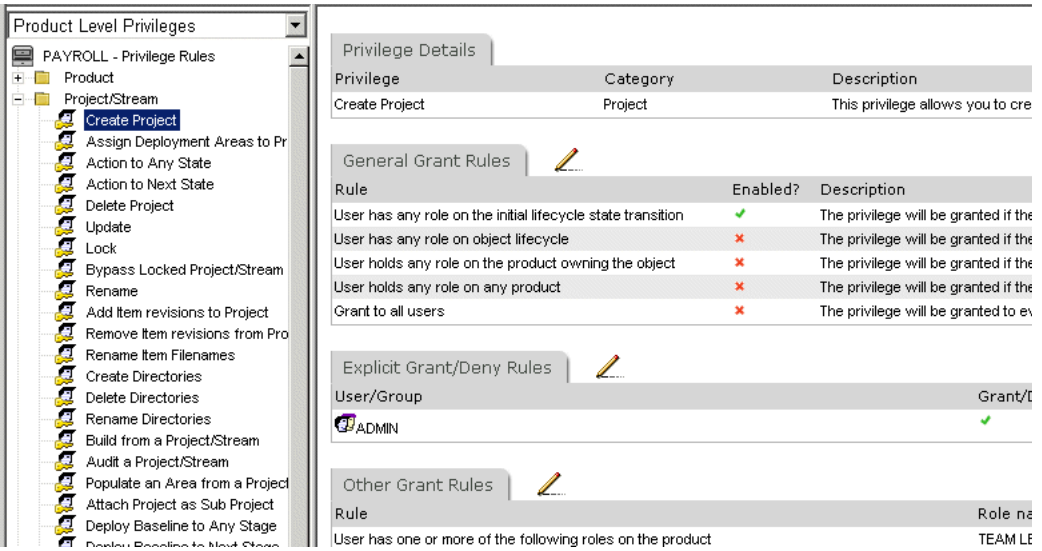
The example below shows how you might set up privileges for creating, updating, and deleting projects:



Scenario

Bill, the Administrator, is a member of the ADMIN group. The ADMIN group have the privilege *Manage Privileges* explicitly assigned, therefore he can assign privileges to other users.

1. He logs into the Administration Console and selects **Users and Roles** | **Privilege Assignments**.
2. He selects **Product Level Privileges** | **Project/Stream** | **Create Project**.



3. He sets the privilege rules:
   - User has any role on the initial lifecycle state transition
   - Explicit Grant/Deny Rules: ADMIN
   - User has one of the following roles on the product: TEAM LEADER

4 He then selects the privileges *Delete Project* and *Update*, and sets the same privilege rules for those privileges.

5 Ted the Team lead, has the role of TEAM LEADER for the product QLARIUS. He logs into the desktop client and creates a new project PROJA.

6 Some months later, development work in PROJA is complete, and it is no longer needed. Ted, however, has left the company.

7 Bill deletes the project PROJA, as the privilege to do this is specifically assigned to the ADMIN group.

Guidelines for Granting Privileges

Privileges, along with lifecycles, roles, and design parts, manage your organization's process control. Tighter process control can be achieved with the use of CM rules.

Some Considerations

When considering the level of process control that you wish to implement you should consider the following guidelines:

- The general grant rules should be sufficient to implement process control for most users.
- Although it is possible to grant or deny privileges to specific users, it is better for administrative and maintenance purposes, to grant the privilege to a group and include the user in that group.

  NOTE The more rules you enforce the more checks will be needed and this may impact your system's performance.
- There may be times when you need to correct or undo an action just taken. For such occurrences you could create a group to perform such semi-administrative tasks, for example, product level item privileges:

  - Move item to another design part
  - Action to any state
  - Revise item content

  For example, a user has actioned an item revision to the next normal state but wishes to return the item revision to the previous state.

The Deliver into Project/Stream Privilege

The ability to deliver item revisions into a project/stream is a key action and the privilege to do so needs careful consideration. Dimensions can be configured, using privileges, to enforce varying degrees of restriction. These range from the very restrictive "Object is in the user's inbox or user has current role", to the unrestrictive, just get it done, "Grant to all users".

The options are:

- Object is in the user's inbox or user has current role
- User is the originator of the object

- User has any role on the initial lifecycle state transition
- User has any role on object lifecycle
- User holds any role on the product owning the object
- User holds any role on any product
- Grant to all users

Example

Qlarius, who have small development teams, have decided that their project/stream can be delivered to by users with any role on the initial lifecycle state transition. To achieve this general grant rules for the Project/Stream privilege Deliver file into Project/Stream have been enabled:

- Object is in the user's inbox or user has current role
- User is the originator of the object
- User has any role on the initial lifecycle state transition

In this example, the "Deliver Files into Project/Stream" privilege is checked against the project/stream into which you are delivering so it checks if:

- The project/stream is in the users inbox or the user has a current role on the project/stream lifecycle, or
- They are the originator of the project/stream, or
- They hold any role on the product owning the project/stream

Also, by default, this privilege is enabled for anyone in the ADMIN group.

Managing Roles

A role is a set of privileges that can be assigned to one or more users or groups that enables them to perform the functions associated with that role. When you install Dimensions CM there is a set of roles already defined with the necessary privileges to enable users to perform the functions for various positions within the organization, such as Developer or Team Leader. For example, having the role of CHANGE-MANAGER gives a user special abilities in relation to requests, such as to view other users' inboxes and action requests to any state in their lifecycle. You can, if you wish, change the rules that determine the privileges assigned to these roles to more closely model your company's processes or you can add new roles. These default roles and their privileges are detailed in "Privilege Reference" on page 364.

You create a new role or change the privileges for an existing role using the Administration Console | Privileges and Roles | Role Definitions function. For details, see "Defining Roles" on page 89.

A role can be assigned to one or more users or groups, and apply to a:

- product
- design part
- design part variant

- project
- lifecycle transition

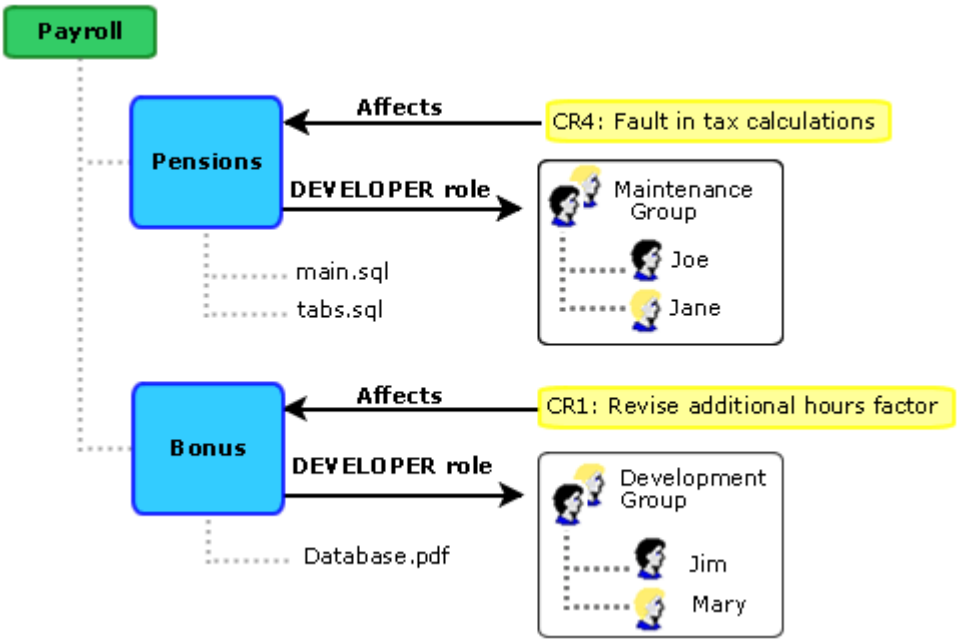
Roles and Design Parts

You assign users to a role on a design part in order to allow those users to work on all objects associated with that design part. In this way, roles promote ownership and responsibility for different functional areas of a product.

Example

Joe and Jane are in the Maintenance group. They need control of the design part called Pensions. Jim and Mary, in the Development group, need control of the Bonus design part. You can assign the role of DEVELOPER to both groups, each on their own design part. If the privilege rule for Update Item Content is set as "user has the DEVELOPER role on the design part", Joe and Jane are able to update the items for Pensions, and Jim and Mary can update items for Bonus.

If, in addition, the lifecycle for request type CR has the DEVELOPER role assigned to the ALLOCATED state, the request will go to Joe and Jane's inboxes for Pensions, and Jim and Mary's inboxes for Bonus.



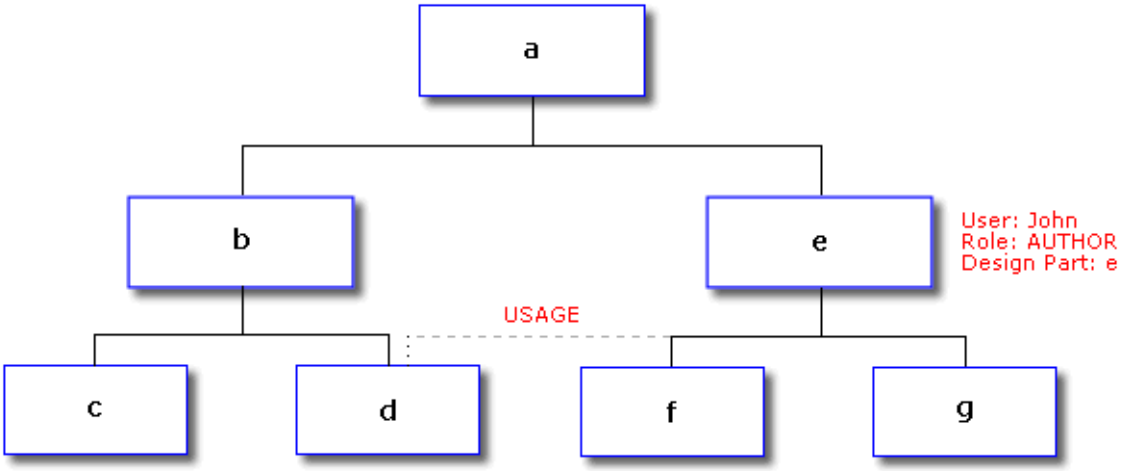
Inheriting Roles

When a user has a role on a design part, that user also inherits the role on all subordinate design parts in the design structure, unless a different user is assigned to the same role on a lower design part. In that case, the latter user assumes responsibility for that design part and any design parts below it.

Example

John holds the AUTHOR role for design part e. In this case, John is responsible for e and its subordinate design parts, f and g. Note that only the hierarchical (breakdown)

structure of the design parts determines role assignments. Therefore, John does not inherit a role on design part d, which is in a usage relationship with design part f. If someone else gets assigned the AUTHOR role for design part f, then John's responsibility would be limited to e and g. John may also create items on these design parts if AUTHOR is specified as the initial role in the lifecycle for the item types.



Roles on Projects and Streams

When you assign a role to a design part without specifying a project/stream, the role assignment will apply to all items in that design part except those belonging to any projects/streams to which another role has been specifically assigned. When you make this assignment, you also have the option of additionally specifying a project/stream, in which case it applies only to item revisions included in that project/stream.

This allows you to assign some users a role for a particular project/stream in the product structure, while assigning other users the same role for any remaining projects/streams.

Example: Assigning Roles to project/streams

Consider the following assignments where <NULL> means that the project/stream field is unspecified.

User	Role	Design Part	project/stream
Dinesh	DEVELOPER	QUOTATION	PROJA
Dawn	DEVELOPER	QUOTATION	<NULL>

Dawn will then have the DEVELOPER role for the QUOTATION design part in every project/stream except PROJA, where Dinesh holds this role.

Scenario

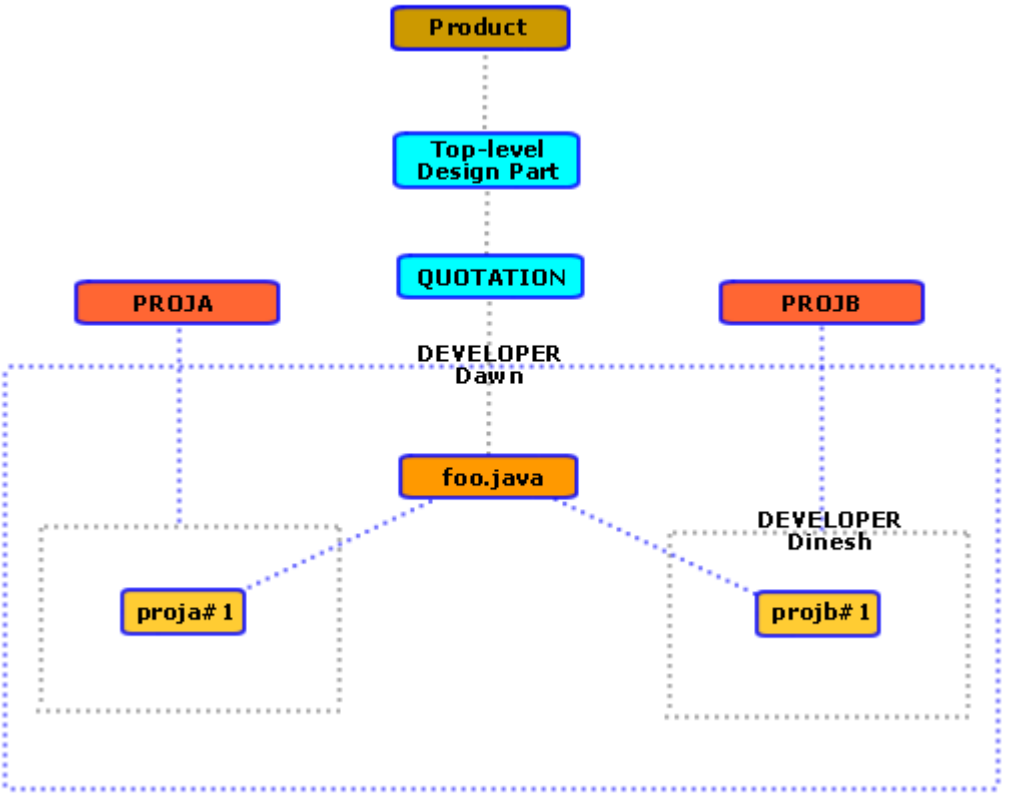
Dawn has been assigned the role DEVELOPER for design part QUOTATION in Product PAYROLL.

Ted the Team Leader creates a new project PROJA in product QLARIUS. Dawn is able to check out file foo.java in PROJA to create a new revision, proja#1

Ted the Team Leader creates another project PROJB in product QLARIUS and has the administrator assign the DEVELOPER role for PROJB to Dinesh.

Dinesh revises (updates) item foo.java in PROJB to create a new revision projb#1

When Dawn tries to check out item foo.java;B#1 she does not have the required role because this revision is belongs to PROJB, and Dinesh has the DEVELOPER role for any items belonging to PROJB.



Assigning Roles to Baselines

When you assign a role to a design part, the role assignment applies to a baseline if its top design part is in that design part segment.

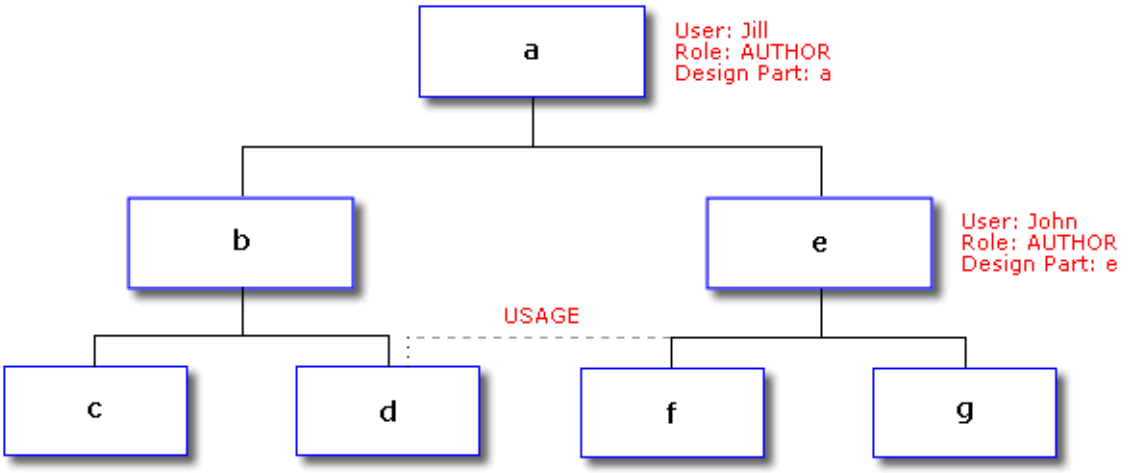
For example, if the role of TEAM LEADER has the privilege to create baselines, and Ted is assigned the role of TEAM LEADER for design part QUOTATION, he can create a new baseline from design part QUOTATION or one of its subordinate design parts.

Assigning Roles to Requests

Requests do not belong to one particular design part or segment, but can have one or more design parts related to them. The way the roles apply to requests is determined by the **Take roles from all affected design parts** option This option is set in the Administration Console *Object Type Definitions* function for requests.

The **Take roles from all affected design parts** option allows you to specify whether a request type should be routed to the role holders on the common ancestor of related design parts, or to the union of the role holders on all related design parts.

For example, consider the design part structure below



The effect of the **Take roles from all affected design parts** option is shown in the table below.

Option set?	Roles apply to...	Example (below)
Yes	All related design parts in a design part segment	John can perform a task on a request if the following is true: ■ The AUTHOR role is assigned to a task in the request's lifecycle. ■ The request is related to part e, f, or g. ■ The request is optionally related to part d in addition to being related to e, f, or g.
No	Nearest common ancestor of all related design parts in a design part segment	John can perform a task on a request if the following is true: ■ The AUTHOR role is assigned to a task in the request's lifecycle. ■ The request is related to part e, f, or g. ■ The request is **not** related to part d. If it is related to part d, then only Jill, who holds the role on the common ancestor, part a, can perform a task on the request.

Roles on Design Part Variants

You can assign a role so that it applies to all variants of a design part or only to a specific design part variant. This allows you to assign some users a role for a particular variant in the product structure, while assigning other users the same role for any remaining variants.

Example: Assigning Roles to Variants

Consider the following assignments where <NULL> means that the variant field is unspecified.

User	Role	Design Part	Variant
Will	DEVELOPER	APPLICATIONS	USA
Sarah	DEVELOPER	APPLICATIONS	<NULL>

In this case, Will is granted the DEVELOPER role for the USA variant of the APPLICATIONS design part. Sarah is granted the same role for all variants of the APPLICATIONS design part, except for USA.

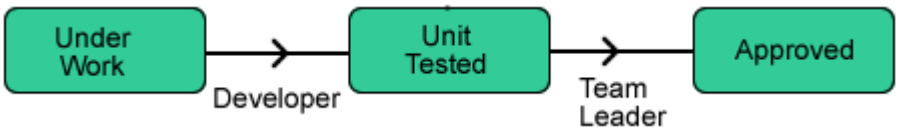
Roles and Lifecycles

You assign roles to transitions in order to:

- Determine which users are able to action an object type between any two specified lifecycle states
- Determine which attributes users can view or update at a specified lifecycle state and which attributes they are required to enter before they can action the object to the next state. This is described in more detail in "Attribute Update Rules" on page 25.

Example of Roles and Lifecycle Transitions

Jane, the Administrator, has defined the normal lifecycle for the item type SRC using the Administration Console. It is set up as shown below.



To action an item of type source from the Under Work state to Unit Tested, you need to have the role of DEVELOPER. To action it from UNIT TESTED to APPROVED, you need to have the role of TEAM LEADER.

Jane has assigned the DEVELOPER role on the PAYROLL product to Bill, and the TEAM LEADER role to Sam

Bill needs to make a change to the source file calcs.c. He checks out the file, makes his changes, and checks the file back in. This creates a new revision with a lifecycle state of UNDER WORK. The item revision appears in his inbox as he is the originator.

After compiling and unit testing the change, he actions the item revision to UNIT TESTED. Since Sam has the TEAM LEADER role, the item appears in his inbox. Sam performs a system test on the Payroll product, which is successful. He therefore actions the item revision for calcs.c to APPROVED.

How do I Assign a Role to a Lifecycle Transition?

You assign roles to lifecycle transitions using the Administration Console | Configuration Object Management | Lifecycles function. For details, see "Managing Transitions" on page 193.

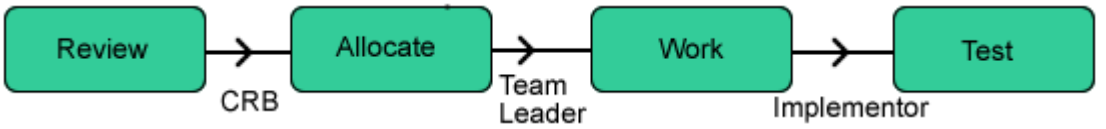
Delegating Roles

If you need to be able to assign or reassign a role for an item or request, you can assign delegation candidates for that role. This means that you will choose one or more of those delegation candidates for a specific item or request at the time that it needs to be worked on.

For example, there are times when allocating a request that you need to decide who it should be forwarded to. You may have a large team of developers and only want to forward the request to one of them. Who you choose may be based on expertise, work load, absences, and so on.

To allow for this, Dimensions CM allows you to create delegate roles and assign delegate candidates to such a role. For example, you could create a delegation role called IMPLEMENTOR, and assign all the appropriate developers to this role. When the team lead actions this request to the Work state, the action wizard requires him to decide who the request should be assigned to, and the request cannot be actioned unless a user has been delegated.

Scenario The lifecycle for request type CR has been defined as below.



The transition from Allocate to Work is assigned the role TEAM LEADER, which for the design part QUOTATION is assigned to Ted. The transition from Work to Test has the role Implementor assigned to it. For the design part QUOTATION, this role is assigned to Dawn, Will, and Dinesh as delegation candidates.

User	Role	Delegation Role on CR Work state
Ted	TEAM LEADER	
Dawn	DEVELOPER	IMPLEMENTOR
Will	DEVELOPER	IMPLEMENTOR
Dinesh	DOC WRITER	IMPLEMENTOR

1. The Change Review Board (CRB) raise a change request against the QUOTATION design part. They action this request to the Allocate state.
2. Ted, the team leader, receives this request in his inbox because he has the Team Leader role on this design part.
3. On evaluating its impact, he decides that a change is required to the item `autoquote.java`. As she is the expert on this part of the application, he decides to allocate this work to Dawn.
4. Therefore, he selects the request in the desktop client and selects Action. In the Action wizard, he selects WORK as the next state.
5. On the Roles page of the Action wizard, he selects the IMPLEMENTOR role, and then selects Dawn from the list of available users for that role.
6. He then completes the Action wizard.

7 The request is actioned to the WORK state, and Dawn, as the delegated user, gets the request in her inbox.

Assigning Delegation Candidates

You assign the required delegate roles to the required design part using the Administration Console | Privileges and Roles | Role Assignment function, and select an **Assignment Type** of **Delegation Candidate** when creating the role assignments. For details, see "Assigning Roles" on page 91. You then assign that role to the required lifecycle transition using the Administration Console | Configuration Object Management | Lifecycles function. For details, see "Managing Transitions" on page 193. This means that the object cannot be actioned through this transition unless a user is delegated.

Inboxes and Roles

Every user has an inbox of requests, items, and baselines. An inbox contains objects that are currently waiting on that user for further action. An object is added to a user's inbox when it is actioned to a state in which the user has a role to action it on to the next state. When the object is added to a user's inbox, the user automatically receives notification in an email message.

Whether an object appears in your inbox, and whether you can action it to the next state can depend on whether the relevant roles are optional or pending as described below.

Assigning Pending and Optional Roles

Each role authorized on a lifecycle transition can be defined as *pending* or *non-pending*. When an object is actioned to a state in which users have a pending role to action it further, the users will automatically receive the object in their inbox and receive email notification. Users with a non-pending role will not receive the object in their inboxes, nor will they be notified via email, but they will still have full authority to work on the object and action it to the next state.

A role on a lifecycle transition can also be defined as *optional*. this means that actioning an object to the transition's from state does not require there to be a user holding that role. If this option is not set, the actioning will be disallowed if there are no users holding that role.

Scenario For example, if the lifecycle for the DOC item type has the following transitions defined in the Administration Console:

From	To	Role	Optional	Pending	User
Draft	Under Review	TESTER QA DEVELOPER	Y N N	Y Y Y	Ted Jane Bill
Under Review	Approved	TEAM LEADER QA	N Y	Y N	Sam Jane

1 Bill, who holds the Developer role, updates an item of type DOC.

2 When he has completed his changes, the new revision is at the initial state of *Draft*.

3 He then actions it to the *Under Review* state.

- Sam, the Team Leader, gets an email notification and the item appears in his inbox because the Pending option is set.
- Jane, who has the QA role does not receive an email, or see the item in her inbox. She can, however, action the item to the *Approved* state just as Sam can do.

In the second example below, the lifecycle for the DOC item type has the following transitions defined:

From	To	Role	Optional	Pending	User
Draft	Under Review	TESTER QA DEVELOPER	Y N N	Y Y Y	None Jane Bill
Under Review	Approved	TEAM LEADER QA	N N	Y N	Sam None

1. Bill updates an item of type DOC to create a new revision (*Draft* state). Although there is no-one with the TESTER role, he is able to do this because the role is optional.
2. After updating the file, he then attempts to action it to the *Under Review* state. Because there is no user holding the QA role, Bill receives a message saying that there is no user holding that role, and is not allowed to complete the action.

Choosing Role Capabilities

Dimensions CM allows you to assign any number of roles to a single user. This is particularly important on smaller development projects, where users may perform many different roles for multiple project/streams and segments of the product structure.

Additionally, Dimensions CM allows multiple users to share responsibility for a role, even within the same product segment and project/stream. When users share a role on a design part, all users receive email notification and inbox entries when appropriate. To help enforce levels of responsibility, Dimensions CM allows you to assign one of the following capabilities to each user:

- primary
- secondary
- leader

Primary and Secondary

A user with the *primary* capability has the main responsibility for that role on the design part or object. You can only assign this role capability to a single user. If you assign the primary capability to a user, then you cannot assign the leader capability (described below) to the same user; primary and leader capabilities are mutually exclusive.

Users with the *secondary* capability act as backup to the user with the primary capability. They have exactly the same privileges as the user with the primary capability.

Leader

An alternative to assigning primary and secondary capabilities is the *leader* capability. This concept mainly applies to requests, where a large group of users may need to

contribute comments without being granted the ability to action the request. This does not apply to items.

If you assign the leader capability to a user, then you cannot assign the primary capability to the same user; leader and primary capabilities are mutually exclusive.

Users with the leader capability can, by default:

- Update the object's attributes
- Relate and unrelate subordinate objects
- Add and edit existing action descriptions
- Action the object to the next lifecycle state

If you assign the leader capability, then the remaining users can only:

- Relate and unrelate subordinate objects
- Add action descriptions

Special Roles

The default roles and their privileges are described in "Default Privileges for Roles" on page 55. Each role has been configured to enable the user to perform the appropriate tasks associated with a typical position within your organization. In a new installation they will have the necessary privileges to allow them to perform the appropriate functions, but you can change these, or add new roles.

This section describes some of the most significant roles.

- **PRODUCT-MANAGER**

  This role allows users to set up and manage the process model for a particular product. The ADMIN group assigns the PRODUCT-MANAGER role to a single user during product creation (Administration Console | Product Definitions).

- **PROJECT-MANAGER and WORKSET-MANAGER**

  This role grants users special permissions in the handling of projects/streams. The Workset Manager/Project Manager can set up and maintain the project/stream directory structure, add or remove items, set the project/stream options that regulate how users work within projects/streams, and lock projects/streams to create baselines.

  Note:

  - Users who create a new stream/project are automatically granted the WORKSET-MANAGER role for that stream/project.
  - Users who create a personal stream, based on another stream or baseline, are only granted the WORKSET-MANAGER role if they have one of these privileges for the parent stream or baseline:
    - *Update Files from Project/Stream*
    - *Update Files from Baseline*
  - All WORKSET-MANAGER role assignments are assigned from a parent stream to a new personal stream on which it is based.

- **PARTS-CONTROLLER**

This role allows the user to manage design parts for the product. This includes creating, updating, and relating design parts. These functions are described in detail in "Design Part Structure" on page 113.

- **DEPLOY-MANAGER**

  This role allows the user to deploy items, requests, and baselines to any stage in the Global Stage Lifecycle for any design part in the product. For details of deployment, see the *Deployment Guide*.

- **CHANGE-MANAGER**

  This role grants users special permissions in the handling of requests. The Change Manager can view other users' inboxes and action requests to any state in their lifecycle, provided that any enabled rules are followed.

- **$ORIGINATOR**

  This role signifies the creator of a specific request, item, or baseline. It is assigned automatically by Dimensions CM to the user when they create the object, and is only applied to that object. Unlike other assignments, this role is not assigned explicitly, and is not applied to other objects within the design part and/or project/stream that have not been created by that user.

Default Privileges for Roles

The topics in this section show the privileges assigned to Dimensions Roles "out-of-the-box". They are organized by Role and the area the privilege affects.

CAUTION! Default Grant Rules for *Update and Deliver* Privileges

When you create a new base database or upgrade an existing installation, the default grant rules for the **Update Files from Project/Stream** and **Deliver Files into Project/Stream** privileges include the **User holds any role on the product owning the object** rule. As a result, there is a security issue where certain users are able to download and upload files from any project in the product including those to which they should not have access. To correct this, you must remove the **User holds any role on the product owning the object rule** from the grant rules for the **Update Files from Project/Stream** and **Deliver Files into Project/Stream** privileges.

Change Manager

Product:

- Override Process Checks

Request:

- Create
- Prime
- Browse
- Delete

- Update Attachments
- Update Request
- Add Action Description
- Edit Action Description
- Delegate
- Action to Any State
- Action to Next State
- Add/Edit Detailed Description
- Move
- Perform Replication Operations
- Relate Request to Request
- Relate Request to Design Part
- Relate Request to Item
- Relate Request to Baseline

Deleter

No default privileges assigned

Deploy-Manager

Baseline

- Deploy to Any Stage
- Deploy to Next Stage

Request

- Deploy to Any Stage
- Deploy to Next Stage

Item

- Deploy to Any Stage
- Deploy to Next Stage

Dev

Item

- Revise Item Content

Developer

No default privileges assigned

Network-Administrator

No default privileges assigned

Parts-Controller

Design Part

- Create
- Delete
- Update
- Relate Design Part to Design Part
- Rename
- Suspend

Privilege Manager

Process Management

- Manage Privileges

Product Manager

Product

- Delete
- Update
- Rename
- Manage Libraries
- Manage Object Types
- Assign Roles to Users And Groups
- Refresh Inboxes for All Users
- Perform Mover Deployments
- Run Reports
- Override Process Checks
- Manage Validsets

- View Other Users' Privileges
- Manage Project/Stream Upload Inclusions/Exclusions
- Relate Requests to Requirement

Project/Stream

- Create Stream
- Delete Stream

Baseline

- Create
- Delete
- Update
- Rename
- Action to Any State
- Action to Next State
- Create Archive
- Delete Archive
- Transfer Baseline In
- Transfer Baseline Out
- Build from a Baseline
- Relate Baseline to Baseline

Release

- Create
- Delete
- Forward to Customer

Item

- Create
- Browse
- Delete
- Update
- Delegate
- Move Item to Another Design Part
- Suspend
- Rename
- Action to Any State

- Action to Next State
- Relate Item to Item
- Relate Item to Design Part
- Revise Item Content
- Archive

Project Access

No default privileges assigned

QA

No default privileges assigned

Reader

No default privileges assigned

Reg

No default privileges assigned

Release-Manager

No default privileges assigned

Reviewer

No default privileges assigned

Team-Leader

Project/Stream

- Create Project
- Rename Item Filenames
- Create Directories
- Delete Directories
- Rename Directories

- Attach Project as Sub Project
- Relate Requests to Project/Stream

Design Part

- Create

Request

- Update Attachments
- Update Request
- Delegate
- Action to Any State
- Action to Next State
- Add/Edit Detailed Description
- Relate Request to Request
- Relate Request to Design Part
- Relate Request to Item

TTDim

Request

- Relate Request to Request
- Relate Request to Baseline

Valid-Set-Manager

Product

- Manage Validsets

Workset-Manager

Project/Stream

- Create Project
- Assign Deployment Areas to Project/Stream
- Action to Any State
- Action to Next State
- Delete Project
- Update
- Lock

- Bypass Locked Project/Stream
- Rename
- Add Item revisions to Project
- Remove Item revisions from Project
- Rename Item Filenames
- Create Directories
- Delete Directories
- Rename Directories
- Build from a Project/Stream
- Audit a Project/Stream
- Populate an Area from a Project/Stream
- Attach Project as Sub Project
- Attach Baseline as Sub Project
- Relate Requests to Project/Stream
- Update Files from Project/Stream
- Deliver Files into Project/Stream
- Change CM Rules for Project/Stream
- Import Request into Project
- Create Stream
- Delete Stream

Baseline

- Update Files from Baseline

Writer

No default privileges assigned

Chapter 3

The Dimensions Administration Console

In this Chapter

About the Dimensions CM Administration Console

The Dimensions® CM Administration Console enables Dimensions CM administrative tasks to be performed from a web-based GUI interface. It provides a framework in which all administrative areas are visible and available with a single click. Only users with the appropriate privileges can update the details in the Administration Console.

Highlights The Administration Console:

- Displays a comprehensive view of Dimensions CM administration while still allowing quick access to common tasks
- Provides a consistent look and feel across the administrative tools
- Allows you to switch between administrative tools without having to log in again

Working in the Administration Console

This section describes how to access the Administration Console and provides a brief overview of the interface.

-

Logging into the Administration Console

Purpose Follow this procedure to access the Administration Console when you are ready to perform administrative tasks on a Dimensions CM base database.

NOTE

- It is not recommended to have more than one browser tab or browser window containing the Administration Console main view. If there are multiple windows or tabs displaying an Administration Console view, actions performed in one window or tab may have an unexpected effect in the other window or tab.
- If Java applets have been disabled by your Dimensions CM administrator all version management operations for local work areas are disabled, including deliver, update, check in, and check-out. Operations on remote work areas will work normally, except for update from stream and merge.
- If using Internet Explorer 6.0 SP2 or 7 on Windows XP SP2:
  - The Pop-up Blocker may interfere with the operation of the Administration Console if your Dimensions CM Web tools server is in the Internet security zone. If this happens, just add the Dimensions CM Web tools server to the Pop-up Blocker Settings as an Allowed site. Alternatively, you could add the Dimensions CM Web tools server to the Local Intranet or Trusted Sites security zones to prevent the Pop-up Blocker from stopping pop-ups from Dimensions CM Web tools.
  - The Pop-up Blocker will by default interfere with the operation of the Administration Console help. To enable Web help to function correctly you must check the **Allow active content to run in files on My Computer** check box in the **Security** section accessed from Tools | Internet Options | Advanced tab.

Using SSO and Smart Card

If your Administration Console has been configured to work using SSO authentication, you will be presented with a **Dimensions CM** log in page the first time that you log in. You will then be presented with an Administration Console login page. Depending on how you log out, you may, or may not, be presented with the **Dimensions CM** log in page the next time you access the Administration Console.

There is also the option to use Smart Card (CAC) authentication. If you are using this feature, you will be prompted to supply your PIN, and the login credentials from the certificate on the specified card will be passed to the application.

If you are not using SSO you will not see the **Dimensions CM** log in page and you will be presented with a slightly different Administration Console login page.

To log in to Dimensions CM using SSO:

1 Start your web browser, and type the URL for the **Dimensions CM** log in page.

The format of the URL is:

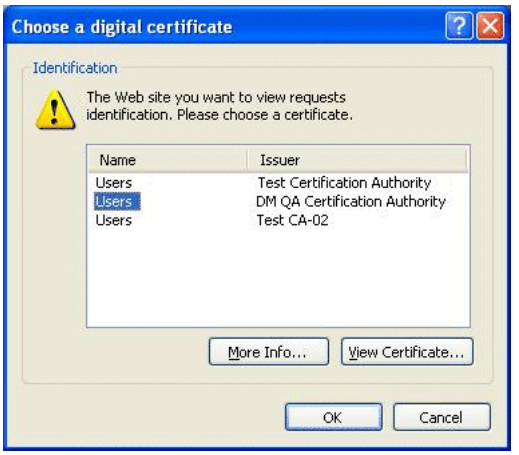
`http://hostname:port/adminconsole/hostname` is the **Dimensions CM** web server and `port` is the web server port number.

2 If you are not already authenticated for SSO, you will be presented with the Dimensions CM authentication page.

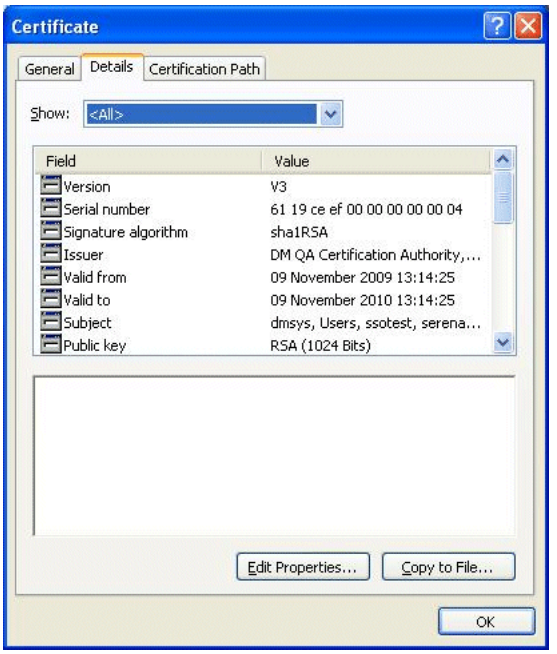
3 Do one of the following:

- If you are logging in using Smart Card Log in:

a Make sure your card is inserted, and click the **Smart Card Log In** button.

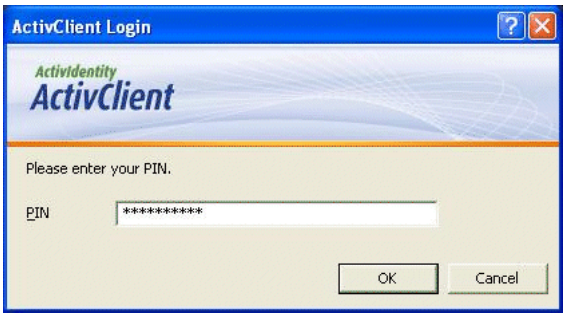


b Select the certificate you want to use. If you want to view the details, click **View Certificate**.



c Click OK in the Choose a digital certificate page.

d If you are authenticating for the first time, you are presented with a Login page asking for your PIN



e Enter your PIN and click OK.

- If you are not using Smart Card log in, Enter your **Username** and **Password**, and click **Log In**.

4 If your login is successful, or you are already authenticated with SSO, "Your user name and password has been validated with SSO" will be displayed.

Click **Log in**.

To log out of Dimensions CM using SSO:

- To partially log out, i.e. be able to log back in without entering your login/password, do one of the following:
  - Close the browser displaying Dimensions web client.
  - Close the browser tab displaying Dimensions web client.
  - Open another Web application, site, etc in a tab within Dimensions web client.

- To fully log out,
  - Click the **Logout** button in Dimensions web client.

    This will require you to sign in to the Dimensions CM page the next time you access the Dimensions web client.

To log in to Dimensions CM without SSO:

1. Open your web browser and enter the Administration Console URL provided by your administrator. The format of the URL is:
   `http://hostname:port/adminconsole/`

   where `hostname` is the machine hosting the Administration Console and `port` is the web server port number.

   Each time you access the Administration Console URL with your browser, you will be prompted to allow the Administration Console applet to run. Click Yes to allow the applet to run.
2. To log in, complete the fields on the Login page. Enter your user ID in the **User Name** field and password in the **User Password** field. These are typically assigned by your administrator.
3. If you want to manually select the server and database:
   - Select **Manual** from the **Connect to** list. The possible settings here are configured in the `web.xml` file for the server installation. See the System Administration Guide for details.
   - Enter the name of the server in the **Server** field.
   - Enter the name of the base database that you want to access in the **DB Name** field.
   - Enter the database connection string in the **DB Connection** field, if this is not already present. This field enables you to connect to the database server for your Dimensions CM database.
4. Click **Login**. The Administration Console opens.

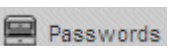
To view information about the Administration Console:

If you want to check the version of the Administration Console that is installed, or other system information, click the **About** link.

Logging into a Remote Node

Purpose Follow this procedure when you want to log in to a remote node or change the password for a remote node.

To log in to a remote node or change the password on a remote host:

1. Log in to Dimensions CM.
2. In the status area, click the Passwords link: 

3. In the **Network Node** field, type the name of the remote node or select it from the list.
4. For **Username**, type your user name for the remote node.

5 For **Password**, type your password for the remote node.

6 If you want to change your password on the remote node:

a Enter the new password in the **New password field.**

b Re-enter the new password in the **Confirm password** field.

NOTE You can only change the password for a Windows remote node.

7 Click the OK button.

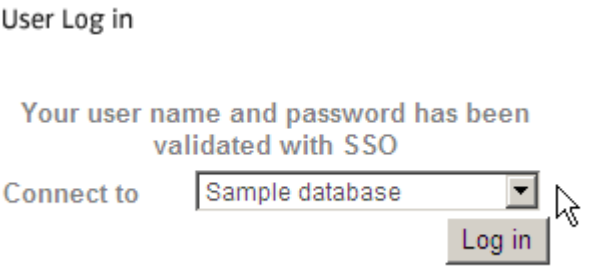
Changing to Another Base Database

If you are using SSO, you can change to another base database without logging out.

1 In the Administration Console main window, click the Change DB link at the top right.

2 On the Login page, if necessary, enter your username and password.

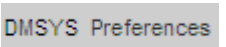
3 Select the database from the list and click the Log in button.



Setting the Preferences for the Administration Console

Purpose Follow this procedure when you want to set preferences for the Administration Console.

To set user preferences:

1 In the status area, click the Preferences link: 

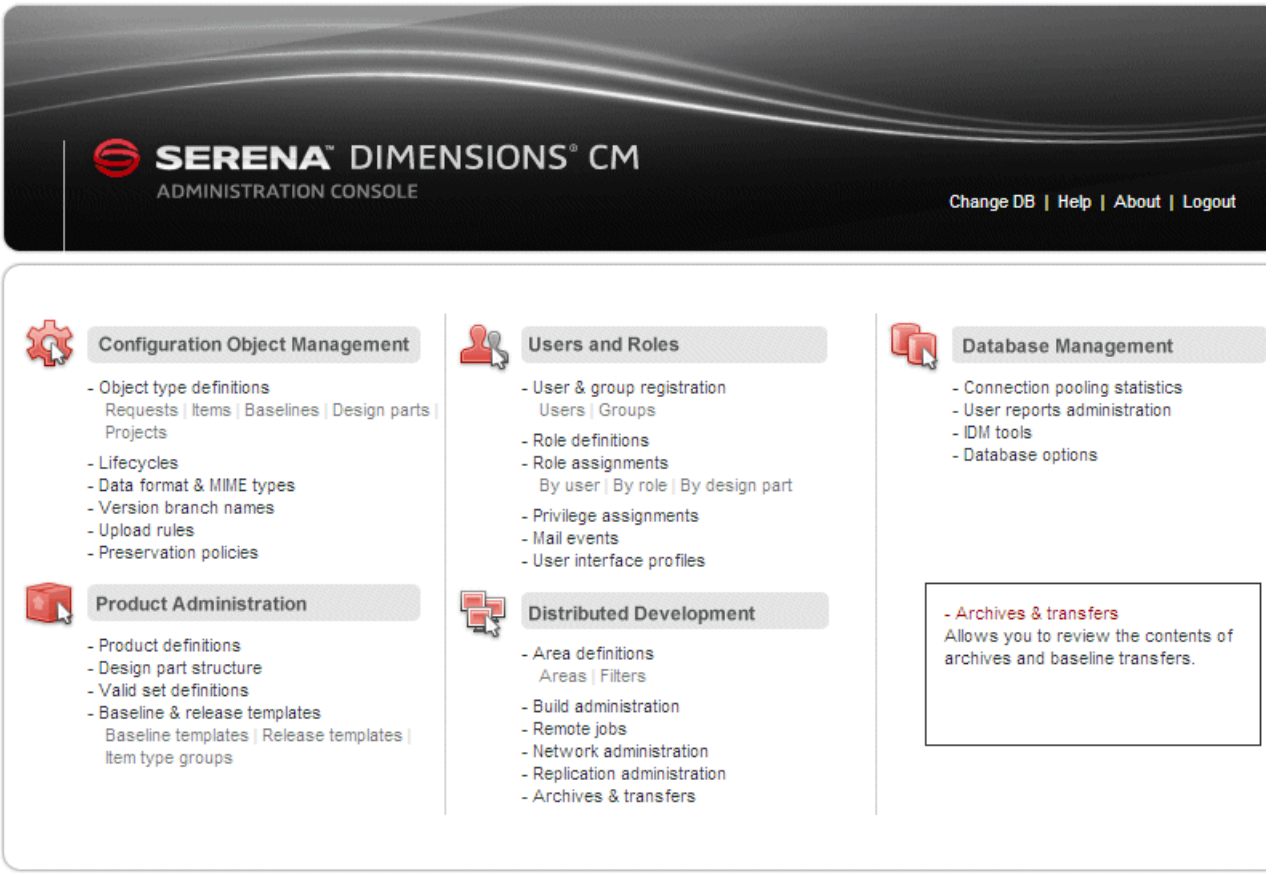
2 In the Set User Preferences dialog box:

- In the **User display format** list, select the label to be displayed for the user name in the status bar and dialog boxes:
  - Select **USERID** if you want the only the ID of the user to be displayed.
  - Select **User Full Name** if you want only the name of the user to be displayed.
  - Select **USERID** (**User Full Name)** if you want the ID followed by the name in brackets to be displayed.
  - Select **User Full Name (USERID)** if you want the name followed by the ID in brackets to be displayed.
- In the **Paging size** field, enter the number of rows that you want to be displayed in each page of a table in the Administration Console

3 Click the **OK** button.

The Administration Console Main Window

The Administration Console main window consists of links to the administrative tools, which are grouped together by overall functionality.



The table below describes each tool available in the Administration Console and refers to the Dimensions CM guide where the tool is documented.

Tool	Description	Refer to the...
Configuration Object Management		
Object Type Definitions	Allows you to add and manage objects types for a particular product.	Process Configuration Guide
Lifecycles	Allows you to add and manage lifecycles.	Process Configuration Guide
Data Format & MIME Types	Allows you to add and manage data formats and MIME types.	Process Configuration Guide
Version Branch Names	Allows you to add and manage version branch names in the base database.	Process Configuration Guide
Upload Rules	Allows you to manage upload rules for Dimensions CM, IDEs, and IDE projects.	Process Configuration Guide

Tool	Description	Refer to the...
Preservation Policies	Allows you to define how different types of build targets are preserved in Dimensions CM.	Process Configuration Guide
Product Administration		
Product Definitions	Allows you to add and manage products in the base database.	Process Configuration Guide
Design Part Structure	Allows you to add and manage design parts within the current product.	Process Configuration Guide
Valid Set Definitions	Allows you to add and manage valid sets and assign them to attributes.	Process Configuration Guide
Baseline & Release Templates	Allows you to add and manage baseline and release templates in the base database.	Process Configuration Guide
Users and Roles		
User Registration	Allows you to add and manage users in the base database.	Process Configuration Guide
Role Definitions	Allows you to add and manage roles in the base database.	Process Configuration Guide
Role Assignments	Allows you to assign roles to users and manage existing role assignments. You can make assignments from the user role, or design part perspective.	Process Configuration Guide
Privilege Assignments	Specifies which users, groups, or roles can perform which operations in Dimensions CM.	Process Configuration Guide
Mail Events	Defines events in Dimensions CM that cause email messages to be sent and specifies which users or groups they are sent to.	Process Configuration Guide
User Interface Profiles	Allows you to select which features in the Dimensions CM client tools are visible to a specified user or group.	Process Configuration Guide

Tool	Description	Refer to the...
Distributed Development		
Area Definitions	Allows you to define file areas on Dimensions CM network nodes. Also allows you to define deployment areas and assign them to a stage in the Global Stage Lifecycle.	Process Configuration Guide
Build Administration	Allows you to add and manage build projects.	Dimensions CM Build Tools User's Guide
Remote Jobs	Allows you to list the remote jobs in the job queue, to view their logs, and to edit and delete jobs in the queue.	System Administration Guide
Network Administration	Allows you to add and manage network nodes and client/server connections.	System Administration Guide
Replication Administration	Allows you to add and manage configurations for replicating items in projects and baselines, and requests, and to review replication logs.	System Administration Guide
Archives & Transfers	Allows you to review the contents of archives and baseline transfers.	System Administration Guide
Database Management		
Connection Pooling Statistics	Allows you to monitor database connection pooling information.	System Administration Guide
User Reports Administration	Allows you to add and manage user report definitions.	Reports Guide
IDM Tools	Allows you to integrate an IDM provider external to Dimensions to provide requests/issues for change management operations.	
Database Options	Allows you to set database options such as whether to allow access to streams and/or projects.	

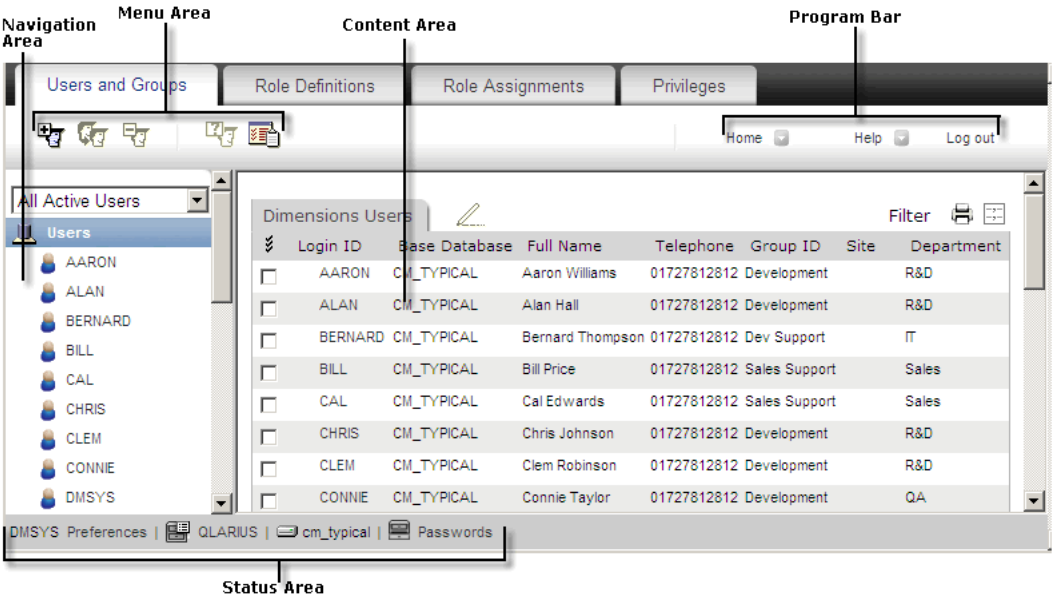
The Specific Function Window

When you access a specific function in the Administration Console, the window consists of:

Menu Area: This contains a set of buttons that allow you to perform various operations on the objects in the content pane.

Content Pane: This contains icons in a tree structure. You can expand the tree and select the nodes beneath it. Selecting an icon either displays a list of the objects that are contained beneath one of these nodes, or the details of an individual Dimensions object in the content pane.

Content Pane: This either contains details of a list of objects in a table or contains more comprehensive details about a specific object.



Status Area: This contains icons and links relating to what you have currently selected and that you can change by clicking the icons or links.

The Administration Console Program Bar

The program bar, located in the upper-right corner of the Administration Console main window, contains the following buttons:

Button	Function
Help	Clicking the Help link launches the online help, which contains procedural and reference information on the Administration Console features. Clicking the list button allows you to select: ■ e-Learning Tutorials. Takes you to a page where there is a selection of on-line tutorials. ■ Customer Support: Takes you to the support site. ■ About this Application: Displays details on the version of Dimensions CM.

Button	Function
Log out	Logs you out of the Administration Console.
Home	Appears if you are in a submenu. ■ Clicking the link returns you to the Administration Console main window. ■ Clicking the list button displays a window containing a list of all the functions in the Administration Console. Click one of the names to take you to that function in a single click.

The Status Area

When you click a tool, the tool's main window contains a status area at the bottom of the window.

It contains the following icons:

Icon	Description
User name and Preferences link	The user ID of the user currently logged in. Click the Preferences link to display the Set User Preferences dialog box.
Product name	The name of the current product. You can click the icon to change the current product. **NOTE** When you log in to the Administration Console under your user, the current product will be the last one you have previously set under that user.
Database	The name of the base database currently accessed.
Passwords Link	Click this link to display the Passwords dialog box to change a user password.

Part 2

Process Configuration Functions

Part 2 Process Configuration Functions contains the following chapters.

Chapter 4

Users and Roles

In this Chapter

About Users and Roles Management

To control access to Dimensions® CM, you define users and groups of users. You also define roles and privileges and assign these to users and/or groups. Dimensions uses this information to ensure that only the users with the authorized roles or privileges can perform specified actions on a particular part of the product.

The management of users and groups in the Dimensions Administration Console comprises the following functions:

- **User and Group Registration:** Allows you to add new user accounts and modify existing ones. See About User and Group Registration on page 80.
- **Role Definitions:** Allow you to add new role definitions and modify existing ones. See About Role Definitions and Assignments on page 88.
- **Role Assignments:** Allows you to assign roles to users and modify existing role assignments. See About Role Definitions and Assignments on page 88.
- **Privileges:** Allows you to grant or deny privileges to users, groups, or roles according to specified rules. See About Privilege Management on page 94.
- **Mail Subscriptions:** Allows you to subscribe users or groups to receive email notifications when certain events occur. See About Mail Notifications on page 100.
- **User Interface Profiles:** Allows you to define and assign profiles to users that determine which features are visible in the client tools. See About User Interface Profiles on page 104.

Invocation Dimensions Administration Console | Users and Roles

NOTE When you access any of the functions above, you can easily switch between them by selecting the Users, Role Definitions, and Role Assignments tabs.

Types of Dimensions User

These types of users exist in Dimensions:

- **Normal:** A user who has normal access to Dimensions. You can create this type of user by adding a new user.
- **Auto Registered (Proxy):** A user who has limited access to Dimensions. This type of user is created when an unregistered user logs on to Dimensions. You can promote a proxy user to a normal user account.

  The ability for an unregistered user to log on to Dimensions is controlled by the Proxy User Login feature, which can be enabled by the Admin group. Proxy Login is enabled via the creation of a new account with the reserved word "*" (asterisk) as the user login name, and is disabled by de-registering this special user. See the *System Administration Guide* for details.
- **Dormant:** An inactive user who has been deleted from Dimensions. You can still view details about a dormant user.

About Groups

A group is a set of users to which you can assign a privilege, thus assigning that privilege to all the users that belong to that group. This provides a manageable way of granting and

denying privileges. For example, you could create a group called DEVELOPMENT, and assign users USER1, USER2, and USER3 to that group. You could then grant to that group the privileges to create, browse, and update items, thus giving these abilities to USER1, USER2, and USER3.

The ADMIN Group

When you first install Dimensions CM, there is a group called ADMIN already defined. This group is set up for administration purposes and has the necessary privileges granted to perform tasks such as creating users and granting privileges to them, and so on.

About Privileges

A privilege is the ability to perform a certain operation on a particular class of object, for example to create a project/stream, or to action an item. It can also be the ability to perform a general administrative function, such as to manage lifecycles or privileges. A privilege can be granted to a user, a group, or a role, or be made to apply generally to any user or under particular rules. For more details see "Managing Privileges" on page 40

About Roles

Roles determine:

- Who can action an object from one particular state to another in its lifecycle. This is done by assigning a role to a lifecycle transition for an object type. For details see "Managing Transitions" on page 193.
- Which users can work on a particular product segment. Roles can be assigned to users at Product or Design Part level, but can also be assigned to a specific design part variant and/or project/stream. When you assign users to a role on a design part for example, this assignment allows those users to work on the objects associated with that design part.

Having been defined, a role can be assigned to a user or a group, thus controlling which users can perform the various operations on the Dimensions objects in particular parts of a product. This distinguishes roles from privileges. Privileges determine which users can perform specified operations on particular object classes within the product as a whole.

For more details about roles, see "Managing Roles" on page 45

Users and Roles Main Window

The Users and Roles main window consists of the following:

- **Menu area:** Displays the program's tabs and toolbars to help you carry out various tasks.
- **Navigation pane:** Allows you to view and select from a list of objects, which includes users, roles, or design parts, depending on the selected tab.
- **Content pane:** Displays details about the selected objects or a summary of all objects.
- **Status area:** Displays log in details. See "The Status Area" on page 73.

For the location of the parts of the Administration Console main window, see "The Specific Function Window" on page 71

The information that displays in the menu, navigation, and content panes varies, depending on which tab is active and what you have selected. For more details see:

- "Users and Groups Tab" on page 80
- "Role Definitions Tab" on page 89
- "Role Assignments Tab" on page 91
- "Privileges Tab" on page 95
- "About User and Group Registration" on page 80

About User and Group Registration

Purpose

User and Group Registration allows you to:

- List, create, edit, and delete Dimensions user accounts for the current base database
- List, create, edit, and delete Dimensions user groups for the current base database
- Promote proxy users to normal user accounts
- Define custom attributes for users

PRIVILEGES Manage Users and Group Definitions

Invocation

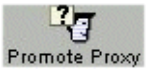
Dimensions Administration Console | Users and Roles | User and Group Registration

Users and Groups Tab

Users and Groups Tab: Menu Area

The Users and Groups tab enables you to create new users and groups of users as well as modify existing users and groups. See the table below for a description of the toolbar buttons that appear on the Users tab. The buttons that appear depend on the selection you have made in the navigation pane.

Button	Function
 or 	Define a new user using the New User dialog box. Define a new group using the New Group dialog box.
 or 	Copy details of the selected user/group to create a new user/ group using the Copy User or Copy Group dialog box.
 or 	Delete the selected user(s) or group(s).

Button	Function
	Promote a proxy (auto registered) user to a normal user using the Promote Proxy dialog box.
	Add, edit, or remove custom attributes from users using the Manage Attributes dialog box.

Users and Groups Tab: Navigation Pane

The navigation pane on the Users and Groups tab lists the users in the base database. In the navigation pane, you can:

- Filter users by type by selecting **All Groups**, **All Active Users**, **Auto Registered**, or **Dormant Users** from the list at the top of the navigation pane.
- View the users by full name by clicking  , or view by user ID by clicking  (not available when **All Groups** is selected).
- Select a user icon  or group icon  to view or modify the its associated details in the content pane.
- Select the top-level icon:  to view a summary of all users or all groups in the content pane.
- Select a user/group section icon. This appears if a large number of users or groups exist and they are separated into sections, so that you can view the users/groups in that section.

Users and Groups Tab: Content Pane

The information displayed in the content pane depends on what you have selected in the navigation pane.

If you select...	Then the content pane displays:
A single user	■ **General:** Section that displays details about the selected user and any user-defined attributes. ■  : Click to modify details and attributes for the user. ■ **Groups:** Section that displays details about the groups to which the selected user belongs, and any user-defined attributes. ■  : Click to modify details and attributes for the user using the Edit User dialog box.
A single group	■ **General:** Section that displays details about the selected group and any user-defined attributes. ■  : Click to modify details and attributes for the group. ■ **Users:** Section that displays details about the users that belong to the selected group and any user-defined attributes. ■  : Click to modify details and attributes for the group using the Edit Group dialog box.
Users node  or Groups node 	■ **Dimensions Users/Groups:** Section that displays a detailed summary of all users in the user section or base database. ■  : Click to modify details and attributes for the selected user or group. ■ Filter : Click to filter the list of users or groups by completing the filter dialog box. ■  : Click to print the summary of users or groups. ■  : Click to save the summary of users or groups as comma-separated values.

Managing Users and Groups

Adding Users

Purpose Follow this procedure when you want to add a new user account to the Dimensions base database. This procedure applies to the creation of normal users only.

PRIVILEGES Manage Users and Group Definitions

NOTE You cannot create a user ID with the same name as a group ID.

To add a normal user:

1 From the User Registration main window, click the New button: 

. The New User dialog box appears.

2 Enter the login ID for the new user in the **ID** field.

3 Enter the email address to be used for notification e-mails sent by Dimensions CM to the user. in the **E-mail** field.

4 Optionally, complete the rest of the fields in the dialog box.

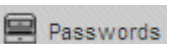
5 Click the Attributes tab to enter values for custom attributes for this user.

6 If you want the dialog box to remain open after adding the new user, select the **Keep open** check box.

7 Click OK to add the new user.

To set a user password:

NOTE You can only change the password for a Windows node.

1 In the status area, click the Passwords link: 

2 For **Username**, type the Windows user name for the user on the Dimensions CM server.

3 For **Password**, type your password.

4 Enter the new password in the **New password field.**

5 Re-enter the new password in the **Confirm password** field.

To reactivate a dormant user

Follow the instructions for **To add a normal user** above and enter the user ID of the dormant user in the **ID** field.

Promoting Proxy Users

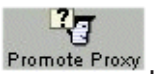
Purpose Follow this procedure when you want to promote a proxy (auto registered) user to a normal user. Promoting a proxy user grants that user greater access to Dimensions functionality, including the ability to perform operations on item revisions.

NOTE When you promote a proxy user, Dimensions CM does not check that mandatory attributes are present. You will need to make sure that such attributes have been entered

PRIVILEGES Manage Users and Group Definitions

To promote a proxy user:

1 From the User Registration main window, select the proxy user that you want to promote.

2 Click the Promote Proxy button: 

.

The Promote Proxy dialog box appears.

3 Click OK to promote the user.

Modifying Users

Purpose Follow this procedure when you want to:

- Edit the information associated with a user account.
- Copy a user so that you can reuse all or most of the user's details for a new user account, including role assignments.

  NOTE Only the role assignments for the current product are copied to the new user. Role assignments for other products are not copied.
- Delete a user account when the user no longer needs access to Dimensions. This frees up the resources allocated to that user. The deleted user becomes "dormant", which means the user account can still be displayed when you filter by Dormant Users in the navigation pane, though the user is inactive.

  NOTE To reactivate a dormant user, add the user using the same ID and details in the New User dialog box.

PRIVILEGES Manage Users and Group Definitions

To edit, copy, or delete a user:

1 From the User Registration main window, select the user that you want to edit, copy, or delete. You can select multiple users in the content pane if you want to edit or delete a group of users.

2 Do one of the following:

If you want to...	Then...
Edit a user	**1** Click the Edit button:  . **2** In the Edit User dialog box, change the values as necessary. **3** If you selected multiple users, you can enter a value in any **<Leave Unchanged>** field to apply it across users.
Copy a user	**1** Click the Copy button:  . **2** In the Copy User dialog box, complete the **ID** field. **3** Change any of the values that have been copied from the selected user as necessary.
Delete a user	**1** Click the Delete button:  . **2** Confirm that you want to delete the user(s) by clicking Yes.

Assigning Attributes to Users

Purpose Follow this procedure when you want to define a custom attribute and make it available to all user accounts in the $GENERIC product.

PRIVILEGES Manage Users and Group Definitions

Constraints

- You can only define single-field, single-value attributes for users.
- To assign a valid set to a user attribute, the valid set must be defined in the $GENERIC product.

To assign an attribute:

1 From the Users Registration main window, click the Manage Attributes button: 

.

2 In the Manage Attributes dialog box, click the Assign Attribute button:  . The Assign Attribute dialog box appears.

3 Enter a name for the new attribute in the **Name** field.

4 If necessary, change the default value for the maximum character length allowed in the **Max Length** field.

5 Select the format for the attribute's value from the **Data Type** list.

6 Enter a field label for the attribute in the **User Prompt** field.

7 If necessary, change the default values for the display length and height of the attribute in the **Display Length** and **Display Height** fields.

8 Optionally, complete the rest of the fields in the dialog box.

9 Click OK to add the new attribute.

NOTE

Editing or Removing Attributes

Purpose Follow this procedure when you want to edit an attribute or remove it from the global user account.

If you remove an attribute that is not in use by another object type, it will be deleted from the database.

PRIVILEGES Manage Users and Group Definitions

To edit or remove an attribute:

1 From the User Registration main window, click the Manage Attributes button:

The Manage Attributes dialog box appears.

2 In the Attributes for User section, select the attribute you want to edit or remove. You can select multiple attributes to remove a group of attributes.

3 Do one of the following:

- Click the Edit Attribute button:  . In the Edit Attribute dialog box, change the values as necessary.
- Click the De-Assign Attribute button:  . Confirm that you want to remove the attribute(s) by clicking Yes.

Filtering Users or Groups

Purpose Follow this procedure when you want to restrict the list of users or groups in the navigation pane or content pane to certain criteria. You can filter by user type or by user details, such as Department or Group ID. You can filter groups by Group Name or Description.

To filter by user type:

1 From the Users and Groups tab, click the drop-down list in the navigation pane.

2 Select the user type you want to filter by:

- **All Active Users:** Lists all users who are activated in the base database.
- **Auto Registered** (proxy): Lists all users who have been automatically registered in the base database.
- **Dormant Users:** Lists all users who have been deleted from the base database.

The list of users in the navigation pane adjusts to the filter.

To filter by user details:

1 From the Users and Groups tab, select **All Active Users** from the drop-down list in the navigation pane.

2 Click the top-level icon:  in the navigation pane. A summary of all users appears in the content pane.

3 In the content pane, click the Filter link:  .

4 Set up your filtering criteria in the User Filter dialog box.

5 Click OK.

To view all users again, open the User Filter dialog box and click OK without entering any values.

To filter user groups:

1 From the Users and Groups tab, select **All Groups** from the drop-down list in the navigation pane.

2 Click the top-level icon:  in the navigation pane. A summary of all groups appears in the content pane.

3 In the content pane, click the Filter link:  .

4 Set up your filtering criteria in the Group Filter dialog box.

5 Click OK.

Creating Groups

Purpose Follow this procedure when you want to add a new group of users to the Dimensions base database.

PRIVILEGES Manage Users and Group Definitions

NOTE You cannot create a group ID with the same name as a user ID.

To add a group:

1 In the Users and Groups tab, select *All Groups* from the drop-down list in the navigation pane.

2 Click the New button: 

. The New Group dialog box appears.

3 Enter the Name for the new group in the **Group Name** field.

4 Enter a description for the group in the **Description** field.

5 Include the users you want to belong to the group:

- To add a user, select the name in the **Available Users** list, and click the **>>** link to move it to the **Assigned Users** list
- To remove a user, select the name in the **Assigned Users** list, and click the **<<** link.

6 Click OK to create the new group.

To create a group by copying an existing group:

1 In the Users and Groups tab, select *All Groups* from the drop-down list in the navigation pane.

2 Select the group you want to copy in the navigation pane and click the Copy button: 

. The Copy Group dialog box appears.

3 Enter the Name for the new group in the **Group Name** field.

4 Enter a description for the group in the **Description** field.

5 To edit the list of users that will belong to the group:

- To add a user, select the name in the **Available Users** list, and click the **>>** link to move it to the **Assigned Users** list
- To remove a user, select the name in the **Assigned Users** list, and click the **<<** link.

6 Click OK to create the new group.

Modifying Groups

Purpose Follow this procedure when you want to edit the description or the list of users in a group, or to delete a group.

PRIVILEGES Manage Users and Group Definitions

To modify a group:

1 In the Users and Groups tab, select *All Groups* from the drop-down list in the navigation pane.

2 Select the group in the navigation pane and Click the Edit button:  . The Edit Group dialog box appears.

3 If necessary, Edit the description for the group in the **Description** field.

4 To edit the list of users that belong to the group:

- To add a user, select the name in the **Available Users** list, and click the **>>** link to move it to the **Assigned Users** list
- To remove a user, select the name in the **Assigned Users** list, and click the **<<** link.

5 Click OK to commit the changes.

To delete a group:

1 In the Users and Groups tab, select *All Groups* from the drop-down list in the navigation pane.

2 Select the group in the navigation pane and Click the Delete button: 

. The Delete dialog box appears.

3 Click Yes to delete the group.

About Role Definitions and Assignments

Purpose

Role Definitions and Role Assignments allow you to:

- List, create, edit, and delete roles in the base database.
- Assign roles to users from the user, role, or design part view.

Constraints

- Role definitions are not constrained to a specific product but can be used throughout the base database.
- Each role assignment must consist of an existing user, role, and design part.

Invocation

- Dimensions Administration Console | Users and Roles | Role Definitions
- Dimensions Administration Console | Users and Roles | Role Assignments

About Role Capabilities

When more than one user has the same role, you can assign a role capability to determine different levels of responsibility within those users. The role capabilities that can be assigned to the users within the role are Leader, Primary, or Secondary, described below:

- **Leader**: It is sometimes useful to have more than one user with a particular role with respect to an item-spec or request e.g. so that they can add comments (called Action Descriptions in requests). However, it may also be appropriate to restrict the number of users in this group who can actually action the object to the next stage in its lifecycle. The way to implement this is via the Leader function. When the Leader capability is defined in a group of users who have the same role for a given object, only the Leader can update the associated attributes and action on the object. All other users with that role may add only Action Descriptions or user comments. If the Leader role capability is assigned to a user, then the Primary role capability (described below) cannot be assigned to the same user i.e. Leader role and Primary role capabilities are mutually exclusive.
- **Primary**: The Primary capability for a role in the lifecycle of an object is assigned to the user regarded in the project as having the main responsibility for that role on the object. There cannot be more than one Primary user defined for a role (as applicable

to any particular design part or segment of the product structure). If the Primary role function is assigned to a user, then Leader role capability (described above) cannot be assigned to the same user.

- **Secondary**: Secondary users are intended to act as deputies for the Primary. They have exactly the same privileges as the Primary: they can add action comments and also, unless the Leader capability has been assigned, update the object's attributes and action it.

Defining Roles

Role Definitions Tab

The Role Definitions tab enables you to define new roles in the base database as well as modify existing roles.

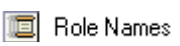
Role Definitions Tab: Menu Area

See the table below for a description of the toolbar buttons that appear on the Role Definitions tab.

Button	Function
	Add a new role using the New Role Definition dialog box.
	Copy details of the selected role to create a new role by using the Copy Role Definition dialog box.
	Delete the selected role(s).

Role Definitions Tab: Navigation Pane

The navigation pane on the Role Definitions tab lists the roles in the base database. In the navigation pane, you can:

- Select a role:  to view or modify its associated details in the content pane.
- Select the top-level icon:  to view a summary of all roles in the content pane.

Role Definitions Tab: Content Pane

The information displayed in the content pane depends on what you have selected in the navigation pane.

If you select...	Then the content pane displays:
A single role	■ **General:** Section that displays details about the selected role. ■ : Click to modify details for the role using the Edit Role Definition dialog box.
Role Names	■ **Role Definitions:** Section that displays a detailed summary of all users in the user section or base database. ■ : Click to modify details for the selected role(s) using the Edit Role Definition dialog box. ■ : Click to print the summary of roles. ■ : Click to save the summary of roles as comma-separated values.

Adding Roles

Purpose Follow this procedure when you want to create a new role in the base database. You can then use this role to limit access to Dimensions by assigning the role to individual users for a specific product or design part.

PRIVILEGES Manage Role Definitions

To add a new role:

1 From the Role Definitions main window, click the New button: 

. The New Role Definition dialog box appears.

2 Enter a name for the new role in the **ID** field.

3 Enter a description for the new role in the **Description** field.

4 Select the **Keep open** check box if you want the dialog box to remain open after adding the role.

5 Click OK to add the new role.

Modifying Roles

Purpose Follow this procedure to edit, copy, or delete an existing role.

PRIVILEGES Manage Role Definitions

Constraints
- You cannot delete a role while it is specified within any lifecycle.
- When you remove a Dimensions user or role from a product, the inbox for that user does not change. A relevant new user role will be able to carry out those actions.
- You cannot copy or edit the ID of an existing role.

- When you copy a role, the role assignments of the existing role are also copied to the new role.

To edit, copy, or delete a role:

1 From the Role Definitions main window, select the role that you want to edit, copy, or delete. You can select multiple roles from the content pane if you want to edit or delete a group of roles.

2 Do one of the following:

If you want to...	Then...
Edit a role	**1** Click the Edit button:  . **2** In the Edit Role Definition dialog box, change the value of the **Description** field.
Copy a role	**1** Click the Copy button:  . **2** In the Copy Role Definition dialog box, enter the name of the new role in the **ID** field. **3** Change the value of the **Description** field that was copied from the selected role as necessary.
Delete a role	**1** Click the Delete button:  . **2** Confirm that you want to delete the role(s) by clicking Yes.

Assigning Roles

Role Assignments Tab

The Role Assignments tab enables you to define new role assignments as well as delete existing role assignments.

Role Assignments Tab: Menu Area

See the table below for a description of the toolbar buttons that appear on the Role Assignments tab.

Button	Function
	Switch to the User view to define role assignments by user.
	Switch to the Roles view to define role assignments by role.

Button	Function
	Switch to the Design Part view to define role assignments by design part.
	Add a role assignment without selecting any objects by using the New Role Assignment dialog box.

Role Assignments Tab: Navigation Pane

The navigation pane on the Role Assignments tab lists the users or roles in the base database, or the design parts in the current product. In the navigation pane, you can:

- Switch between users, roles, and design parts by clicking the associated buttons in the toolbar.
- Select one or more users: , design parts: , or roles: , to view the associated role assignments in the content pane.
- Select the top-level icon: Users , Product, or Role Names to view a summary of all role assignments in the content pane.

Role Assignments Tab: Content Pane

The information displayed in the content pane depends on what you have selected in the navigation pane.

If you select...	Then the content pane displays:
One or more users, design parts, or roles; or these top-level icons: Users Role Names Product	■ **Role Assignments for:** Section that displays a detailed summary of assignments for the selected object(s). ■ : Click to add a new assignment for the selected object(s) using the New Role Assignment dialog box. ■ : Click to delete the selected assignment(s). ■ Filter : Click to filter the list of assignments using the Assignment Filter dialog box. ■ : Click to print the summary of assignments. ■ : Click to save the summary of assignments as comma-separated values.

Assigning Roles

Purpose Follow this procedure when you want to assign a role to a user or group for a particular product or design part. You can make an assignment based on the selected role, user/group, or design part. Alternatively, you can make an assignment without selecting any objects first.

PRIVILEGES Manage Role Definitions

NOTE When you assign or unassign a role for a group, this is merely equivalent to assigning/unassigning that role to each individual user that currently belongs to the group at that time.

NOTE When certain privilege or role assignments are updated in the Administration Console, the changes may not take effect during existing client sessions because of privilege caching. In this case, the user will need to restart the client session for the new privilege settings to apply. This behavior depends on variables that can be set in the `dm.cfg` file. For further details, see the System Administration Guide.

NOTE If you specify a project/stream when assigning a role, this will apply to requests and baselines, and not just items as was the case in earlier releases of Dimensions. It is therefore recommended that roles assigned for managing requests or baselines should not normally be restricted to specific project/streams.

To assign a role to a user or group:

1 From the Role Assignments main window, select the view from which you want to make the assignment by clicking:

-  for the User view
- 

for the Design Part view
-  for the Roles view.

2 If you selected the Design Part view, select the product in which you want to make the role assignment from the list in the navigation pane.

3 Do one of the following:

- Select one or more users/groups, design parts, or roles in the navigation pane, then click the New button:  in the content pane.
- Without making any selections, click the Add Assignment button in the toolbar:

The New Role Assignment dialog box appears.

4 Select a user or group from the **User/group** field, unless you already selected one or more users or groups in step 2.

5 Select a role from the **Project Role** field, unless you already selected one or more roles in step 2.

6 Select a design part from the **Design Part** field, unless you already selected one or more design parts in step 2.

7 If you want to require individual users to be delegated the role for specific items or requests, select *Delegation Candidate* for the **Assignment Type**. This means that the object cannot be actioned to the from state for a transition if a user has not been delegated the required role for that transition.

8 Select a level of capability from the **Role Capability** field.

9 Optionally, complete the rest of the fields in the dialog box.

10 Select the **Keep open** check box if you want the dialog box to remain open after adding the role assignment.

11 Click Yes to add the role assignment.

Deleting Assignments

Purpose Follow this procedure to delete one or more role assignments.

PRIVILEGES Manage Role Definitions

To delete a role assignment:

1. From the Role Assignments main window, select the user/group(s), role(s), or design part(s) from which you want to delete the role assignment(s).
2. In the content pane, select the role assignments that you want to delete.
3. Click the Delete button:  in the content pane. Confirm that you want to delete the role assignment(s) by clicking Yes.

Filtering Assignments

Purpose Follow this procedure when you want to restrict the list of role assignments in the navigation pane or content pane to certain criteria. In the User view, you can filter by user type (active, auto registered, or dormant). In the Design Part view, you can filter by products in the base database. In any view, you can filter by role assignment details, such as Project or Capability.

To filter by user type or product:

From the Role Assignments main window, do one of the following:

- In the User view, click the drop-down list in the navigation pane and select the user type that you want to filter by.
- In the Design Part view, click the drop-down list in the navigation pane and select the product that you want to filter by.

The list of users or design parts in the navigation pane adjusts to the filter.

To filter by role assignment details:

1. From the Role Assignments main window, select the objects associated with the role assignments you want to filter.
2. In the content pane, click the Filter link:  .
3. Set up your filtering criteria in the Assignment Filter dialog box.
4. Click OK.

About Privilege Management

Purpose Managing privileges allows you to:

- Grant or deny a privilege to a user or group.
- Enable or disable a general privilege rule for an object class or administrative function
- Enable or disable a privilege rule for a role.

Invocation
- Dimensions Administration Console | Users and Groups | Privilege Rules

Privilege assignments are defined for the current product in which you are working. When you create a new product based on an existing product, the privilege assignments are copied to the new product. You can also set up privilege assignments by setting your current product to the $GENERIC product. Doing this enables you to use the $GENERIC product as a template for privilege assignments when creating a new product.

Privileges Tab

Privileges Tab: Menu Area

Use the Privilege Rules tab to enable or disable the privilege rules for each class of objects and assign those rules to users, groups or roles. See the table below for a description of the toolbar buttons that appear on the Privilege Rules tab.

Button	Function
	Switch to the User view to define privileges by users/groups.
	Switch to the Roles view to define privileges by role.
	Switch to the Privileges view to define privileges either by user or by role.

Privileges Tab: Navigation Pane

The navigation pane on the Privilege Rules tab lists the privileges in the base database. What appears in the navigation pane depends on which toolbar button you select in the menu area.

If you have clicked the Privileges button:  you can:

- Choose which category of privileges you want to access by selecting **Product Level Privileges**, or **Administration Privileges from** the list at the top of the navigation pane.
- Expand a second-level icon:  to view the privileges for an object class or administrative function in the navigation tree.
- Select a privilege icon:  to view or modify the privilege rules for that privilege in the content pane.

If you have clicked the Users button:  you can:

- Choose how you want to access the privileges by selecting **All Groups**, **All Active Users**, **Auto Registered**, or **Dormant Users** from the list at the top of the navigation pane.
- Select a user icon  or group icon  to view or modify its associated privilege rule assignments in the content pane.

If you have clicked the Roles button:  you can:

- Select a role: to view or modify its associated privilege rule assignments in the content pane.

Privileges Tab: Content Pane

The information displayed in the content pane depends on what you have selected in the navigation pane.

If you select...	Then the content pane displays:
A single user: , group: , or role:	■ A number of sections that display the current assignments for each object class and administration area for the selected user, group, or role. ■ : Click to modify details for the assigned privilege rules.
A Privilege icon:	■ **Privilege details:** Section that displays the name, category and description of the privilege ■ **General Grant Rules:** Section that displays the rules that apply at a general level and are not applicable to specific users or groups. ■ Click to grant this privilege rule. ■ **Explicit Grant/Deny Rules**: Section displaying the rules that apply at a specific user/group level for this privilege ■ Click to grant this privilege rule to specific users or groups. ■ Click to remove the selected privilege rule for the user or group. ■ **Other Grant Rules:** Section that displays the rules for this privilege that apply to roles. ■ : Click to remove an assignment for this privilege rule.

Managing Privileges

Granting or Denying Explicit Privileges to Users or Groups

Purpose Follow this procedure when you want to grant or deny a privileges to users or groups, or to a role.

PRIVILEGES Manage Privileges

NOTE Administration privileges apply to all products in the base database. Product level privileges only apply to your currently selected product.

NOTE When certain privilege or role assignments are updated in the Administration Console, the changes may not take effect during existing client sessions because of privilege caching. In this case, the user will need to restart the client session for the new privilege settings to apply. This behavior depends on variables that can be set in the `dm.cfg` file. For further details, see the System Administration Guide.

To grant or deny Privileges to Users or Groups

1 Do one of the following:

To grant or deny a privilege to one or more users or groups:

a On the Privileges tab, click the Privileges button:  on the toolbar.

b In the navigation pane, select the category to which the privilege belongs from the list, **Administration Privileges**, or **Product Level Privileges**.

c Expand the folder:  for the subcategory to which the privilege belongs in the navigation tree and select the privilege.

d In the **Explicit Grant/Deny Rules** section of the content pane, click the  button and choose an option: **Grant User**, **Deny User**, **Grant Group**, **Deny Group**.

To grant or deny a user or group one or more privileges:

a On the Privileges tab click the User button:  .

b In the navigation pane, select the category of users to which you want to grant or deny privileges from the list, **All Active Users**, **All Groups**, **Auto Registered**, or **Dormant Users**.

c Select the user or group in the navigation pane.

d In the content pane, locate the section corresponding to the object class or functional area for which you want to grant or deny privileges.

e Click the  button in that area and choose an option: **Grant** or **Deny.**

2 Depending on the dialog box displayed, move the users or groups you want to be granted or denied into the **Granted Users, Granted Groups, Denied Users, or Denied Groups** list:

- To add a user/group, select the name in the **Available Users/Groups to grant/deny** list, and click the **>>** link to move it to the **Granted/Denied Users/Groups** list
- If you do not want to include a user/group, select the name in the **Granted/Denied Users/Groups** list, and click the **<<** link.

3 For certain privileges related to deployment, options are displayed that enable you to grant or deny the privilege for a specific project or stream, deployment stage, or area. For details, see "Granting or Denying Explicit Privileges for Deployment Functions" on page 98.

4 Click OK to confirm the changes. A confirmation dialog box is displayed.

5 Click Yes to commit the changes.

To de-assign an explicit privilege rule for a privilege:

1 On the Privileges tab, click the Privileges button:  on the toolbar.

2 In the navigation pane, select the category to which the privilege belongs from the list, **Administration Privileges**, or **Product Level Privileges**.

3 Expand the folder: for the subcategory to which the privilege belongs in the navigation tree and select the privilege.

4 In the **Explicit Grant/Deny Rules** section of the content pane, select the User/Group and click the button.

5 Click OK in the De-assign User/Group from Privilege Rule dialog box.I.

Granting or Denying Explicit Privileges for Deployment Functions

Purpose Follow this procedure when you want to grant or deny explicit privileges to users or groups, for deployment features. Certain privileges relating to promotion and demotion can be specified for specific projects/streams and deployment stages, and privileges for deployment and rollback can also be specified for specific areas. When this is the case these extra fields will be displayed in the Grant/Deny Users/Groups dialog boxes.

For details of these functions see the Deployment Guide.

PRIVILEGES Manage Privileges

NOTE When certain privilege or role assignments are updated in the Administration Console, the changes may not take effect during existing client sessions because of privilege caching. In this case, the user will need to restart the client session for the new privilege settings to apply. This behavior depends on variables that can be set in the `dm.cfg` file. For further details, see the System Administration Guide.

To grant or deny Privileges for Deployment for specific projects/streams/deployment stages

1 On the Privileges tab, click the Privileges button: on the toolbar.

2 In the navigation pane, select the category to which the privilege belongs from the list, **Administration Privileges**, or **Product Level Privileges**.

3 Expand the folder: for the subcategory to which the privilege belongs in the navigation tree and select the privilege.

4 In the **Explicit Grant/Deny Rules** section of the content pane, click the button and choose an option: **Grant User**, **Deny User**, **Grant Group**, **Deny Group**.

5 For **Project/stream**, if you want the assignment to only apply to a specific project or stream, click the browse button and select a project or stream.

6 For **Stage**, if you want the assignment to only apply when the selected project/stream is at a specific deployment stage, select the stage from the list.

7 If the privilege relates to deployment or rollback, for **Deployment Area(s)** select the areas to which the privilege applies from the list.

8 Depending on the dialog box displayed, move the users or groups you want to be granted or denied into the **Granted Users, Granted Groups, Denied Users, or Denied Groups** list:

- To add a user/group, select the name in the **Available Users/Groups to grant/deny** list, and click the **>>** link to move it to the **Granted/Denied Users/Groups** list
- If you do not want to add a user/group, select the name in the **Granted/Denied Users/Groups** list, and click the **<<** link.

9 Click OK to confirm the changes. A confirmation dialog box is displayed.

10 Click Yes to commit the changes.

To de-assign an explicit privilege rule for a Deployment privilege:

1 On the Privileges tab, click the Privileges button:  on the toolbar.

2 In the navigation pane, select the category to which the privilege belongs from the list, **Administration Privileges**, or **Product Level Privileges**.

3 Expand the folder:  for the subcategory to which the privilege belongs in the navigation tree and select the privilege.

4 In the **Explicit Grant/Deny Rules** section of the content pane, select the User/Group and click the  button.

5 Click OK in the De-assign User/Group from Privilege Rule dialog box.I.

Enabling or Disabling General Grant Rules

Purpose Follow this procedure when you want to enable or disable general privilege rules for an object class or administrative function.

PRIVILEGES Manage Privileges

To enable or disable general grant rules:

1 On the Privileges tab, click the Privileges button:  on the toolbar.

2 In the navigation pane, select the category to which the privilege belongs from the list, **Administration Privileges**, or **Product Level Privileges**.

3 Expand the folder:  for the subcategory to which the privilege belongs in the navigation tree and select the privilege.

4 In the **General Grant Rules** section of the content pane, click the  button. The Enable/Disable General Grant Rules dialog box appears.

5 Select or deselect the rules as required:

- To select a rule, check the corresponding box
- To deselect a rule, uncheck the corresponding box
- To select all the rules, click the Select All button
- To deselect all the rules, click the Deselect All button.

6 Click OK to confirm the changes.

Assigning or Unassigning Privilege Rules for a Role

Purpose Follow this procedure when you want to assign or unassign privilege rules to a role.

PRIVILEGES Manage Privileges

To assign or unassign privilege rules for a role:

1 Do one of the following:

To assign/unassign a selected privilege rule for a role:

a On the Privileges tab, click the Privileges button:  on the toolbar.

b In the navigation pane, select the category to which the privilege belongs from the list, **Administration Privileges**, or **Product Level Privileges**.

c Expand the folder:  for the subcategory to which the privilege belongs in the navigation tree and select the privilege.

d In the **Other Grant Rules** section of the content pane, click the  button and select the rule.

To assign/unassign a selected role for a privilege rule:

a On the Privileges tab click the Roles button: 

b In the navigation pane, select the role name for which you want to assign a privilege rule.

c In the content pane, locate the section corresponding to the object class or functional area for which you want to assign privilege rules.

d Click the  button in that area and select a rule.

2 In the Assign Other Grant Rules dialog box move the roles you want to be assign into the **Assigned Roles** list:

- To add a role, select the name in the **Available Roles to Assign** list, and click the **>>** link to move it to the **Assigned Roles** list
- To remove a role, select the name in the **Assigned Roles** list, and click the **<<** link.

3 Click OK to confirm the changes. A confirmation dialog box is displayed.

4 Click Yes to commit the changes.

About Mail Notifications

A mail notification defines an event in Dimensions CM that causes an email message to be sent to specified users or groups. Dimensions CM provides a set of system-defined notification rules that are triggered when common events take place, such as:

- An item or request arrives in a user's inbox.
- An attribute of an item, request, or baseline in a user's inbox is updated.
- An item is added or removed from a project.
- A user is added or removed from the base database.

- A product is added or removed from the base database.

In the Dimensions Administration Console, you can:

- View system-defined notifications
- Subscribe users or groups to notifications

PRIVILEGES Manage Email Notifications

Invocation Dimensions Administration Console | Users and Roles | Mail events

Mail Events Main Window

The Mail Events main window consists of the following:

- **Menu area:** Displays the program's tabs and toolbars to help you carry out various tasks.
- **Navigation pane:** Allows you to view and select from a list of mail events in the base database.
- **Content pane:** Displays details about the selected mail event or a summary of all events.
- **Status area:** Displays log in details.

For the location of the parts of the Administration Console main window, see "The Specific Function Window" on page 71

Mail Events: Toolbar

The toolbar displays the available commands for mail events.

Button	Function
	Subscribe groups to a mail event using the Subscribe Groups dialog box.
	Subscribe users to a mail event using the Subscribe Users dialog box.

Mail Events: Navigation Pane

The navigation pane lists mail events in the base database, grouped by object class. You can:

- Select the top-level icon:  to view a summary of all mail events in the content pane.
- Select an object class icon:  to view a summary of all mail events for that object class.

- Select an event icon: to view a the details for a specific mail event in the content pane.

Mail Events: Content Pane

The information displayed in the content pane depends on what you have selected in the navigation pane:

If you select...	Then the content pane displays...
Base Database or Object Class	■ List of mail events and associated descriptions, creation details, and update details. ■ : Click to print the list of mail events. ■ : Click to save the list of mail events as comma-separated values.
A specific mail event	■ General tab that displays the mail event's name, description, creation details, update details, event type, templates used for the email notification, object class, object type, and event context. ■ Group Subscribers and User Subscribers tabs that display the groups and users subscribed to the mail event and subscription details. For promotion and demotion subscriptions, user/group subscriptions for specific stages are displayed. For Deploy and Rollback subscriptions, areas are also displayed. ■ : Click to add to the list of users or groups that are subscribed to the mail event. ■ : Click to delete an entry from the list of users or groups that are subscribed to the mail event.

Subscribing Users and Groups

Subscribing Users to Mail Events

Purpose Follow this procedure to specify which users are to receive an email notification when the selected event occurs. This subscription cannot be removed or modified by the user.

PRIVILEGES Manage Email Notifications

To add users to a mail event subscription:

1 In the navigation pane, select the mail event that you want to subscribe the user to.

2 Do one of the following:

- Click the User button:
- Click the button on the User Subscribers tab.

The Subscribe Users dialog box appears.

3 If you want to apply the subscription to a specific project or stream, use the browse button to select the **Scope to Project/stream.** (This option only applies to deployment-related events.)

4 If you want to apply the subscription to a specific stage in the GSL, select it from the **Stage** list. (This option only appears for deployment-related events.)

5 If you want to apply the subscription to a specific area, select it from the **Area** list. (This option only appears for deployment and rollback events.)

6 If you want multiple notifications for the same user to be combined into a single email, select **Digest**.

7 If you do not want a user who triggers an event to receive an email notification, select **Do not notify originator of the event**.

8 Select the user that you want to subscribe from the **Available Users** list. For multiple users, press CTRL while selecting the users.

9 Click the **>>** link.

10 If you do not want to add a user to the list, select it in the **Users to subscribe** list, and click the **<<** link.

11 Click OK.

To unsubscribe a user from a mail event:

1 In the navigation pane, select the mail event that you want to unsubscribe the group from.

2 In the content pane, select the entry on the Subscriber Users tab.

3 Click the Delete button:  .

The Unsubscribe Users dialog box appears.

4 Click OK.

Subscribing Groups to Mail Events

Purpose Follow this procedure to specify which groups are to receive an email notification when the selected event occurs. This subscription cannot be removed or modified by the users in the subscribed groups.

PRIVILEGES Manage Email Notifications

To subscribe a group to a mail event:

1 In the navigation pane, select the mail event that you want to subscribe the group to.

2 Do one of the following:

- Click the Group button: 
- Click the  button on the User Subscribers tab.

The Group Subscription dialog box appears.

3 If you want multiple notifications for the same group to be combined into a single email, select **Digest**.

4 If you do not want a user who triggers an event to receive an email notification, select **Do not notify originator of the event**.

5 If you want to apply the subscription to a specific project or stream, use the browse button to select the **Scope to Project/stream.** (This option only applies to deployment-related events.)

6 If you want to apply the subscription to a specific stage in the GSL, select it from the **Stage** list. (This option only applies to deployment-related events.)

7 If you want to apply the subscription to a specific area, select it from the **Area** list. (this option only appears for deployment and roll back events.)

8 Select the group that you want to subscribe from the **Available Groups** list. For multiple groups, press CTRL while selecting the users.

9 Click the **>>** link.

10 If you do not want to add a group to the list, select it in the **Groups to subscribe** list, and click the **<<** link.

11 Click OK.

To unsubscribe a group from a mail event:

1 In the navigation pane, select the mail event that you want to unsubscribe the group from.

2 In the content pane, select the entry on the Group Subscribers tab.

3 Click the Delete button:  .

The Unsubscribe Groups dialog box appears.

4 Click OK.

About User Interface Profiles

Purpose User interface (UI) profiles allow you to select which features in the Dimensions CM client tools are visible to a user or group. By showing only the relevant views and operations, profiles simplify the UI and enable users to focus on their tasks.

Profiles are not meant to enforce security. Use privileges to grant or deny access to features.

There is a default UI profile that is created when you first install Dimensions CM. When you create a new user, the default UI profile is automatically assigned to that user. You can change this UI profile, and you can create other UI profiles and assign them to users or groups.

NOTE When upgrading from a previous version of Dimensions CM 12.1 (or later) the installer does not automatically add new functionality to existing user profiles. You may need to add these functions manually to each existing user profile you have defined.

NOTE There is also another way of determining which objects a user can see in the Dimensions CM clients. This is by setting an entry in the `DM.CFG` file. If this option is set, the clients will only list objects that are owned by products on which the current user holds a role. For details of how to set this option, see the *System Administration Guide*.

PRIVILEGES Manage User Interface Profiles

Invocation Dimensions Administration Console | Users and Roles | User interface profiles

User Interface Profiles Main Window

The User Interface Profiles main window consists of the following:

- **Menu area:** Displays the program's tabs and toolbars to help you carry out various tasks.
- **Navigation pane:** Allows you to view and select from a list of objects, which includes profile names, profile assignments, or users, depending on what you have selected.
- **Content pane:** Displays details about the selected objects or a summary of all objects.
- **Status area:** Displays log in details.

For the location of the parts of the Administration Console main window, see "The Specific Function Window" on page 71

The information that displays in the menu, navigation, and content panes varies, depending on which tab is active and what you have selected. For more details see:

- "Profile Definitions Tab" on page 105
- "Profile Assignments Tab" on page 108

Profile Definitions Tab

Profile Definitions Tab: Toolbar

The toolbar displays the available commands for managing profiles.

Button	Function
	Create a new profile using the New UI Profile dialog box.
	Create a new profile using the details of an existing profile.
	Delete the selected profile.

Profile Definitions Tab: Navigation Pane

The navigation pane lists the profiles in the base database. You can:

- Click **Profile Names** at the top of the navigation pane to:
  - View a summary of all profiles in the content pane.
  - Select multiple profiles in the content pane.
- Select a profile to view its details or to perform an operation on it.

Profile Definitions Tab: Content Pane

The information displayed in the content pane depends on what you have selected in the navigation pane:

If you select...	Then the content pane displays...
Profile Names	■ A list of all profiles in the base database and their descriptions. ■  : Click to modify details for the selected profile using the Edit UI Profile Definition dialog box. You can also click the name of a profile to open the Edit UI Profile Definition dialog box. ■  : Click to print the summary of profiles. ■  : Click to save the summary of profiles as comma-separated values.
A single profile name	■ Tabs that display the profile's description and view and operation settings. ■  : Click to modify details for the profile using the Edit UI Profile Definition dialog box.

Defining Profiles

Adding Profiles

Purpose Follow this procedure to create a new profile that controls which features are visible in the Dimensions CM clients. You can also create a profile using the details of an existing profile.

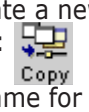
PRIVILEGES Manage User Interface Profiles

NOTE When new functions are added in a release of Dimensions CM, they are tuned off by default, so you may need to update existing profiles if you want users to have the new features available.

To add a new profile:

1 On the Profile Definitions tab, do one of the following:

- To create a new profile, click the New button: 

- To create a new profile using the details of an existing profile, click the Copy button: 

2. Enter a name for the profile in the **Name** field.
3. Optionally, enter text that describes the profile in the **Description** field.
4. Click the Views tab and select which views and subviews should be visible with the profile.
5. Click the Operations tab and select which commands should be visible with the profile.
6. To create another profile after adding this profile, select the **Keep open** check box.
7. Click OK.

   You can now assign the new profile to the appropriate users or groups.

Modifying Profiles

Purpose Follow this procedure to change the details of an existing profile, or to delete an existing profile.

PRIVILEGES Manage User Interface Profiles

To modify a profile:

1. On the Profile Definitions tab, select the profile that you want to change.
2. Click the **Edit** button: 
3. Change the profile as necessary.
4. Click OK.

   The changes are in place the next time the users log on and select the profile.

To delete a profile:

1. Select the profile that you want to delete.
2. Click the **Delete** button: 

Profile Assignments Tab

Profile Assignments Tab: Menu Area

The menu area displays the available commands for assigning profiles.

Button	Function
	Switch to view by users and groups.
	Switch to view by profiles.
	Assign a profile to a user or group using the Add Assignment dialog box.

Profile Assignments Tab: Navigation Pane

The navigation pane lists the profiles or users and groups in the base database. What appears in the navigation pane depends on which toolbar button you select in the menu area.

If you clicked the User button:  , you can:

- Select from the list to view profile assignments by all active users, all groups, auto registered users (proxy), or dormant users (inactive).
- Select a user or group to view and manage profiles currently assigned to that user or group.

If you clicked the Profiles button:  , you can:

- Click **Profile Names** to view and manage all profile assignments, listed by profile name.
- Select a profile to view and manage all assignments for that profile.

Profile Assignments Tab: Content Pane

The information displayed in the content pane depends on what you have selected in the navigation pane:

If you select...	Then the content pane displays...
The Users or Groups icon	■ A list of profile assignments for all of the users or groups. ■ : Click to assign a profile to all of the users or groups. ■ : Click to print the summary of profile assignments. ■ : Click to save the summary of profiles assignments as comma-separated values.
A single user or group name	■ A list of profile assignments for the selected user or group. ■ : Click to assign a profile to the user or group. ■ : Click to print the summary of profile assignments. ■ : Click to save the summary of profile assignments as comma-separated values.
The Profile Names icon	■ A list of assignments for all of the profiles. ■ : Click to assign all of the profiles to a user or group. ■ : Click to print the summary of profile assignments. ■ : Click to save the summary of profiles assignments as comma-separated values.
A single profile name	■ A list of assignments for the selected profile. ■ : Click to assign the profile to a user or group. ■ : Click to print the summary of profile assignments. ■ : Click to save the summary of profiles assignments as comma-separated values.

Assigning Profiles

Assigning Profiles

Purpose Follow this procedure to assign one or more profiles to a user or group. This assignment determines which profiles users can choose from when working in the client tools.

PRIVILEGES Manage User Interface Profiles

To assign a profile by user or group:

1 On the Profile Assignments tab, click the Users button: 

.

2 Do one of the following:

- To assign a profile to all users or groups, select a value from the list, and then select the Users or Groups icon.
- To assign a profile to a user or group, select a value from the list, and then select a single user or group.

3 Click the Add button:  .

The New UI Profile Assignment dialog box appears.

4 Select the profile from the **Project Profile** list.

5 Click OK.

To add an assignment by profile:

1 On the Profile Assignments tab, click the Profiles button: 

2 Do one of the following:

- To assign all profiles to a user or group, select the Profile Names icon.
- To assign a profile to a user or group, select a single profile.

3 Click the Add button:  .

The New UI Profile Assignment dialog box appears.

4 Select the user or group from the **User Name** list.

5 Click **OK**.

To assign a profile without selecting anything:

1 On the Profile Assignments tab, click the Add UI Profile Assignment button: 

.

2 Select the user or group from the **User Name** list.

3 Select the profile to assign from the **Project Profile** list.

4 Click **OK**.

Removing Profile Assignments

Follow this procedure when you no longer want a profile to be available to a user or group.

PRIVILEGES Manage User Interface Profiles

To remove a profile assignment:

1. On the Profile Assignments tab, click the Users button: 

or Profiles button: 

.
2. Select a user, group, or profile in the navigation pane.
3. Select the assignments that you want to remove from the content pane.
4. Click the Delete button:  .

   The Delete dialog box displays the profile assignments that will be removed.
5. Click **Yes** to remove the assignments.

Chapter 5

Design Part Structure

In this Chapter

About Design Part Structure

Purpose

Dimensions® CM Design Part Structure allows you to:

- Create design parts and design part variants.
- Move design parts within the design structure.
- Edit, suspend, or delete existing design parts.
- Manage design part relationships.

Constraints

You must have one of the Dimensions roles described below:

- Product Manager: These roles can perform all part operations and additionally assign Dimensions user roles within the design structure.
- PCMS Part Manager: This role can create and manipulate design parts within the scope of their role assignment in the design structure.

Invocation

Dimensions Administration Console | Product Administration | Design Part Structure

Design Part Structure Main Window

The Design Part Structure main window consists of the following:

- **Menu area:** Displays the program's toolbar to help you carry out various tasks. See Design Part Structure Menu Area on page 114.
- **Navigation pane:** Allows you to view and select from the design part structure. See Design Part Structure Navigation Pane on page 115.
- **Content pane:** Displays details about the selected design part. See Design Part Structure Content Pane on page 115.
- **Status area:** Displays log in details. See The Status Area on page 73.

For the location of the parts of the Administration Console main window, see "The Specific Function Window" on page 71

Design Part Structure Menu Area

The menu area displays the following toolbar buttons:

Button	Function
	Define a new design part or design part variant using the New Design Part dialog box or the New Variant dialog box.
	Delete the selected design part.
	Move the selected design part to a different position in the design structure using the Move Design Part dialog box.

Button	Function
	Suspend the selected design part.
	Create a usage relationship between the selected design part and another design part using the Relate To dialog box.
	Break a usage relationship between the selected design part and another design part using the Unrelate From dialog box.

Design Part Structure Navigation Pane

The navigation pane displays the product's design part structure. You can select a design part to view associated details in the content pane and perform actions on the design part.

Design Part Structure Content Pane

When you select a design part, you can do the following in the content pane:

- Edit the selected design part by clicking the Edit button:  in the Design Part Details section.
- Update the design part's PCS by clicking the PCS button:  in the Design Part Details section.

When you select the top-level icon: Product , you can do the following in the content pane:

- Select a design part to operate on it. Click  to select or deselect all of the design parts in the list.
- Print the list of design parts by clicking  .
- Save the list of design parts as comma-separated values by clicking  .
- Sort the design parts by clicking the column headings.

Defining Design Parts

Defining Design Parts

Purpose Follow this procedure to define a new design part beneath the selected design part (the parent).

When you create a new design part, a breakdown relationship forms between it and the parent design part. The new design part automatically inherits the role assignments from its parent, unless you make explicit role assignments on that design part. See "Assigning Roles" on page 92 for instructions on how to assign roles to a design part.

PRIVILEGES Create design part

To define a new design part:

1. From the Design Part Structure main window, select an existing design part to be the parent of the new design part.
2. Click the New button:  and select **Part**. The New Design Part dialog box appears.
3. Enter a name for the design part in the **Design Part ID** field.

NOTE You cannot enter a part ID that contains a colon character ":".

4. Select a design part type from the **Design Part Category** list.
5. Change the **Variant** or **PCS** default values if necessary.
6. Enter a description of the new design part in the **Description** field.
7. Click the Attributes tab to specify values for any user-defined attributes for the design part.
8. Click OK to create the new design part.

Defining Design Part Variants

Purpose Follow this procedure to define a new variant of an existing design part. You can use a variant to represent an alternative implementation of a design part or item. For example, a team might develop a product that has two different markets, American and European. To represent those markets, the team creates a set of design part variants specific to each market.

When you define a variant, it will appear at the same level as the original design part with the same part ID and PCS value. For example:

PAYROLL:REPORTS.A;1 (original)

PAYROLL:REPORTS.EUROPEAN;1 (variant)

PAYROLL:REPORTS.USA;1 (variant)

Once you've defined a variant, you can operate on it like any other design part.

PRIVILEGES Create design part

To define a design part variant:

1. From the Design Part Structure main window, select the design part from which you want to create a new variant.
2. Click the New button:  and select **Variant**. The New Variant dialog box appears.
3. Enter a name for the variant in the **Variant ID** field.
4. Optionally, enter a description of the variant in the **Description** field.
5. Click the Attributes tab to specify values for any user-defined attributes for the variant.
6. Click OK to create the new variant.

Managing Design Part Usage Relationships

Purpose Follow this procedure to define a usage relationship between two design parts. You define a usage relationship in order to reuse a design part in another part of the design structure. Usage relationships may be important when building configurations and baselines.

NOTE If you want to include usage relationships in a baseline, be sure to create the usage relationships before building the baseline.

You can break the usage relationship by removing the used design part from the parent design part.

PRIVILEGES Relate design part to design part

To create a usage relationship between design parts:

1. From the Design Part Structure main window, select the design part for which you want to create a usage relationship.
2. Click the Relate button: 

. The Relate To dialog box appears.
3. Do one of the following:
   - Enter the design part that you want to relate in the text field.
   - Click the browse button:  and complete the Select Design Part wizard.
4. Click Yes to create the relationship.

To break a usage relationship between design parts:

1. From the Design Part Structure, select the design part from which you want to break a usage relationship.
2. Click the Unrelate button: 

. The Unrelate From dialog box appears.
3. Select the design part(s) that you want to remove from the selected design part from the **Used Parts** list.
4. Click OK to remove the relationship(s).

Modifying Design Parts

Editing Design Parts

Purpose Follow this procedure to change a design part's description or attribute values.

PRIVILEGES
Update Design Part
Rename Design Part

To edit a design part:

1. From the Design Part Structure main window, select the design part you want to edit.
2. Click the Edit button:  on the Design Part Details section. The Edit Design Part dialog box appears.

3 Change the **Design part ID** or **Description** field on the General tab or any of the attribute values on the Attributes tab.

4 Click OK to save the changes.

Updating the PCS

Purpose Follow this procedure to change the part change status (PCS) for a selected design part. The PCS represents the version number of the design part.

Updating a PCS is not a commonly performed operation. Typically you would use this command when you want to work on a new version of a design part.

PRIVILEGES Update Design Part

To update a design part's PCS:

1 From the Design Part Structure window, select the design part that you want to update.

2 In the Design Part Details section, click the PCS button:  . The Update PCS dialog box appears.

3 Enter the new value for the PCS in the **New PCS** field.

4 Enter a new description for the design part in the **Description** field, if necessary.

5 Click Yes to update the PCS.

Moving Design Parts

Purpose Follow this procedure to move a design part to a new location in the design structure beneath a new parent. When you move a design part, its child design parts are moved as well.

By moving a design part, note that you are creating a new breakdown relationship between the design part and its new parent.

PRIVILEGES Relate Design Part to Design Part

Constraints
- The design part to be moved and the new parent design part must not be suspended.
- You cannot move a design part to another product.
- The design part must not have any open requests related to it.

To move a design part:

1 From the Design Part Structure window, select the design part you want to move.

2 Click the Move button: 

. The Move Design Part dialog box appears.

3 Select the design part under which you want to move the design part from the **New Parent Design Part** list.

4 Click Yes to move the design part.

Deleting Design Parts

Purpose

Follow this procedure to remove a design part from the design structure. This procedure also deletes a design part's audit history.

Deleting design parts is not a commonly performed operation. Typically you would use this command for design parts that have been created in error or have never been used.

PRIVILEGES Delete Design Part

NOTE You cannot delete a top-level design part or variant within a product.

Constraints

To be deleted, the design part must not:

- Be the top-level design part or its variant within the product.
- Be referenced by any requests.
- Have any children design parts.
- Own any items.
- Exist in any baseline.

To delete a design part:

1. From the Design Part Structure main window, select the design part you want to delete.
2. Click the Delete button: 

.
3. Confirm that you want to delete the design part by clicking Yes in the Delete dialog box.

Suspending Design Parts

Purpose

Follow this procedure to suspend a design part from the design structure. When you suspend a design part, you will still be able to view it in the design structure but will not be able to operate on it.

Suspending design parts is not a frequently performed operation. Typically you would use this command for design parts that are no longer used or required.

PRIVILEGES Suspend Design Part

To suspend a design part:

1. From the Design Part Structure main window, select the design part you want to suspend.
2. Click the Suspend button: 

3. Confirm that you want to suspend the design part by clicking Yes in the Suspend dialog box.

Chapter 6

Product Definitions

In this Chapter

About Product Definition

Purpose

Dimensions CM Product Definition allows you to:

- Create a new product based on an existing product.
- View, edit, and delete existing products.
- Assign Product Manager, Change Manager, and Parts Controller roles for a product to Dimensions CM users.

Invocation

Dimensions Administration Console | Product Administration | Product Definitions

Product Definitions Main Window

The Product Definitions main window consists of the following:

- **Menu area:** Displays the program's toolbar to help you carry out various tasks. See Product Definitions Menu Area on page 122.
- **Navigation pane:** Allows you to view and select from a list of product definitions. See Product Definitions Navigation Pane on page 122.
- **Content pane:** Displays details about the selected product definition or a summary of all products. See Product Definitions Content Pane on page 123.
- **Status area:** Displays log in details. See The Status Area on page 73.

For the location of the parts of the Administration Console main window, see "The Specific Function Window" on page 71

Product Definitions Menu Area

The menu area displays the following toolbar buttons:

Button	Function
	Define a new product using the New Product dialog box.
	Delete the selected product(s).

Product Definitions Navigation Pane

The navigation pane displays the products that exist in the base database. You can:

- Select a product:  to view associated details in the content pane and perform actions on the product.
- Select the top-level icon:  to view a summary of all products in the content pane.

Product Definitions Content Pane

When you select a product, you can edit it by clicking the Edit button: .

When you select the top-level icon: Product , you can do the following in the content pane:

- Select a product to edit or delete it. Click to select or deselect all of the products in the list.
- Print the list of products by clicking .
- Save the list of products as comma-separated values by clicking .
- Sort the products by clicking the column headings.

Defining Products

Defining Products

Purpose Follow this procedure to define a new product in the base database. When creating a new product, you must assign a registered Dimensions CM user to the Product Manager role, and can additionally assign users to the Change Manager and Parts Controller roles. Refer to "Managing Users and Groups" on page 82 for details on how to register new users.

When you create a new product, you use an existing product, or the $GENERIC product to base its details on. Object type definitions and privilege assignments are copied from that product.

You can also define values for any product attributes that appear on the Attributes tab. These attributes must have been previously defined for the product selected in the **Based On** field. To define product attributes, use the Administration Console | *Object Type Definitions* | *Attribute* function and assign attributes to the PRODUCT design part category.

PRIVILEGES Create Product

To define a product:

1. From the Product Definitions main window, click the New button: New. The New Product dialog box appears.
2. Specify a unique name for the product in the **Product ID** field.
3. Select the product that you want to use as the basis for the new product from the **Based On** list.
4. Enter a description of the product in the **Description** field.
5. Change the **Variant** or **PCS (revision no.)** default values if necessary.
6. Select a user to be the product manager for the product from the **Product Manager** list.
7. Optionally, complete the rest of the fields in the dialog box.

8 Click the Attributes tab to specify values for any user-defined attributes for the product.

9 Click OK to add the new product.

Modifying Products

Purpose Follow this procedure to:

- Edit the description and attributes associated with a product.
- Delete a product.

PRIVILEGES Modify Product, Delete Product

NOTE When you delete a product, Dimensions deletes all information about it from the database and removes any reference to a product's items from within a project/stream. However, the contents of the libraries themselves are left intact. The Tool Manager can back up or delete these files using standard operating system commands.

To edit or delete a product:

1 From the Product Definitions main window, select the product that you want to edit or delete.

2 Do one of the following:

If you want to...	Then...
Edit a product	**1** Click the Edit button:  **2** In the Edit Product dialog box, you can change the **Description** field and any user-defined attributes.
Delete a product	**1** Click the Delete button:  . **2** Confirm that you want to delete the product(s) by clicking Yes. **NOTE** If you are attempting to delete your current product, you will receive an error message. To delete it, you will need to switch to another product by clicking the product link in the status bar. After you delete the product, you will be reminded to delete the library files using operating system commands.

Chapter 7

Valid Sets

In this Chapter

About Valid Sets Management

Purpose

The management of valid sets in the Dimensions® CM Administration Console allows you to:

- Define and modify valid sets.
- Assign valid sets to user-defined attributes.

Invocation

Dimensions Administration Console | Product Administration | Valid Set Definitions

For more information

For a description of valid sets, see Valid Sets of Values on page 43.

Valid Sets Main Window

The Valid Sets main window consists of the following:

- **Menu area:** Displays the program's toolbar to help you carry out various tasks. See Valid Sets Menu Area on page 126.
- **Navigation pane:** Allows you to view and select from a list of valid sets. See Valid Sets Navigation Pane on page 126.
- **Content pane:** Displays details about the selected valid set or a summary of all valid sets. See Valid Sets Content Pane on page 127.
- **Status area:** Displays log in details. **Status area:** Displays log in details. See The Status Area on page 73.

For the location of the parts of the Administration Console main window, see "The Specific Function Window" on page 71

Valid Sets Menu Area

The menu area displays the following toolbar buttons:

Button	Function
	Define a new valid set using the New Valid Set dialog box.
	Copy details of the selected valid set to create a new valid set using the Copy Valid Set dialog box.
	Delete the selected valid set(s).

Valid Sets Navigation Pane

The navigation pane displays the valid sets that exist in the product. You can:

- Select a valid set to view associated details in the content pane or perform actions on the valid set.
- Select the top-level icon: Valid Sets to view a summary of all valid sets in the content pane.

Valid Sets Content Pane

When you select a valid set, the content pane displays the following sections:

- **General:** Displays details about the current valid set. From this section you can edit the general details of the valid set as well as the values of the valid set.
- **Values:** Displays the values that have been defined for the valid set. From this section you can edit the values of the valid set as well as the general details of the valid set.
- **Used By:** Displays details about the attributes that have been assigned to the valid set. From this section you can assign and unassign the valid set from an attribute.

When you select the top-level icon: Valid Sets, the content pane displays a summary of the valid sets in the current product. You can:

- Select a valid set to edit, copy, or delete it. Click to select or deselect all of the valid sets in the list.
- Print the list of valid sets by clicking
- Save the list of valid sets as comma-separated values by clicking
- Sort the valid sets by clicking the column headings.

Defining Valid Sets

Defining Valid Sets

Purpose Follow this procedure to define a new valid set within the current product. You can then assign the valid set to one or more attributes.

PRIVILEGES Manage Valid Sets

To define a valid set:

1. From the Valid Sets main window, click the New button: New. The New Valid Set dialog box appears.
2. Enter a name for the valid set in the **Valid Set Name** field.
3. Enter the number of columns that you want the valid set to be comprised of in the **No. of Cols** field.
4. Optionally, complete the rest of the fields in the dialog box.
5. Click the Values tab. The Values tab displays the number of columns that you specified on the previous tab.

   You can return to the General tab to change the number of columns that appear.

6 Enter attribute values in the table cells, clicking:

-  to insert a new row below the current row.
-  to insert a new row above the current row.
-  to delete the current row.

7 Click OK to add the valid set.

Modifying Valid Sets

Purpose Follow this procedure to:

- Edit the details or values specified for a valid set.
- Copy a valid set so that you can reuse all or most of the valid set's values for a new valid set.
- Delete a valid set.

NOTE You can delete values from a valid set while they are still in use in existing attributes.

PRIVILEGES Manage Valid Sets

To edit, copy, or delete a valid set:

1 From the Valid Sets window, select the valid set that you want to edit, copy, or delete.

2 Do one of the following:

If you want to...	Then...
Edit a valid set's general details or values	1 Click the Edit button:  in the **General** or **Values** section. 2 In the Edit Valid Set dialog box, click the General tab to change the valid set's details or the Values tab to change the valid set's values. **NOTE** You can change any of the valid set's details except **Valid Set Name**.
Copy a valid set	1 Click the Copy button:  . 2 In the Copy Valid Set dialog box, enter the name of the new valid set in the **Valid Set Name** field. 3 Optionally, complete the rest of the fields in the dialog box.
Delete a valid set	1 Click the Delete button:  . 2 Confirm that you want to delete the valid set(s) by clicking Yes.

Assigning Valid Sets to Attributes

Purpose Once you've defined a valid set, you can assign it to one or more attributes. This restricts the values of an attribute to those defined in the valid set.

PRIVILEGES Manage Valid Sets

Constraint To assign a valid set to a user attribute, the valid set must be defined in the $GENERIC product.

To assign a valid set to an attribute:

1. From the Valid Sets main window, select the valid set that you want to assign to an attribute.
2. On the Used By section, click the Assign Valid Set button: . The Assign Valid Set wizard appears.
3. Select the object class that the attribute belongs to from the **Object Class** list.
4. Select a particular type in the object class from the **Object Type** list.
5. Click the Next button.
6. For single-column valid sets, select the name of the attribute to which you want to assign the valid set from the **Attribute** list.
7. For multi-column valid sets, do the following:
   a. Select the column number that you want to use as the set of values for the attribute in the **Valid Set Colum**n field.
   b. Select the attribute to which you want to assign the valid set from the **Attribute** list.
   c. Select an existing validation group name from the **Validation Group Name** list, or enter the name of a new one.
   d. Select the **Auto populate** option if you want to automatically populate the attribute if a single match against the valid set is found.
8. Click Finish to assign the valid set to the attribute.

To unassign a valid set from an attribute:

1. From the Valid Sets main window, select the valid set that you want to unassign.
2. On the Used By section, select the row with the attribute from which you want to unassign the valid set.
3. Click the Unassign Valid Set button: and confirm that you want to unassign the valid set.

Chapter 8

Object Type Definitions

In this Chapter

About Object Type Definitions

Purpose

The management of Object Type Definitions within the Dimensions® CM Administration Console allows you to:

- List the existing object types for a particular object class (item, request, baseline, design part, or project) for the current product.
- Define a new object type for a particular object class.
- Delete an object type of a particular object class.
- Specify additional options for item types, request types, and project types.
- Assign an existing lifecycle to an object type.
- View and edit the associated lifecycle details for a specific object type using the Edit Lifecycle dialog box. This is described in About Editing Lifecycles on page 187.
- List, define or update attributes for a particular object type.
- Define item libraries for the current product and also for specific item types within the product.
- Define templates for items and requests.
- Specify valid change management relationships between request types and other request or item types for the current product.
- Add, edit or delete item to item relationships or request to request relationships within the base database (i.e. across products).
- Add, edit or delete the attribute mappings used for the priming of requests for the current product or across products.
- Specify the change management rules for controlling the development of item or request types for the current product.

As there are numerous functions. Object Type Definitions is divided into a number of sections that are presented in different tabs within the content pane. More detailed descriptions of these functions will be given in each section.

PRIVILEGES Manage Object Types

Constraints

- Object types (item types, request types, baseline types, design part categories, and project types) cannot be deleted if there are any instances of that type existing within the product. For request types, this constraint applies to requests located in both the main and secondary catalogs.
- An item type or request type cannot be deleted if its rules are still enabled (see What are Change Management Rules? on page 166).

NOTE An object type is defined in relation to the currently selected product. If you want to set up an object class of the same type in different products you will need to define them separately in each product.

When you create a new product based on an existing product, the object type definitions are copied to the new product. You can also set up object types by setting your current product to the $GENERIC product. Doing this enables you to use the $GENERIC product as a template for object types when creating a new product.

Invocation

Dimensions Administration Console | Configuration Object Management | Object Type Definitions

Object Types Main Window

The Object Types main window consists of the following:

- **Menu area**: Displays a toolbar with buttons to carry out various functions. See Object Types Main Window Toolbar on page 133.
- **Navigation pane**: Displays a list of all the object types in the product for the selected object class (item, request, baseline, design part, or project). See Object Types Navigation Pane on page 133.
- **Content pane**: If no object type is selected, this contains a table listing details of all the object types for the product. See Object Types Content Pane on page 134. If an object type is selected in the navigation pane, the content pane displays specific details for that object type. Which details are displayed depends on which tab has been selected. See About the Object Type Definitions Sections on page 134.
- **Status area**: Displays log in details. See The Status Area on page 73

For the location of the parts of the Administration Console main window, see "The Specific Function Window" on page 71

Object Types Main Window Toolbar

The Object Types toolbar displays the following buttons:

Button	Function
	Allows you to create a new object type using the New Object Type dialog box.
	Allows you to create a new object type by copying the details from the selected object type using the Copy Object Type dialog box.
	Allows you to delete the selected object type(s).
	Allows you to copy the attributes and/or their attribute rules from another object type using the Copy Type Attributes dialog box.

Object Types Navigation Pane

The navigation pane contains the following:

- **Object Types:** A list of all the types for the class of object (item, request, baseline, design part, or project) selected in the **Object Class** field.
- **Object Class:** A list selection from which you choose the class of object whose types appear in the **Object Types** list. If you made a specific selection under Object Type Definitions from the main Dimensions Administration Console, this field will be set to that selection. The selections are:
  - Change Doc Types
  - Item Types

- Baseline Types
- Design Part Categories
- Project Types

Object Types Content Pane

When you select an object type in the navigation pane, the content pane displays details for that object type. Object type details are grouped into a number of sections displayed within different tabs. These groupings are discussed in About the Object Type Definitions Sections on page 134.

When the top level node, e.g: 

is selected in the navigation pane, the content pane displays a table of details for all the object types in product for the selected object class. It contains the following:

- A  button to edit the selected object type's description, and also to edit various options for item and request types.
- A  button to display the list of object types as a separate HTML page to print or save.
- A  button to display the list of object types as comma-separated values to save as a text file.
- Column headings for the object type fields. Clicking the  icon selects or deselects all the object types in the list. You can change the sort order of the list by clicking the column headings.
- A table of object types with the following details:

Field	Description
Check Box	A check box to select or deselect the object type(s) for the operations performed by the toolbar buttons.
Name	The name of the object type. Clicking on the link will open the Edit Object Type dialog box for that object type.
Lifecycle	The lifecycle that is assigned to the object type.
Description	The description of the object type.

About the Object Type Definitions Sections

Object type details within Object Type Definitions are grouped into a number of sections displayed in different tabs. When an object type has been selected in the navigation pane, one of these sections is displayed in the content pane. Clicking another tab at the top of the content pane brings that section to the front, allowing you to switch to another set of object type details.

Each section deals with a particular aspect of the object type, and may or may not exist for a particular object class. It contains a toolbar enabling you to perform certain actions on those details. Each is discussed under its own topic, listed below:

Content Pane Section	Object Classes	For information see...
General	All	General Section for Object Types on page 142.
Attributes	All	Attributes Section on page 144.
Lifecycle	All	Lifecycle Section on page 149.
Relationship Names	Items, Change Docs	Relationship Names Section for Items on page 151 *or* Relationship Names Section for Requests.
Valid Relationships	Items, Change Docs	Valid Relationships Section for Items on page 156 or Valid Relationships Section for Requests.
Prime Mapping	Change Docs	Prime Mappings Section on page 160.
CM Rules	Items, Change Docs	CM Rules Section for Items on page 167 *or* CM Rules Section for Requests.
Templates	Items, Change Docs	Templates Section for Items and Requests on page 162.
Item Library	Items	Item Libraries Section on page 173.

Adding a New Object Type

Purpose

Follow this procedure when you want to create a new type of item, request, design part, baseline, or project in the current product. You can either create a new object type from scratch, or you can base it on an existing object type. See Copying an Object Type on page 136.

When you create an object type from scratch, you create it with a name, description and general options. You can subsequently assign attributes, a lifecycle, and define other properties using the various sections within Object Type Definitions.

PRIVILEGES Manage Object Types

To add a new object type:

1 From the Object Types main window, click the New button: 

The New Object Type dialog box appears.

2 On the General tab, enter the name and description for the new object type in the **Name** and **Description** fields.

3 Optionally, associate a lifecycle with the object type.

4 If you are creating an item or request type, choose an appropriate option for the **Closest Functional Match** field based on the descriptions and the purpose for which you intend to use it.

5 Complete the Options tab as necessary.

6 If you want the dialog box to remain open after adding the new object type, select the **Keep Open** check box.

7 Click OK to commit the details.

Copying an Object Type

Purpose Follow this procedure when you want to create a new type of item, request, design part, baseline, or project in the current product based on the details from an existing object type.

PRIVILEGES Manage Object Types

To copy an object type:

1 In the Object Types main window, select the object type whose details you want to copy in the navigation pane or the content pane.

2 Click the Copy button: 

The Copy Object Type dialog box appears.

3 On the General tab, enter the name of the new object type in the **Name** field, and optionally change the existing description in the **Description** field.

4 Under the Extended Copy options, choose the additional sets of details you want to copy from the selected object type to the new object type.

5 If you are copying an item, request, or project type, complete the Options tab as necessary.

6 If you want the dialog box to remain open after copying the object type, select the **Keep Open** check box.

7 Click OK to commit the details.

Deleting an Object Type

Purpose Follow this procedure when you want to delete an object type that you created in error or is no longer required.

PRIVILEGES Manage Object Types

NOTE You cannot delete the baseline type BASELINE.

Constraints
- Object types (item types, request types, baseline types, design part categories, and project types) cannot be deleted if there are any instances of that type existing within the product. For request types, this constraint applies to requests located in both the main and secondary catalogs.
- An item type or request type cannot be deleted if its Change Management rules are still enabled.

To delete an object type:

1 In the Object Types main window, select the object type you want to delete in the navigation pane or the content pane.

2 Click the Delete button: 

. This displays a dialog box asking you if you are sure you want to delete the object type.

3 Click Yes.

Copying Attributes and Rules

Purpose Follow this procedure when you want to copy the attributes and/or their associated rules from one object type to another.

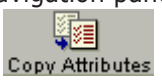
PRIVILEGES Manage Object Types

Constraints Only attributes that are not already defined for the destination object type will be copied.

Any attributes rules currently defined for the destination object type will be deleted before those for the source object type are copied. Also, attribute rules are defined in relationship to lifecycle rules and states; therefore, only those rules involving attributes and lifecycle states relevant to the object type will be copied.

To copy attributes and rules:

1 From the Object Types main window, select the object type to which you want to copy the attributes and/or rules in the navigation pane.

2 Click the **Copy Attributes** button: 

in the content pane. The Copy Type Attributes dialog box appears.

3 Select the source product and object type that you want to copy from the **Type** lists.

4 Select the **Copy Attribute Definitions** check box and/or **Copy Update Rules** check box to specify which details you want to copy.

5 Click OK to copy the details.

General Object Type Details

About the General Object Type Details

Purpose In the General section of Object Type Definitions, you can edit the following details for an object type:

- The Description given to the object type.
- For item, request, and project types, a number of options that determine their behavior. For details of these, see:
  - About Item Type Options on page 138.
  - About Request Type Options on page 139.
  - About Project Type Options on page 140.

You set these details when you first create the object type, and they can be displayed and edited using the General section.

About Item Type Options

The Item Type Options are a group of settings that determine the behavior of a particular type of item. They are represented as check boxes that you check to enable them. You can set or unset them in the Options tab of the New/Copy/Edit Object Type dialog boxes. The options for item types are:

Option	Description
Allow Parallel Checkout	When set, users can check out different or the same revisions of an item in parallel provided that the project owning the item also has the **Parallel Checkout option** set. (See "About Project Type Options" on page 140.)
Require User Comment	When set, users must enter a comment concerning the reason for the item revision update before Dimensions will permit the item revision to be created, updated or checked in. This constraint applies when creating, updating, editing or checking in an item revision, or merging two item revisions.
Auto-generate Item Identifier	When set, an item identifier is generated automatically when an item is created. In this case Dimensions will generate a unique identifier. Note that the identifier will be automatically generated even if the user enters a value.
Enable Item Header Substitution	When set, item header substitution is enabled when users get a copy of an item or browse an item (*but not when checking one out*). See Item Format Templates on page 280 for more details concerning item header substitution.
Don't Force new Revision Number on Checkin	When set, either the originator or all users (determined by the **Restrict previous Option to Originator** check box) can overwrite an item at its initial lifecycle state without changing the revision. **Note**: When a request is specified when checking out items, new revisions are always created.
Restrict previous Option to Originator	When the **Don't Force new Revision Number on Checkin** check box is set, this check box can be set or unset to specify, respectively, that either all users or only the originator can perform such overwrites.
Compress Library File	When set, files will be stored in compressed form in the Dimensions item library. For more details, see About Library File Compression on page 139. **NOTE** The compress storage option is **not** available for delta library storage.options:item types;item type:options **CAUTION!** It is strongly recommended that compression is **not** used for storing text items.
Check in only changed revisions	If set, when Dimensions receives a check in item request, it checks that the checksum for the item revision in the item library is *different* from its last value. This ensures that you cannot check in exactly the same item revision that was checked out; it forces you to change checked out item revisions before you check them in. **NOTE: This setting does not affect the web client**. In the web client, this behavior can be achieved for all item types on a per-user basis in the Preferences dialog box by deselecting *Check in* for the *If workfile unchanged* option.

Option	Description
Enforce Primary Role Constraint	When set, the user can action an item to a new, non-terminal state **only** if a primary user exists for that lifecycle state. If primary role function is assigned to a user, then leader role function (described below) cannot be assigned to the same user i.e. primary role and leader role functions are mutually exclusive.
Enforce Leader Role Constraint	When set, the user can action an item to a new, non-terminal state **only** if at least one leader user exists for that lifecycle state. If leader role function is assigned to a user, then primary role function (described above) cannot be assigned to the same user i.e. leader role and primary role functions are mutually exclusive.
Use standard level numbering scheme for revisioning	Use the standard Dimensions Algorithm for determining the numbering of item revisions when a new branch is created. This means that the first revision of the first new branch of an item will have ".1.0" appended. For example, a new branch created from revision 2.1 would be 2.1.1.0 (if a named branch is not used) and the next revision in this new branch would be 2.1.1.1, etc. If this option is not set, the first revision in a new branch will have ".1" appended. For example, a new branch created from revision 2.1 would be 2.1.1 (if a named branch is not used) and the next revision in this new branch would be 2.1.2, etc.

About Library File Compression

Dimensions CM provides a file compression utility that you can enable for an item type. This causes all files of that item type to be stored in compressed form in the relevant item library. When a user browses, gets a copy, or checks out a compressed file, Dimensions CM uncompresses the file. The compression used is affected by the settings of the parameters DM_COMPRESS_FILES_ON_TRANSFER and DM_COMPRESSION_MIN_FILELENGTH in the `dm.cfg` file. See the System Administration Guide for more details.

For example, you might want to store all binary files in compressed form.

NOTE You cannot enable file compression for delta libraries.

The compress and uncompress commands used by Dimensions CM by default are the standard Lempel-Ziv *compress* and *uncompress* utilities. This default can be overridden by setting the parameters described below. However, if you do set these parameters to use another compression utility, it must be **consistently** used for all such item types.

- UNIX

  The parameters *COMPRESS* and *UNCOMPRESS* in the Dimensions CM initialization file `$DM_PROG/dm.cfg` to suitable settings.

- Windows

  The parameters *DM_PARAM_COMPRESS* and *DM_PARAM_UNCOMPRESS* in the Dimensions initialization file `%DM_ROOT%\dm.cfg` to suitable settings.

About Request Type Options

The Request Type Options are a group of settings that determine the behavior of that type of request. They are represented as check boxes that you check to enable them. They can

be set or unset in the Options tab of the New/Copy/Edit Object Type dialog boxes. The options for request types are:

Option	Description
Create Requires Role	When set, users can create a request of this type **only** if they have the Create Request privilege. By default this means they require a role on the product to have this privilege. Note however, that users can also be granted the Create privilege for requests under General Grant Rules, "Grant to all users".
Save history	When set, request attribute and action description history is saved when actioning to a new state. If set, the user can browse a request and view it at any particular action number. If not set, the user can only browse the request at the current action number.
Notify Originator on Close	When set, Dimensions CM automatically e-mails the request's originator when a request is actioned to its end lifecycle state.
Enforce Primary Role Constraint	When set, the user can action a request to a new state **only** if a primary user exists for that state. If primary role function is assigned to a user, then leader role function (described below) cannot be assigned to the same user i.e. primary role and leader role functions are mutually exclusive.
Enforce Leader Role Constraint	When set, the user can action a request to a new state **only** if at least one leader user exists for that state. If leader role function is assigned to a user, then primary role function (described above) cannot be assigned to the same user i.e. leader role and primary role functions are mutually exclusive.
Take roles from all affected design parts	This option affects the way that roles are calculated for the Request type. When the option is set, the roles will be calculated on a request of that type by including the role assignments on all the design parts to which it is related. When the option is **not** set, role assignments will be taken only from the common ancestor design part for all the related design parts. For a more detailed explanation, see "Assigning Roles to Requests" on page 48.
Disabled request type	When set, users cannot create any new requests of this type.

About Project Type Options

Project type options determine the behavior of that type of project. They can be set in the Options tab of the New/Copy/Edit Object Type dialog boxes. The options for projects are as follows.

NOTE
These are default settings for a new project of this type. You can override these.

NOTE For streams, which are actually a project of type STREAM, some of these options are grayed-out, and cannot be changed.

Field	Description
Project is trunk	When set, the numbering of item revisions is determined by incrementing the last numeral. For example, if an item is at revision 5, subsequent revisions would be 6, 7, and so on. When the option is not set, item revisions are incremented by appending a period before the new number. For example, if a branch is created from revision 5, would be 5.1, 5.2, and so on. For streams, by default this option is not set, and cannot be changed.
Automatic revision generation	When set, new revisions are automatically generated when an item is edited or updated. When the option is not set, new revisions are not automatically generated. The user is asked to supply a revision ID. For streams, by default this option is set, and cannot be changed.
Parallel check out	When set, an item revision in this project can be checked out by multiple users at a time. Note that: ■ This will be overridden by the **Allow Parallel Checkout** setting for an individual item type. (See "About Item Type Options" on page 138.) ■ This setting will only apply to newly created projects of this type. To change this option for existing projects you will need to use the UWA command. (See the *Command-Line Reference Guide* for details.) For streams, by default this option is not set, and cannot be changed.

Field	Description
Use local stages	This deployment related option determines whether the stage of item revisions is "local" to this project/stream or can be affected by changes in other projects/streams. (Note that this means that the same item revision could be at a different stage in different projects/streams.) ■ If you select this option, the stage of an item revision in this project/stream is not affected when the same item revision is promoted/demoted to another stage as a result of activities in any other project/stream. ■ If you do not select this option, when an item revision in any project/stream is promoted/demoted to another stage, then the stage of the same item revision in this project/stream will also be updated. **Note**: The stage of an item revision can also be changed when it is actioned if its lifecycle states have been mapped to stages in the GSL.
Always disable CM Rules	When set, a project of this type will have the option for CM rules for items in that project set to **Always disabled** by default when it is created (but you can override this). When this option is set for a specific project, item operations in that project will act as if CM Rules were not enabled for any item types or request types.
Always enable CM Rules	When set, a project of this type will have the option for CM rules for items in that project set to **Always enabled** by default when it is created (but you can override this). When this option is set for a specific project, item operations in that project will act as if CM Rules were enabled for all item types and request types. For details of these CM Rules, see "CM Rules Section for Items" on page 167.
Require change control when refactoring	When set, a project of this type will have the option **Request required to refactor** set by default when it is created. When this option is set, the user will be required to provide a request ID when making any refactoring changes to the project, such as moving or renaming items or project folders.

NOTE If neither **Always disable CM rules** nor **Always enable CM rules** is set, the default behavior is for items in a project to follow the CM rules as defined for their item types. For details, see "CM Rules Section for Items" on page 167.

General Section for Object Types

The General section of the content pane displays basic details for the selected object type, and consists of:

- A Toolbar

- Details for the selected object type.

The Toolbar contains buttons to perform the following functions:

Button	Function
	Allows you to edit the General details for the object type using the Edit Object Type dialog box.

The details displayed for the object type are:

Field	Description
Name	The name that identifies the object type.
Description	The description of the object type.
Lifecycle	The Lifecycle ID associated with this object type.
Enabled Options (Only present for items, requests, and projects)	The options that have been set for the object type, indicated by a checked box. See "About Item Type Options" on page 138, "About Request Type Options" on page 139, and "About Project Type Options" on page 140 for a list and descriptions of these.

Editing an Object Type

Purpose Follow this procedure when you want to edit the general details of an item, request, design part, baseline, or project type.

PRIVILEGES Manage Object Types

To edit an object type:

1. From the Object Types main window, select the object type you want to edit in the navigation pane.
2. Click the General tab in the content pane.
3. Click the  button in the content pane. This will display the Edit Object Type dialog box.
4. On the General tab, update the **Description** field if required.
5. Optionally, if you are editing an item, request, or project type, update the Options tab for item types, request types, or project types.
6. Click OK to commit the changes.

Assigned Attribute Details

About the Assigned Attribute Details

Purpose The following functions are provided for the attributes of an object type via the Assigned Attributes section:

- List, define or update attributes for a particular object type for a particular product.
- Assign and unassign the attributes for an object type.
- Associate valid sets of values (single- or multi-column) for object attributes in the current product. Constraining certain attributes to valid sets is done for particular object types only.
- Set default values, help text, valid set usage, and so on for attributes, and to arrange any of these values into multi-row multi-column blocks.

  NOTE For individual objects this function specifies the existence of particular user-defined attributes, but does not assign values to them (other than default values). You specify attribute values for a specific object using the Dimensions web client or Dimensions desktop client.
- Specify rules on updating attributes and role sections.
- Copy existing rules and role sections from elsewhere.

Constraints In addition to the Tool Manager and Product Manager, who can update attributes for all object classes, Change Managers can update request attributes and Part Managers can update design part attributes.

Reserved Attribute Names

There are a number of existing system-defined attributes defined for the various object types. Do not use these when you define new attributes. These reserved attribute names are:

Item attributes: FORMAT, FILENAME, FILE_VERSION, ITEM_SPEC_UID, DIR_UID, LIB_FILENAME, LIB_CHECKSUM, LIB_FILE_LENGTH, CHECKSUM, FILE_LENGTH, EDITABLE, COMPRESSED, DIRPATH, USER_FILENAME, REVISED_DATE, SENDER_ID, SHARED_BRANCH, STAGE_ID, EXTERNAL, CREATE_DATE, ORIGINATOR, PHASE, LIFECYCLE

Design part attributes: PARTNO, LOCALNO, PHASE, ORIGINATOR

Request attributes: SENDER_ID, STAGE_ID, CHSEQ, CREATE_DATE, ORIGINATOR, NO_ACTIONS, SUPER_TYPE, PHASE, UPDATE_DATE, LIFECYCLE, DELEGATED_OWNER_SITE, REFERENCE_ONLY, OWNER_SITE, LOCKED

User attributes: PRIVILEGE, EMAIL, BASEDB, SITE, DEPT, PHONE, FULL_USERNAME, GROUP

Baseline attributes: SENDER_ID, STAGE_ID, BASELINE_METHOD, TEMPLATE, BASELINE_TYPE, CREATE_DATE, ORIGINATOR, PHASE

Attributes Section

The Assigned Attributes section of the content pane displays details for the attributes that are assigned to the object type selected in the navigation pane, and consists of:

- A Toolbar containing buttons to perform the following functions:

Button	Function
	Allows you to assign another attribute to the object type using the Assign Attribute dialog box. It displays two options: ■ Single Field. This displays the Assign Attribute dialog box. ■ Multi Field. This displays the Assign Block Attribute dialog box.
	Allows you to edit the details of the selected attribute. This displays the Edit Attribute dialog box corresponding to the type of attribute: ■ Single Field. This displays the Edit Attribute dialog box. ■ Multi Field. This displays the Edit Block Attribute dialog box.
	Allows you to unassign an attribute from the object type.

- A table of details for each attribute assigned to the object type:

Field	Description
Check box	A check box to select the attribute.
Name	The name that identifies the attribute.
Type	The type of attribute (SS = single field, single value; SM = single field, multi-value; MM = multi-field, multi-value).
Grouped by	The name of the block attribute (if any) that this attribute is associated with.
Prompt	The field label which is displayed for the attribute.
Data Type	The format: char (character), date, or number.
Length	The length that is displayed for the attribute.
MaxLength	The maximum number of characters this attribute can contain.
Is Sensitive	Whether the attribute is sensitive. This means that the user is required to re-enter their password when they attempt to update it.
Valid Set	The valid set assigned to this attribute (if any).

Assigned Attributes section

Assigning an Attribute to an Object Type

Purpose Follow this procedure when you want to assign an attribute to a request type, item type, baseline type, design part category, or project type. Note that you can assign attributes to a product by assigning them to the PRODUCT design part category.

You can either assign an existing attribute or create a new one.

PRIVILEGES Manage Attribute Definitions

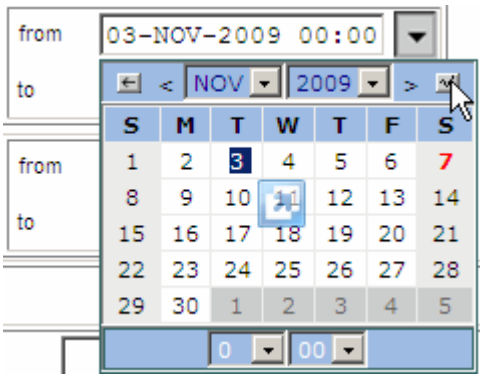
NOTE Do not use the name of an existing system-defined attribute when you define a new attribute. For a list of the reserved attribute names, see "Reserved Attribute Names" on page 144.

NOTE The ability to add a prompt or help message does not function for the Title Attribute for requests. Adding the prompt or help message will have no effect in the client GUIs in this case for the Title attribute.

To assign an attribute:

1. From the Object Types main window, select the object type you want to assign an attribute to in the navigation pane.
2. Click the Attributes tab in the content pane.
3. Click the  button in the content pane. This will display the options:
   - **Single Field**
   - **Multi Field** (block)
4. Choose the required option. The Assign Attribute dialog box or the Assign Block Attribute dialog box appears.
5. Enter a name for the attribute or attribute block in the **Name** field.
6. Enter a field label for the attribute or attribute block in the **User prompt** field.
7. For single-field attributes, follow these additional steps:
   - **a** If necessary, change the default value for the maximum character length allowed in the **Max length** field.
   - **b** Select the format for the attribute's value from the **Data type** list.
   - **c** If necessary, change the default values for the display length and height of the attribute in the **Display length** and **Display height** fields.
   - **d** Optionally, complete the rest of the fields in the dialog box. If you want the dialog box to remain open, select the **Keep Open** check box.
8. If the **Data type** is *date*, and you want to define validation for the date range:
   - Select **before** if you want to validate that the entered date must be before another date attribute entered for the object type. Select the attribute from the list:

- Select **after** if you want to validate that the entered date should be after another date attribute, and select the attribute from the list.
- Select **Fixed range** if you want to validate that the entered date should be between two specified date.
  - Use the date picker to select the **from** date and click the tick in the top right to confirm it



  - Use the date picker to select the **to** date.
- Select **Relative range** if you want to validate that the entered date should be between two specified dates relative to the current date (at the time the date is entered).
  - Select **+** or **-** from the **from** list, and enter the number of days from the current date.
  - Select **+** or **-** from the **to** list, and enter the number of days from the current date.

9 For block attributes, follow these additional steps:

a Select whether you want users to be able to modify or delete rows from the **Block Update Type** field.

b In the Attributes in Block fields, add existing single-field, multi-value attributes to the block as necessary to form the multi-field multi-value (block) attribute.

NOTE If a block attribute contains multi-value (MVA) attributes with assigned values defined as mandatory, each populated row must have values assigned to those mandatory attributes when the user enters them. This default behavior is different from previous releases. You can revert to the previous behavior by setting the DM_MANDATORY_MVA_ANY_ROW parameter in the dm.cfg file. For details see the System Administration Guide.

10 Click OK to commit the details.

NOTE If you have set the **Is sensitive** option, you will be presented with an Authentication Point dialog box. Enter your Dimensions CM password and click OK.

Unassigning an Attribute from an Object Type

Purpose Follow this procedure when you want to unassign an attribute from an object type.

PRIVILEGES Manage Attribute Definitions

To unassign an attribute:

1. From the Object Types main window, select the object type you want to unassign an attribute from in the navigation pane.
2. Click the Attributes tab in the content pane.
3. Select the attribute you want to unassign in the content pane.
4. Click the Delete button: in the content pane. This displays a dialog box asking you if you are sure you want to unassign the attribute.
5. Click the Yes button.

Editing an Attribute

Purpose Follow this procedure when you want to edit an attribute for an object type.

PRIVILEGES Manage Attribute Definitions

To edit an attribute:

1. From the Object Types main window, select the object type whose attribute you want to edit in the navigation pane.
2. Click the Attributes tab in the content pane.
3. Select the attribute you want to edit in the Attributes list.
4. Click the button in the content pane. The Edit Attribute or Edit Attribute Block dialog box appears, depending on what type of attribute you selected.
5. Make changes in the dialog box as necessary.
6. If you want the dialog box to remain open, select the **Keep Open** check box.
7. Click OK to commit the changes.

Assigned Lifecycle Details

About Assigning Lifecycles to Object Types

Purpose The majority of lifecycle operations are normally performed using the Lifecycles option from the Administration Console main window. These functions are described in About Lifecycle Management on page 178.

The lifecycle functions provided here enable you to:

- View the states and transitions of existing lifecycles as well as any associated graphical images.
- Assign a lifecycle to a particular object type for a particular product.
- Edit the details of the lifecycle associated with an object type. This lifecycle may also be currently used by a number of other object classes and types.
- Edit or delete transition and lifecycle state details for existing lifecycles.

Constraints

Please refer to About Lifecycle Management on page 178 for details on constraints in the use of lifecycles.

Lifecycle Section

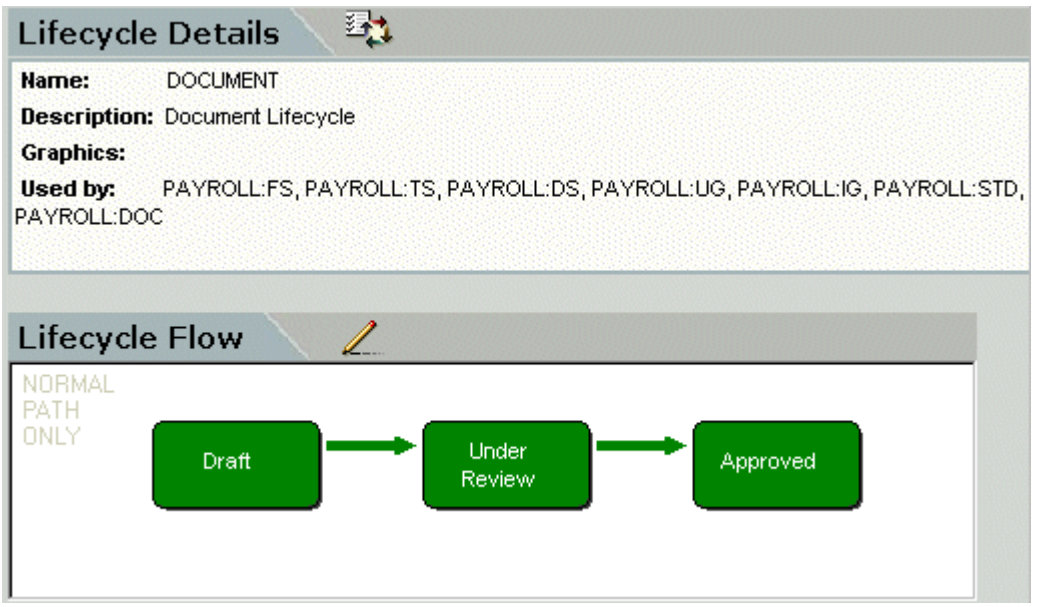
The Assigned Lifecycle section of the content pane displays details for the lifecycle that is assigned to the object type, and consists of:

- A **Lifecycle Details** area. This consists of:
  - A  **button.** This allows you to assign a lifecycle to the object type using the Assign Lifecycle dialog box.
  - Details for the assigned lifecycle:

Field	Description
Name	The name that identifies the assigned lifecycle.
Description	The description for the lifecycle.
Graphics	The name of the graphic file (if any) associated with the lifecycle.
Used By	A list of the object types that are assigned to this lifecycle.

- A **Lifecycle Flow** area. This consists of:
  - A  **button.** This allows you to edit the details of the assigned lifecycle, including its transitions and states, using the Edit Lifecycle dialog box.
  - A diagram showing the normal states in the lifecycle.

Lifecycle section



Assigning a Lifecycle to an Object Type

Purpose

Follow this procedure when you want to assign a lifecycle to an object type, change the lifecycle assignment, or unassign the assigned lifecycle.

PRIVILEGES Manage Lifecycles

NOTE The lifecycle for a design part is set to the value *LC_PART*, and cannot be changed.

To assign a lifecycle:

1. From the Object Types main window, select the object type to which to you want to assign a lifecycle in the navigation pane.
2. Click the Lifecycle tab in the content pane.
3. Click the Assign button:  in the content pane. The Assign Lifecycle dialog box appears.
4. Do one of the following

If you want to...	Then...
assign an existing lifecycle or change the assignment to another existing lifecycle	Choose the name of the lifecycle from the **Name** list.
unassign the currently assigned lifecycle	Choose **<None>** from the **Name** list.

5. Click OK.

Editing an Object Type's Lifecycle

Purpose Follow this procedure when you want to edit the details, states or transitions for the lifecycle assigned to an object type.

PRIVILEGES Manage Lifecycles

To edit a lifecycle:

1. From the Object Types main window, select the object type whose lifecycle you want to edit in the navigation pane.
2. Click the Lifecycle tab in the content pane.
3. Click the  button in the content pane. The Edit Lifecycle dialog box appears.
4. See The Edit Lifecycle Dialog Box on page 187 for details of how to edit the states, transitions and rules for the selected lifecycle.
5. When you have finished editing, click Close to return to the Object Types main window.

Relationship Names for Items and Requests

About Relationship Names

Purpose

Relationship names are used when assigning actual relationships between items and items or requests and requests. Relationship names for request to item relationships are system-defined only.

Dimensions provides functions to list, create, edit or delete item to item relationship names or request to request relationship names within the base database (i.e. across products).

PRIVILEGES Manage Item Relationship Names

Constraints

This functionality is available for items and requests only.

Relationship Names Section for Items

The Relationship Names section of the content pane for items displays details for all the relationships between items and other items in the base database. It consists of:

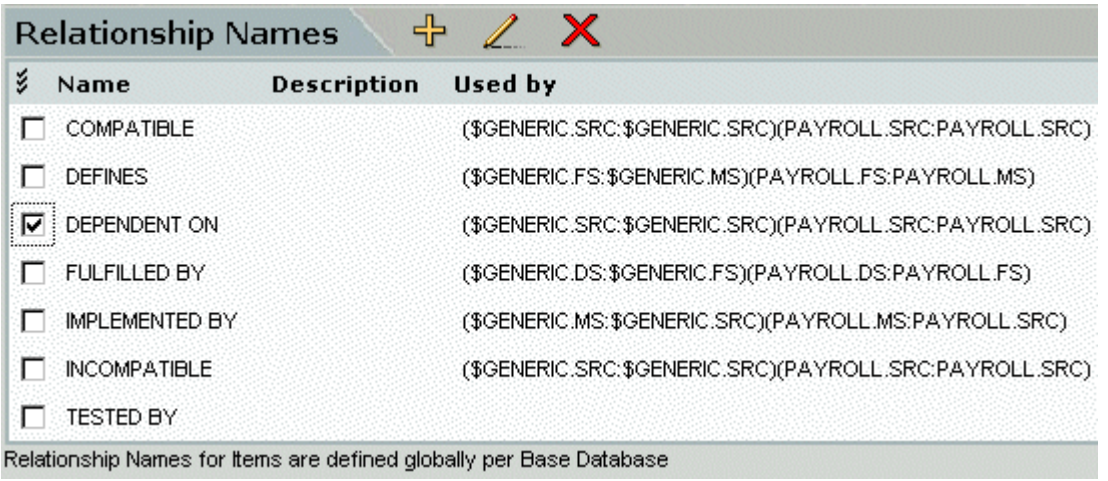
- A Toolbar containing buttons to perform the following functions:

Button	Function
	Allows you to add a new item relationship name using the New Relationship Name dialog box.
	Allows you to edit an item relationship name using the Edit Relationship Name dialog box.
	Allows you to delete an item relationship name.

- A table of details for each item-item relationship name defined in the base database:

Field	Description
Check Box	A check box to select the relationship name.
Name	The name that identifies the relationship.
Description	The description for the relationship name.
Used By	A list of the item types that have this relationship name assigned to them.

Relationship Names section for Items



Relationship Names Section for Requests

The Relationship Names section of the content pane for requests displays details for all the relationships types between requests and other requests in the product. It consists of:

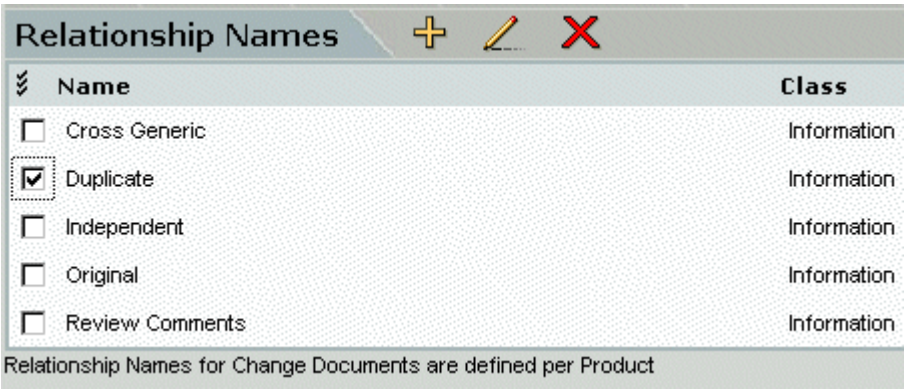
- A Toolbar containing buttons to perform the following functions:

Button	Function
	Allows you to add a new request relationship name using the New Relationship Name dialog box.
	Allows you to edit a request relationship name using the Edit Relationship Name dialog box.
	Allows you to delete a request relationship name.

- A table of details for each request-request relationship name defined within the product:

Field	Description
Check Box	A check box to select the relationship name.
Name	The name that identifies the relationship.
Class	The class of relationship: *Information* or *Dependant*.

Relationship Names section for Requests



Creating a New Relationship Name

Purpose Follow this procedure when you want to create a new relationship name to assign to valid relationships between items and items or requests and requests.

PRIVILEGES Manage Item Relationship Names/Manage Request Relationship Names

To create a relationship name:

1. From the Object Types main window, select Item Types or Request Types from the **Object Class** list in the navigation pane, according to the type of relationship name you want to create.
2. Select an object type in the navigation pane and click the Relationship Names tab in the content pane.
3. Click the button in the content pane. The New Relationship Name dialog box appears.
4. Enter the name of the relationship in the **ID** field
5. For a request type relationship, select the class on which to base the relationship from the **Class** list.
6. Optionally, for an item type relationship, enter a description of the relationship in the **Description** field.
7. Click OK to create the relationship name.

Editing a Relationship Name

Purpose Follow this procedure when you want to edit the description of a relationship name used to assign to valid relationships between items and items or requests and requests

PRIVILEGES Manage Item Relationship Names

To edit a relationship name:

1. From the Object Types main window, select Item Types from the **Object Class** list in the navigation pane.

2 Select an item type in the navigation pane and click the Relationship Names tab in the content pane.

3 Select the relationship name you want to edit in the content pane and click the  button. The Edit Relationship Name dialog box appears.

4 Edit the **ID** or **Description** fields as required.

5 Click OK to commit the changes.

Deleting a Relationship Name

Purpose Follow this procedure when you want to delete a relationship name for items or requests.

PRIVILEGES Manage Item Relationship Names

To delete a relationship name:

1 From the Object Types main window, select Item Types or Request Types from the **Object Class** list in the navigation pane, according to the type of relationship name you want to delete.

2 Select an object type in the navigation pane and click the Relationship Names tab in the content pane.

3 Select the relationship name you want to delete in the content pane.

4 Click the Delete button:  in the content pane. This displays a dialog box asking you if you are sure you want to delete the relationship name.

5 Click Yes.

Valid Relationships for Items and Requests

About Valid Relationships

Purpose Valid Relationships are set up to define which type of relationships are possible between items and requests.

NOTE Valid relationships are not effective if CM Rules have not been set up for a particular request type. Therefor any request, whose type rules are disabled, may have any product item revision related to it, provided its item type rules are also disabled.

Dimensions provides functions to list, define, edit or delete item to request relationships within the base database (i.e. across products).

PRIVILEGES Manage Object Types

Constraints This functionality is available for items and requests only.

About Minimum and Maximum Status Attribute Fields

It is sometimes necessary that the progress of a request is measured by the minimum status of its related child requests. For example, when a Problem Report (PR) is raised, a Problem Assessor may want to create a number of Change Requests (CR) from it and

assign these change requests to individual developers for implementation. In this scenario the originator of the PR may want to assess the progress of the PR in terms of the minimum status of the CRs that are related to it.

Dimensions CM provides a "status auto-tracking" option in the process model for every possible pair of parent request types and children request types. When this option is enabled for a pair such as <PR, CR>, Dimensions CM will track the minimum status of the related CRs for each PR in a *user-defined attribute* for PR specified by the project manager. The value of this attribute may be browsed or reported on using standard Dimensions CM browse and report facilities.

As well as providing the minimum status information, maximum status may also be tracked if required. The combination of minimum and maximum status may be used to provide a better indication of the progress of the requests in general.

Constraints

- Only requests that are on the normal lifecycle path can have their status combined in a parent request. If more than two requests are related to the parent request, the attribute value will be set to the minimum status of those requests that have a status in the normal path.

  For example, "WP_1" has an attribute "min_cn_status" which tracks the minimum status of requests "CN_1" and "CN_2" that are related to it as Dependent. "CN_1" is actioned to an off normal state, so the attribute will contain the status of "CN_2."
- If there are no requests related as Dependent, then the minimum status attribute will be NULL, even if requests were related at some earlier time.
- If all dependent requests are off the normal lifecycle path, then the combined status will be $OFF-NORMAL.
- You can only track the minimum status of one type of request by using a specified attribute for that request. You can use a different attribute to track a different request type.
- The minimum status is only calculated based on requests in direct relationship to the parent request. It does not take into account the dependents of the child requests.

Example: Tracking the Minimum Status

The Product Manager wants to track the minimum status of requests of type CN (Change Note) in relationship to requests of type WP (Work Package). The Product Manager defines a new attribute for the WP request type called min_cn_status, and unchecks the **Display In** options so the attribute is not visible to users. Then the Product Manager defines a valid relationship between the CN request type (child) and the WP request type (parent), and specifies the newly created attribute min_cn_status for the minimum status setting.

Example: Tracking Status with a Product-Level Lifecycle

Suppose the following lifecycles exist, with dependent request types assigned to them at the "product-level" and design parts "DP_1" and "DP_2".

Product-level	DP_1	DP_2
Created	Initial	Created
Active		
Review		Fix

Product-level	DP_1	DP_2
Working	Working	
Integrate		Done
Tested	Tested	
Released	Released	Released
Prod_integrate		Supplied
Prod_test		
Closed	Closed	Closed

In this case, if a request follows the product-level lifecycle and has status "Active" and another request follows the DP_1 lifecycle and has status "Tested", then the minimum status is "Active" and the maximum is "Tested".

The DP_2 lifecycle states "Fix", "Done", and "Supplied" do not have an equivalent in the product-level lifecycle, so "Fix" and "Done" map to "Created", which is the previous state that maps to an equivalent, and "Supplied" maps to "Released". So, if a request is on the product-level lifecycle state "Active" and another is on the DP_2 at "Done", then their combined minimum is "Created" and maximum is "Active."

Valid Relationships Section for Items

The Valid Relationships section of the content pane for items displays details for all the possible valid relationships between the selected item type and other item types. It consists of:

- A Toolbar containing buttons to perform the following functions:

Button	Function
	Allows you to add a new valid item relationship using the New Valid Relationship dialog box.
	Allows you to edit a valid item relationship using the Edit Valid Relationship dialog box.
	Allows you to delete a valid item relationship.

- A table of details for each valid relationship to other item types defined in the base database:

Field	Description
Check Box	A check box to select the relationship.
Direction	Whether the selected item type is the Parent or Child in the relationship with the listed item type.
Product	The product to which the item type belongs.
Type	The item type.
Relationship Name	The name of the relationship.

Field	Description
Inherit Parent	A check box that indicates whether a new revision of the child item will inherit all the parent relationships associated with the base revision it is being created from.
Inherit Child	A check box that indicates whether a new revision of the parent item will inherit all the child relationships associated with the base revision it is being created from.

Valid Relationships section for Items

Valid Relationships Section for Requests

The Valid Relationships section of the content pane for requests displays details for all the valid relationships between the selected request type and other item or request types. It consists of:

- A Toolbar containing buttons to perform the following functions:

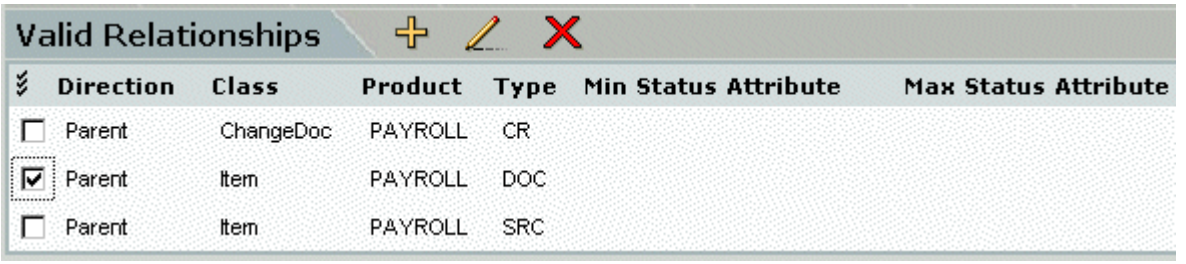
Button	Function
	Allows you to add a new valid request relationship using the New Valid Relationship dialog box.
	Allows you to edit a valid request relationship using the Edit Valid Relationship dialog box.
	Allows you to delete a valid request relationship.

- A table of details for each valid relationship with other request or item types defined in the base database:

Field	Description
Check Box	A check box to select the relationship.
Direction	Whether the selected request type is the Parent or Child in the relationship with the listed request or item type.
Class	The object class: Request or Item.
Product	The product to which the item or request type belongs.
Type	The item or request type.
Relationship	The name of the relationship.

Field	Description
Min Status Attribute	A field that indicates whether the minimum status of the request types related to a the selected request type will be automatically tracked and recorded by Dimensions. It contains the name of the attribute to contain the recorded Minimum Status.
Max Status Attribute	A check box that indicates whether the maximum status of the request types related to the selected request type will be automatically tracked and recorded by Dimensions. It contains the name of the attribute to contain the recorded Maximum Status.

Valid Relationships section for requests



Assigning a Valid Relationship

Purpose Follow this procedure when you want to assign a valid relationship between item types and/or request types.

PRIVILEGES Manage Object Types

To assign a valid relationship:

1. From the Object Types main window, select the object type you want to be the parent of the relationship in the navigation pane.
2. Click the Valid Relationships tab in the content pane.
3. Click the button in the content pane. The New Valid Relationship dialog box appears.
4. For item types, do the following:
   - **a** For the item types to be related, select the product and item type from the **Parent Item Type** and **Child Item Type** lists.
   - **b** Select the name of the relationship from **Relationship** list.
   - **c** Optionally, specify the inheritance options for the relationship.
5. For request types, do the following:
   - **a** For the request type to be related, select the object class from the **Object Class** list and the product and item type from the **Object Type** lists.
   - **b** Optionally, select the attributes to be used to track the minimum and maximum status of the child request type from the **Minimum Status** and **Maximum Status** lists.
6. Click OK to commit the details.

Editing a Valid Relationship

Purpose Follow this procedure when you want to edit the details of a valid relationship between item types and/or request types.

PRIVILEGES Manage Object Types

To edit a valid relationship:

1. From the Object Types main window, select the object type that is the parent of the relationship in the navigation pane.
2. Click the Valid Relationships tab in the content pane.
3. Select the valid relationship you want to edit in the content pane and click the  button. The Edit Valid Relationship dialog box appears.
4. Edit the inheritance options for a valid item relationship, or the **Minimum Status** and **Maximum Status** fields for a valid request relationship.
5. Click OK to commit the details.

Deleting a Valid Relationship

Purpose Follow this procedure when you want to delete a valid relationship for an item or request type.

PRIVILEGES Manage Object Types

To delete a valid relationship:

1. From the Object Types main window, select the object type that is the parent of the relationship in the navigation pane.
2. Click the Valid Relationships tab in the content pane.
3. Select the valid relationship you want to delete in the content pane.
4. Click the Delete button:  in the content pane. This displays a dialog box asking you if you are sure you want to delete the valid relationship.
5. Click the 

button.

Prime Mappings for Requests

About Prime Mappings

Purpose Prime Mappings enable you to create new requests by copying the details from existing requests (priming). When you prime a request, the values of certain attributes from the existing request are copied to certain attributes in the new request. Whether priming is possible between two types of requests, and which attribute values are copied to which, are defined by the Prime Mappings set up between the two request types involved.

NOTE The detailed description (*PCMS_CHDOC_DETAIL_DESC*) of the existing request will automatically be copied to the new detailed description unless a mapping is defined for the detailed description.

You can define the attribute mappings used for priming requests for a particular product or across products in the same database.

Priming takes into account any default attribute values set up by the Product Manager or individual user. Priming also causes the new request to be related to the product design parts that were affected by the parent request. You can change these relationships after priming is complete.

Constraints This function is available for requests only.

The attributes must have been previously defined and declared for both request types.

Prime Mappings Section

The Prime Mappings section of the content pane for requests displays details for all the request types for which attribute mappings have been defined, and allows you to create, edit, or delete such mappings. It consists of:

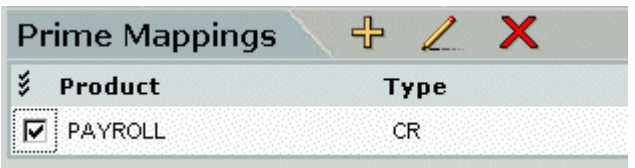
- A Toolbar containing buttons to perform the following functions:

Button	Function
	Allows you to set up a new prime mapping for a request type using the New Prime Mapping dialog box.
	Allows you to edit an existing prime mapping for a request type using the Edit Prime Mapping dialog box.
	Allows you to delete a set of attribute mappings.

- A table of details for each request type that has attribute mappings defined in the base database:

Field	Description
Check Box	A check box to select the attribute mapping.
Product	The product to which the request type belongs.
Type	The request type for which the attribute mappings are defined, that is to which the attributes are to be copied.

Prime Mappings section for requests



Creating a New Prime Mapping

Purpose Follow this procedure when you want to define the attribute mapping to be used between two request types when one request is primed from another.

PRIVILEGES Manage Object Types

To create a new prime mapping:

1. From the Object Types main window, select the request type from which the attributes will be copied in the prime operation in the navigation pane.
2. Click the Prime Mappings tab in the content pane.
3. Click the  button in the content pane.
4. On the first page on the New Prime Mapping Wizard, select the product and request type for the request type to be primed, and then click Next.
5. On the second page on the New Prime Mapping wizard, map the attributes between the parent and child request types.

   NOTE The list of attributes for **Child Attribute** includes the Detailed Description, *PCMS_CHDOC_DETAIL_DESC*, which is not selectable under **Parent Attribute**. If you do not specify a mapping for this attribute, it will be copied from the Detailed Description of the parent request.
6. When you have finished, click OK to commit the details.

Editing a Prime Mapping

Purpose Follow this procedure when you want to edit the attribute mapping to be used between two request types when one request is primed from another.

PRIVILEGES Manage Object Types

To edit a prime mapping:

1. From the Object Types main window, select the request type from which the attributes will be copied in the prime operation in the navigation pane.
2. Click the Prime Mappings tab in the content pane.
3. Select the prime mapping you want to edit in the content pane.
4. Click the  button in the content pane.
5. Update the attributes to be mapped in the Edit Prime Mapping dialog box.

   NOTE The list of attributes for **Child Attribute** includes the Detailed Description, *PCMS_CHDOC_DETAIL_DESC*, which is not selectable under **Parent Attribute**. If you do not specify a mapping for this attribute, it will be copied from the Detailed Description of the parent request.
6. When you have finished, click OK to commit the details.

Deleting a Prime Mapping

Purpose Follow this procedure when you want to delete the attribute mapping to be used between two request types when one request is primed from another.

PRIVILEGES Manage Object Types

To delete a prime mapping:

1. From the Object Types main window, select the request type from which you want to delete the prime mapping in the navigation pane.

2 Click the Prime Mappings tab in the content pane.

3 Select the prime mapping you want to delete in the content pane.

4 Click the button in the content pane. This displays a dialog box asking you if you are sure you want to delete the prime mapping.

5 Click Yes.

Templates for Items and Requests

About Templates for Items and Requests

Purpose

The functions for item header substitution files and request browse template files are similar, as template creation and management are handled in a similar fashion. The functions provided via the Templates section enable you to:

- Add a new template or a new revision of an existing template.
- Set a template as the default for an item or request type.
- Import a template file from your work area into Dimensions CM to be stored against a template revision.
- Export the template file associated with a template revision in Dimensions CM to a file in your work area.
- Delete a template or revision of a template.

PRIVILEGES Manage Object Types

Constraints

These functions are available for items and requests only.

The template files must have been previously been created outside Dimensions as described in "Item Format Templates" on page 280 and "Request Format Templates" on page 280.

What is a Template?

Templates for requests and items are used to view the contents of a request or for item header substitution, respectively. The functions provided allow you to associate pre-prepared item header substitution files or request browse template files with a template. When the template is added or imported, these template files are placed under revision control by Dimensions CM and are stored in the Dimensions CM database in a manner similar to items.

Templates for baselines and releases are different in nature and are discussed in About Baseline and Release Templates on page 202.

Templates Section for Items and Requests

The Templates section of the content pane for items and requests displays details for all the templates defined in the product for items or requests.

It consists of:

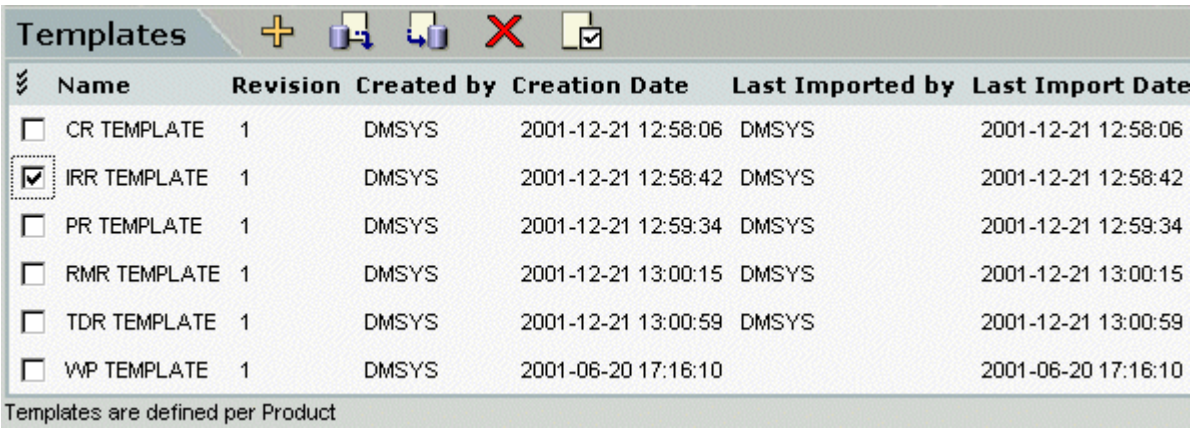
- A Toolbar containing buttons to perform the following functions:

Button	Function
	Allows you to add a new template or template revision using the New Template dialog box.
	Allows you to export a template file using the Export Template dialog box.
	Allows you to import a template file using the Import Template dialog box.
	Allows you to delete a template from the product.
	Allows you to set the selected template as the default for the object type selected in the navigation pane.

- A table listing details for each template in the product:

Field	Description
Check Box	A check box to select the template.
Name	The name that identifies the template.
Revision	The version of the template.
Created by	The user who created the template.
Creation Date	The date and time the template was created.
Last Imported by	The user who last imported the template file.
Last Import Date	The date and time the template file was last imported.
Default	Whether the template is the default for the selected item/request type.

Templates section for items and requests



Creating a New Template

Purpose Follow this procedure when you want to add a new template for an item or request type or add a new revision of an existing template.

PRIVILEGES Manage Object Types

To create a new template:

1. From the Object Types main window, select Item Types or Request Types from the **Object Class** list in the navigation pane, according to the type of template you want to create.
2. Click the Templates tab in the content pane.
3. If you want to create a new revision of an existing template, select the template in the content pane.
4. Click the button in the content pane. The New Template dialog box appears.
5. Enter the name and revision number of the template in the **Name** and **Revision** fields.
6. In the **Filename** field, enter the filename and path of the template file from your work area, or browse to the location.
7. If you want this template to be the default, select the **Default Template** check box.
8. When you have finished, click OK to commit the details.

Setting a Template as the Default

Purpose Follow this procedure when you want to make a template the default for an item or request type.

Note that you can also do this when you first create a new template or revision of a template if you have the required object type selected in the navigation pane.

PRIVILEGES Manage Object Types

To make a template the default:

1. From the Object Types main window, select Item Types or Request Types from the **Object Class** list in the navigation pane, according to the type of template you want to set as default.
2. Click the Templates tab in the content pane.
3. Select the template you want to make the default in the content pane.
4. Click the button in the content pane. The Set Default Template dialog box appears, asking you if you are sure you want to set the template as default.
5. Click Yes.

Deleting a Template

Purpose Follow this procedure when you want to delete a template or revision of a template for an item or request type.

PRIVILEGES Manage Object Types

NOTE Once a default template revision has been assigned, it can be replaced with another template revision, but there always must be a default. Note also it is not permitted to delete a default template

To delete a template:

1. From the Object Types main window, select Item Types or Request Types from the **Object Class** list in the navigation pane, according to the type of template you want to delete.
2. Select an object type in the navigation pane.
3. Click the Templates tab in the content pane.
4. Select the template revision you want to delete in the content pane.
5. Click the button in the content pane. A dialog box is displayed asking you if you are sure you want to delete the template.
6. Click Yes to confirm the deletion.

Importing a Template File

Purpose Follow this procedure when you want to import a new template file for an item or request type without changing the name or revision number of the template in Dimensions.

PRIVILEGES Manage Object Types

To import a template file:

1. From the Object Types main window, select Item Types or Request Types from the **Object Class** list in the navigation pane, according to the type of template file you want to import in the navigation pane.
2. Select an object type in the navigation pane.
3. Click the Templates tab in the content pane.
4. Select the template revision for which you want to import a template file in the content pane.
5. Click the button in the content pane. The Import Template dialog box appears.
6. Enter or browse to the file you want to import in the **Filename** field.
7. When you have finished, click OK to commit the details.

Exporting a Template File

Purpose Follow this procedure when you want to export a copy of the template file for an item or request type to your work area.

PRIVILEGES Manage Object Types

To export a template file:

1. From the Object Types main window, select Item Types or Request Types from the **Object Class** list in the navigation pane, according to the type of template file you want to export in the navigation pane.

2 Select an object type in the navigation pane.

3 Click the Templates tab in the content pane.

4 Select the template revision for which you want to export the template file in the content pane.

5 Click the  button in the content pane. The Import Template dialog box appears.

6 Enter or browse to the file to which you want to export the template in the **Filename** field.

7 When you have finished, click OK to copy the file.

Change Management Rules

About Change Management Rules

Purpose To query and define the change management rules for controlling the development of item or request types for a particular product.

Constraints This function is available for items and requests only.

A request cannot be related to another request if rules are in force for one of their types and not for the other.

What are Change Management Rules?

For item types, the CM Rules functions enable you to:

- Define when a request is required in order to perform certain functions on the item.
- Specify the minimum state an item must be in before an associated request can be closed.

For request types, the CM Rules functions enable you to:

- Specify the states for which the request is in the phases:
  - Analysis
  - Work
  - Frozen.

  The phase associated with a lifecycle state determines what kind of processing of the request is permissible while it is at that lifecycle state.
- Specify the minimum state a child request must be in before its related parent request can be closed.

These rules will apply to item types and request types for which the **CM rules enabled** option has been set. They can also become applicable to item operations in projects which have the CM rules **Always enabled** option set.

For an explanation of the effect of Change Management Rules, see About Change Management Rules on page 50.

CM Rules Section for Items

The CM Rules section of the content pane for items displays details for the Change Management rules for the selected item type. It consists of:

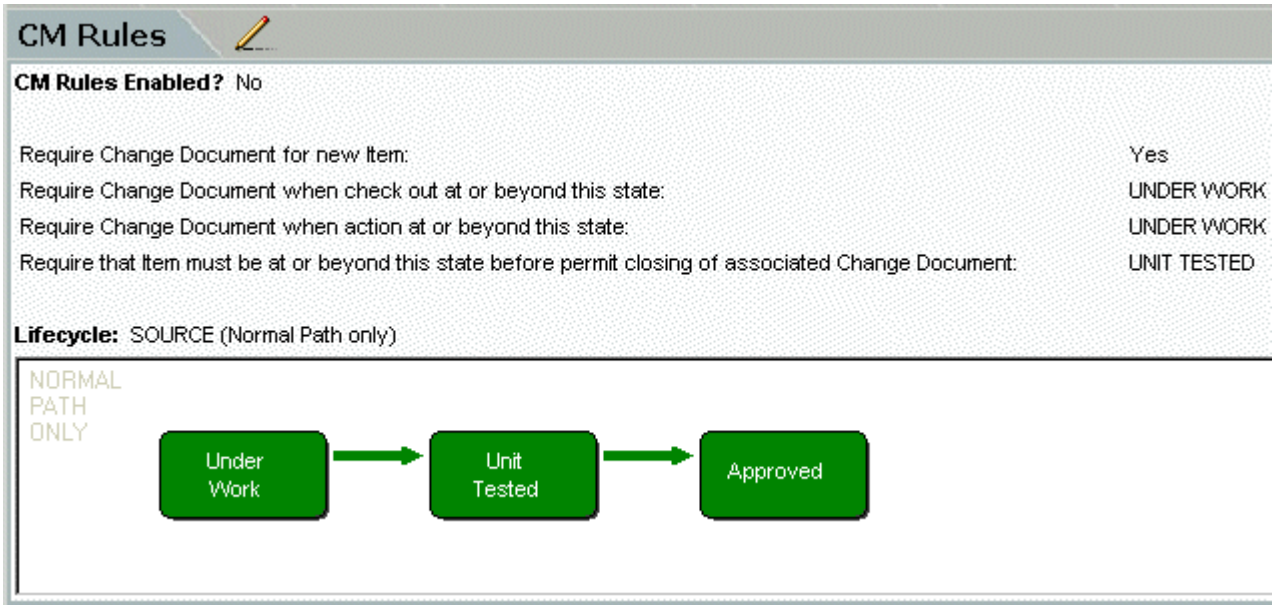
- A Toolbar containing buttons to perform the following functions:

Button	Function
	Allows you to edit the CM Rules for the selected item type using the Edit CM Rules dialog box.

- Details for the item type selected in the navigation pane consisting of:

Field	Description
CM Rules Enabled?	Whether CM Rules are enabled for this item type.
Require Request for new Item	If this option is set, a new item of this type can be created only when in response to a valid request.
Require Request when check out at or beyond this state	The first lifecycle state at which a request is mandatory for checking out a new revision of the item.
Require Request when action at or beyond this state	The first lifecycle state at which a request is mandatory for the item to progress by way of actioning.
Require that item must be at or beyond this state before permit closing of associated request	The minimum state an item must reach before requests related to it as IN RESPONSE TO can be closed.
Lifecycle:	A diagram of the normal states of the lifecycle associated with the item type.

CM Rules section for items



CM Rules Section for Requests

The CM Rules section of the content pane for requests displays details for the Change Management rules for the selected request type. It consists of:

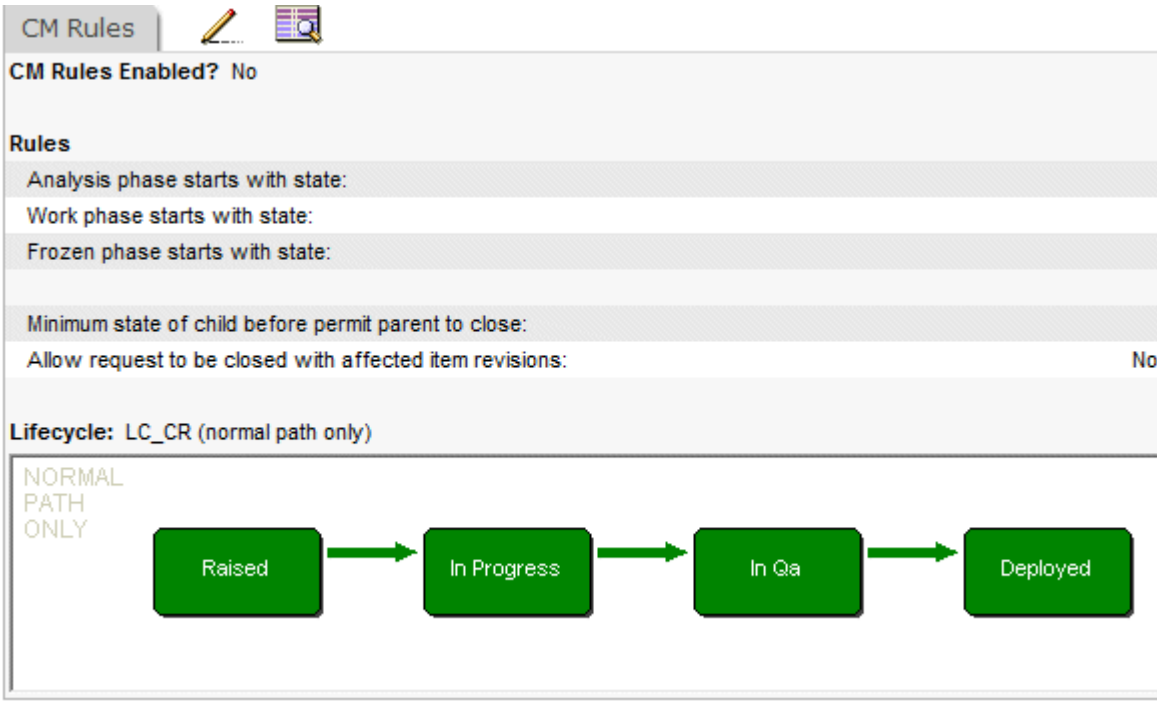
- A Toolbar containing buttons to perform the following functions:

Button	Function
	Allows you to edit the CM Rules for the selected request type using the Edit CM Rules dialog box.
	Allows you to view the phase usage for the selected request type using the CM Phase Usage dialog box.

- Details for the request type selected in the navigation pane consisting of:

Field	Description
CM Rules Enabled?	Whether CM Rules are enabled for this request type.
Analysis Phase starts with State	The first lifecycle state in the Analysis phase.
Work Phase starts with State	The first lifecycle state in the Work phase.
Frozen Phase starts with State	The first lifecycle state in the Frozen phase.
Minimum State of Child before permit Parent to close	If the request type is the child of another request, then this lifecycle state specifies the **minimum state** at which this child request must be before the parent request may be closed.
Allow request to be closed with affected item revisions	Requests can be closed/actioned to a frozen phase without having an *in-response-to* item relationship.
Lifecycle:	A diagram of the normal states of the lifecycle associated with the request type.

CM Rules section for requests



Editing CM Rules

Purpose Follow this procedure when you want to edit the Change Management Rules for a request or item type.

PRIVILEGES Manage Object Types

To edit the CM Rules:

1. From the Object Types main window, select the request or item type whose CM Rules you want to edit.
2. Click the CM Rules tab in the content pane.
3. Click the button in the content pane. The Edit CM Rules dialog box appears.
4. To enforce CM rules, select **Yes** for the **Enable CM Rules** field.
5. For item types, set the rules to coordinate items with request states in the **Require Request Rules** section.
6. For request types, map lifecycle states to phases in the **Lifecycle Phase Rules** section, and set the **Dependent Request Rule**.
7. When you have finished, click OK to commit the details.

Viewing the CM Phase Usage

Purpose Follow this procedure when you want to view the CM phases for a request type and the permitted actions for each of these phases.

To view the CM phase usage:

1. From the Object Types main window, select the request type whose CM phase usage you want to view.
2. Click the CM Rules tab in the content pane.
3. Click the  button in the content pane. The CM Phase Usage dialog box appears.
4. When you have finished, click Close to return to the Object Types main window.

Item Libraries

About Item Libraries

Purpose

Item libraries in Dimensions CM enable you to specify the operating-system directory that Dimensions CM will use to store the associated item files, and the required level of protection for that directory.

PRIVILEGES Manage Libraries

The functions provided enable you to:

- Define the default item library to be used for all item types in the product.
- Define the item library for a specific item type to override the default.
- Delete the item library definition for a specific item type.
- Delete the default item library definition for the product.

Constraints

To maintain the integrity of the library data, follow these precautions:

- Do not use the directories that store Dimensions CM libraries for any non-Dimensions CM files or subdirectories.
- Use each library for items of one or more types that belong to a single product.
- Make sure that there is adequate disk space available. Any disk-space quota limits, if enforced, must be sufficient.
- Keep library names well within any operating system limits, such as the total filename length and number of subdirectory levels. In general, keep directory names to 50-60% of any operating system limits.
- Make sure that you do not use the same directory name for Dimensions CM libraries in multiple Dimensions CM databases. This would cause Dimensions CM to be unaware of potential clashes with filenames when operating on any of these databases.

The Product Manager is permitted to revise the library definitions in this function at any time. However, Dimensions CM will not move the library contents to correspond with the revised definitions – it merely issues a warning message advising the Product Manager of the need to do this. Therefore, **the definitions must not be altered while other Dimensions CM users are logged in and using them**: to do so would be likely to cause Dimensions to report fatal library access errors. After making alterations here, the Product Manager must transfer the files in the library directories so they correspond with the new definitions before permitting other users to access the libraries for this product.

What is an Item Library?

An Item Library is an operating-system directory that you specify for Dimensions CM to use to store the associated item files, including the required level of protection. You can assign a default item library for all the items in a product and define an item library that overrides the default for a specific item type. The latter is recommended for an item type where you anticipate that there will be a substantial number of items of that type created.

Item libraries can be stored on remote servers accessed using UNC path names, or via mapped drives. These could be a Windows Network Access Storage (NAS) a fibre connected storage area network (SAN) or a network share on a PC.

NOTE It is strongly recommended on UNIX platforms that the Item Libraries are owned by the Dimensions CM Admin account and not by root.

When you define item libraries, Dimensions CM creates any directories and subdirectories which do not already exist.

Generally, items that are machine-readable entities, such as software or documentation, are stored directly in the item libraries under the control of Dimensions CM. When an item is not a machine-readable entity—for example, hardware that has design parts represented by part numbers—then a documentary control record must be created and maintained. This is a placeholder item file that represents and documents the existence of the physical item.

Defining an item library involves specifying:

- The Host machine. See About the Host Node on page 171 for details.
- The Directory path. See About the Directory Path on page 172 for details.
- Protection. See About Protection on page 172 for details.
- Whether to use Delta Storage. See About Delta Storage on page 173 for details.

About the Host Node

Dimensions CM item libraries can be defined to enable users with appropriate Dimensions CM authorization to access the items stored in these libraries from any Dimensions CM node in the network. A Dimensions CM network can consist of nodes running on different operating systems and networked in a variety of ways using TCP/IP. For further details on these networking aspects, refer to the related document, the *System Administration Guide*.

If Dimensions Network is installed and running, the host IDs (up to 20 characters and case-sensitive) define the node on the network where the item libraries will be located. The Tool Manager predefines these host IDs when setting up and maintaining the Dimensions Network. You must choose a node for the item library; * is not a valid entry. If you enter a node that is not known to Dimensions Network, you will receive an error message when you attempt to create the library.

NOTE This must not be a logical node.

If Dimensions Network is **not** installed, all filenames must be on the local node. To ensure this, enter the node name of the Dimensions CM server in the combo box.

About the Directory Path

The directory path specifies the area on the host machine in which the item library or libraries are to be stored. If desired, more than one specific item type may be stored in the same library by specifying the same directory path, as long as all of the item types use the same library storage scheme (*Normal* or *Delta*).

For libraries located on a **UNIX** system, the directory path must be an absolute path ending with a /, for example:

```
/usr/user1/libraries/type_src/
```

For libraries located on a **Windows** system, the directory path must be a an absolute path ending with a \, for example:

```
c:\usr\user1\libraries\type_src\
```

For libraries located on **z/OS**, refer to the document *Dimensions for z/OS User's and Administrator's Guide* for details on creating VSAM datasets and bringing these datasets 'online'. For details of defining nodes using the Network Administration Tool, refer to the *System Administration Guide*.

About Protection

The Protection field is used to modify, if desired, the default protections assigned to an item library when it is created (see also note below regarding library ownership).

NOTE The required protection level of the directories which hold the Dimensions CM libraries must be appropriate for the operating system **on the network node where the library is being defined**. If the Dimensions Network is being used, this node could be using a different operating system from the one on which Dimensions CM is running.

- Libraries held on UNIX

  The Protection field specifies what access to the Dimensions CM libraries is permitted directly via the operating system. The normal access indicators:

  UNIX: R = read, W = write, X = execute

  are used in appropriate combinations to provide the required level of protection. The format in which the protection is defined to Dimensions CM is:

  on UNIX: <owner>, <group>, <world>
  e.g. RX,RX,

  In the examples above, "world" has no access indicators and, therefore, no access to the directory (except for access to Dimensions CM items via Dimensions CM control).

- Libraries held on Windows systems

  This Protection field is left blank during library creation. Once a library directory has been created, and before it is used, its owner (the Product Manager) specifies the protection level required for operating-system access by creating an ACL (Access Control List) for it.

- Libraries held on z/OS systems

  Protection is not provided.

- All relevant operating systems

The criteria for choosing a protection level for these libraries is that the protection level specifies what access is required (if any) *other than* via Dimensions CM, e.g. **RX,,** *[UNIX]* may be sufficient. Dimensions will always grant access to an item, provided the user requesting it holds the required role. (*Exception*: with a directory item, although the archive/save-set file which comprises a directory item may itself be retrieved as described, its reconstruction into directories is dependent on the user having access permissions compatible with those recorded for the directories and files in the archive/save-set.)

NOTE When a library is created its ownership will be that of the Product Manager creating it – the access protection on the directory will be that specified by that user. No matter what protection is specified, Dimensions CM will always have access to these libraries as it enjoys special privileges when required to get from or write into libraries.

About Delta Storage

The option is available to store items using the native Version Manager delta library storage scheme on UNIX or Windows. In this scheme, different revisions of the same item are stored as deltas of previous revisions. This provides more efficient disk space utilization, especially for text files. However, delta libraries are not recommended for binary item files such as executables. There are some factors which may cause delta libraries to be less efficient, such as long record lengths, the speed of the delta processing and lastly, internal limits on the number of file branches.

NOTE The use of delta libraries is not supported on HP-UX 64-bit, Itanium.

NOTE UNIX Users: All new items that are created in the delta library will use the Version Manager delta scheme. However, if any items were created in the delta library using a pre-5.0 version of Dimensions CM, which utilized the Source Code Control System (SCCS) mechanism, then existing or new revisions of that item will continue to use the SCCS mechanism.

NOTE The compress storage option is **not** available for delta library storage.

Item Libraries Section

The Item Libraries section of the content pane displays details for the item library that applies to the item type selected in the navigation pane.

It consists of:

- A Toolbar containing buttons to perform the following functions:

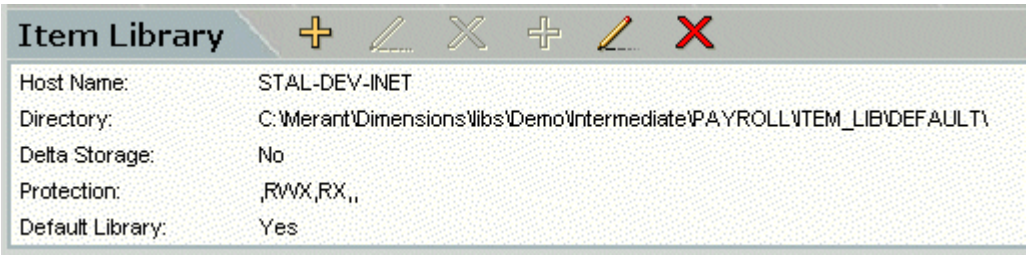
Button	Function
	Allows you to define the item library that is assigned to the selected item type using the Add Specific Library dialog box.
	Allows you to edit the item library details assigned to the specific item type using the Edit Specific Library dialog box.
	Allows you to delete the library definition for the selected item type.
	Allows you to define the default item library for the whole product using the Add Default Library dialog box.

Button	Function
	Allows you to edit the details of the default item library for the whole product using the Edit Default Library dialog box.
	Allows you to delete the default item library definition.

- Details for the item library associated with the selected item type. This may be the default library for the product, or it may be the specifically assigned library for the selected item type if one has been created. The following details are displayed:

Field	Description
Host Name	The name that identifies the network node where the item library is held.
Directory	The path of the directory on the host node where the library is held.
Delta Storage	Whether the items are stored using the native Version Manager delta library storage scheme.
Protection	The required protection level of the directories which hold the Dimensions CM libraries.
Default	Whether this library is the same as the default for the product.

Item Libraries section



Defining a Default Item Library

Purpose Follow this procedure when you want to create or edit the default Item Library Definition for the product. This will be the default for all the item types in the product unless a library is created specifically for an item type.

PRIVILEGES Manage Libraries

To define a default item library:

1. From the Object Types main window, select an item type in the navigation pane.
2. Click the Item Libraries tab in the content pane.
3. Do one of the following:
   - If there is no default item library set up, click the right-most  button in the content pane. (The right-most  button will be grayed out)**.** The Add Default Library dialog box appears.
   - If there is a default item library set up, click the right-most  button in the content pane. (The right-most  button will be grayed out)**.** The Edit Default Library dialog box appears.

4 Select the network node that holds the item library from the **HostName** list.

5 Enter the directory path to the item library on the host node in the **Directory Path** field.

CAUTION! Item libraries must not reside in the root directories of Windows drives or shares! This is unsupported, and may cause many operations to fail.

CAUTION! When the Dimensions server is installed on Windows 2008 server, item libraries cannot be located in the folders beneath the *Program Files* folder.

6 Optionally, complete the rest of the fields in the dialog box.

7 When you have finished, click OK.

NOTE If you change the location of the item library, Dimensions CM does not relocate the contents of the existing library to reflect this change. You will need to move the library files manually to the new location.

Defining an Item Library for an Item Type

NOTE If you change the location of the item library, Dimensions CM does not relocate the contents of the existing library to reflect this change. You will need to move the library files manually to the new location.

Purpose Follow this procedure when you want to create or edit the Item Library Definition for a specific item type. This will override the default library.

PRIVILEGES Manage Libraries

To define an item library:

1 From the Object Types main window, select the item type for which you want to define a library in the navigation pane. Make sure the Item Libraries section is displayed in the content pane.

2 Do one of the following:

- If there is no item library set up for the specific item type, click the left-most  button on the left of the content pane. (The left-most  button will be grayed out)**.** The Add Specific Library dialog box appears.
- If there is an item library set up for the specific item type, click the left-most  button on the left of the content pane. (The left-most  button will be grayed out)**.** The Edit Specific Library dialog box appears.

3 Select the network node that holds the item library from the **HostName** list.

4 Enter the directory path to the item library on the host node in the **Directory Path** field.

CAUTION! Item libraries must not reside in the root directories of Windows drives or shares! This is unsupported, and may cause many operations to fail.

CAUTION! When the Dimensions server is installed on Windows 2008 server, item libraries cannot be located in the folders beneath the *Program Files* folder.

5 Optionally, complete the rest of the fields in the dialog box.

6 When you have finished, click OK to commit the details.

NOTE If you change the location of an item library, Dimensions CM does not relocate the contents of the existing library to reflect this change. You will need to move the library files manually to the new location.

Deleting the Default Item Library

NOTE If you change the location of an item library, Dimensions CM does not relocate the contents of the existing library to reflect this change. You will need to move the library files manually to the new location.

Purpose Follow this procedure when you want to delete the default Item Library Definition for the product.

PRIVILEGES Manage Libraries

To delete the default item library:

1. From the Object Types main window, select an item type in the navigation pane.
2. Click the Item Libraries tab in the content pane.
3. Click the right-most button in the content pane. A dialog box is displayed asking if you are sure you want to delete the default item library.
4. Click Yes to confirm the deletion.

Deleting an Item Library for an Item Type

Purpose Follow this procedure when you want to delete the Item Library Definition for a specific item type. This will cause the default library (if one has been defined) to be used for that item type.

PRIVILEGES Manage Libraries

To delete an item library:

1. From the Object Types main window, select the item type for which you want to delete the library in the navigation pane.
2. Click the Item Libraries tab in the content pane.
3. Click the left-most button in the content pane. A dialog box is displayed asking if you are sure you want to delete the item library.
4. Click Yes to confirm the deletion.

Chapter 9

Lifecycle Management

In this Chapter

About Lifecycle Management

Purpose

Dimensions® CM Lifecycle Management enables you to specify the processing that types of item, request, and baseline can undergo. A lifecycle consists of a number of states. You define these states and the permitted transitions that can occur between any of them. You also specify which Dimensions CM user roles are assigned to each transition. Each lifecycle can be related to one or more object types, meaning that the object types are to be processed according to that particular lifecycle.

Lifecycle Management allows you to:

- List all of the existing lifecycles in the base database, or those for items, requests, or baselines.
- Create a new lifecycle from scratch or by copying an existing lifecycle.
- Delete a lifecycle.
- Relate a lifecycle to one or more object types.
- Edit certain lifecycle properties.
- Edit the states, transitions, and associated roles and rules for a lifecycle.
- View and manage graphical image files associated with lifecycles.

Constraints

- The role titles defined in each lifecycle transition must have been previously defined (except for $ORIGINATOR). See Adding Roles on page 90 for details.
- A lifecycle can be deleted only if no object types in any product currently specify it.
- A different lifecycle ID may be specified for an object type, or the lifecycle relationship may be deleted, only if there are no objects of that type already existing.

Provided the above constraints are complied with:

- The ADMIN group may update any lifecycle in the database.
- A lifecycle which is not in use by any product may be created/updated/deleted by any Dimensions CM user who holds a Product Manager role for any product in the database.
- A lifecycle which is in use in exactly one product may be updated by the Product Manager of that product.
- A lifecycle which is in use by more than one product may be updated by a Product Manager, *only* if this one Dimensions user holds the role of Product Manager *for every one of the products affected*.

In addition to the preceding, an existing object lifecycle can only be modified if no associated items or requests are currently in a checked out or held state, respectively; that is, they need to be checked in or saved. If this is not done, you will receive a warning message such as "Item requires status in lifecycle which is not included in update".

Invocation

Dimensions Administration Console | Configuration Object Management | Lifecycles

Lifecycles Main Window

From the Lifecycles main window you can:

- Create, copy, or delete lifecycles, or edit the name of a lifecycle.
- Relate a lifecycle to one or more object types.
- View and manage the image files associated with a lifecycle.

You edit the states, transitions, and associated roles and rules for a lifecycle using the Edit Lifecycle dialog box. This is described in more detail in The Edit Lifecycle Dialog Box on page 187.

The Lifecycles main window consists of the following:

- **Menu area**: Displays a toolbar with buttons to carry out various functions. See Lifecycles Menu Area on page 179.
- **Navigation pane**: Displays a list of all the lifecycles for the base database, optionally restricted by a filter that selects only lifecycles for requests, items, or baselines. See Lifecycles Navigation Pane on page 179.
- **Content pane**: If no lifecycles are selected, this contains a table listing details of all the lifecycles. If a lifecycle is selected in the navigation pane, it displays various details for the lifecycle and a diagram showing its normal states. See Lifecycles Content Pane on page 180.
- **Status area**: Displays log in details. See The Status Area on page 73

For the location of the parts of the Administration Console main window, see "The Specific Function Window" on page 71

Lifecycles Menu Area

The Lifecycles menu area contains the following toolbar buttons

Button	Function
	Allows you to create a new lifecycle using the New Lifecycle dialog box.
	Allows you to create a new lifecycle by copying the details from the selected lifecycle using the Copy Lifecycle dialog box.
	Allows you to delete the selected lifecycle(s).

Lifecycles Navigation Pane

The navigation pane contains the following:

- **Lifecycles:** A list of all the lifecycles for the base database, or those selected by the filter.
- **Filter:** A list from which you select the class of object (item, request, or baseline) whose lifecycles appear in the Lifecycles list. The selections are:

- All Lifecycles
- Requests
- Items
- Projects
- Baselines.

Lifecycles Content Pane

When you select a lifecycle in the navigation pane, the content pane displays its individual details. See Content Pane for a Single Lifecycle, described on page 180. When the top level node:  is selected in the navigation pane, the content pane displays a table of details for all the lifecycles in the navigation pane. It consists of the following:

- A  button to display the list of lifecycles as a separate HTML page to print or save.
- A  button to display the list of lifecycles as comma-separated values to save as a text file.
- Column headings for the lifecycle fields. Clicking the  icon selects or deselects all the lifecycles in the list. You can change the sort order of the list by clicking the column headings.
- A table of lifecycles with the following details:

Field	Description
Check Box	A check box to select or deselect the lifecycle for the operations performed by the toolbar buttons.
Name	The name of the Lifecycle. Clicking on the link will open the Edit Lifecycle dialog box for that lifecycle.
Description	The description of the lifecycle.
Creation Date	The date and time the lifecycle was created.
Originator	The user who created the lifecycle.

Content Pane for a Single Lifecycle

When an individual lifecycle is selected in the navigation pane, the content pane displays the following fields for that lifecycle:

- **Lifecycle Details** section. This consists of:
  - A toolbar with the following buttons:

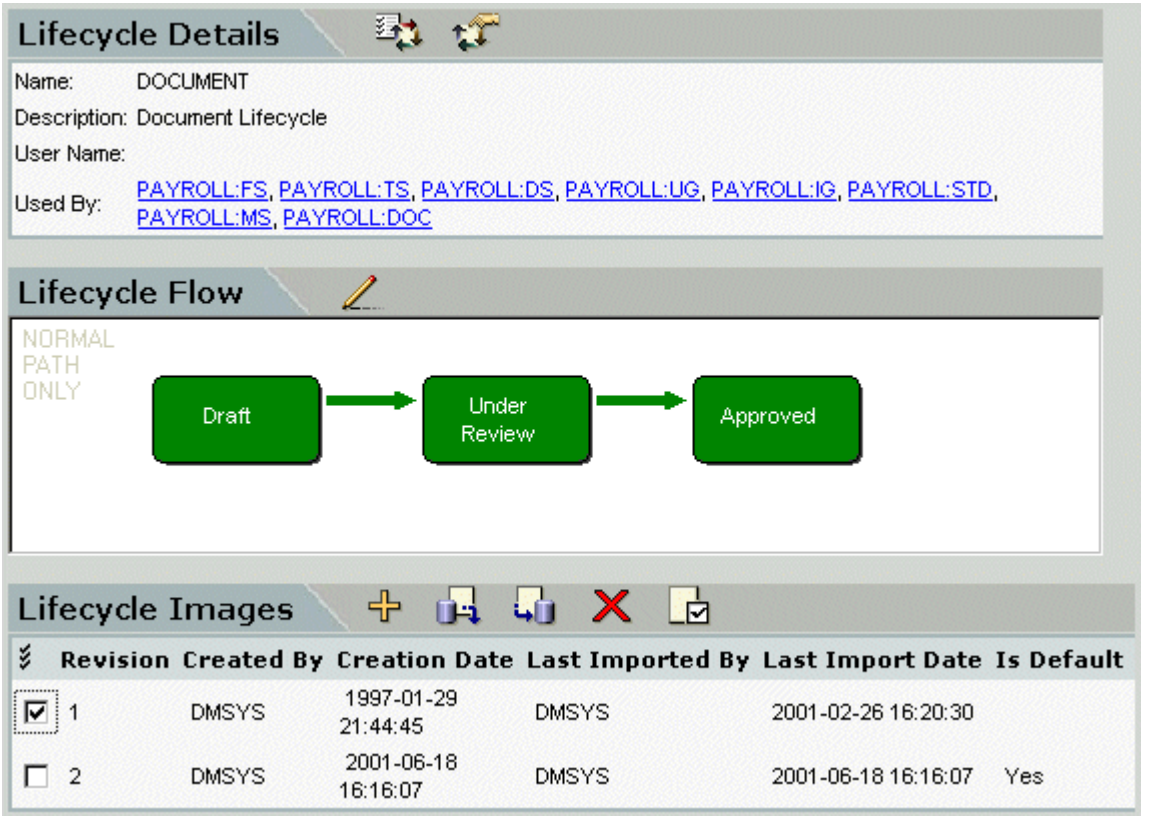
Button	Function
	Allows you to assign object types to a lifecycle using the Assign Object Types to Lifecycle dialog box.
	Allows you to edit the description for the selected lifecycle using the Edit Lifecycle Properties dialog box.

- A table displaying the following fields:

Field	Description
Name	The name of the lifecycle.
Description	The description of the lifecycle.
User Name	The user who created the lifecycle.
Used By	A comma-separated list of the object types that use the lifecycle. Clicking on the name link opens the Edit Lifecycle Dialog box with that object type selected in the Object Types filter.

- **Lifecycle Flow** section. This consists of:
  - An Edit button:  This displays the Edit Lifecycle dialog box, from which you can perform a number of lifecycle editing functions.
  - A diagram of the states in the normal path for the lifecycle.
- **Lifecycle Images** section. This enables you to maintain the images associated with the lifecycle. For details, see Lifecycle Images Section, described below.

Lifecycles Content Pane with a Lifecycle Selected



Lifecycle Images Section

The Lifecycle Images section of the content pane displays details for all the graphic files associated with the lifecycle.

It consists of:

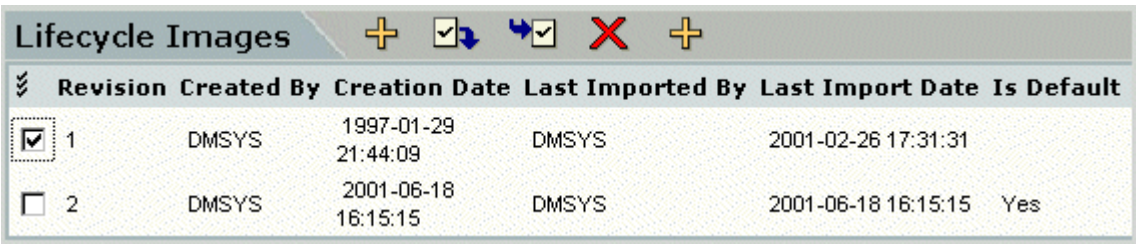
- A Toolbar containing buttons to perform the following functions:

Button	Function
	Allows you to add a new associated image using the New Lifecycle Image dialog box.
	Allows you to export an image file associated with a lifecycle to your work area.
	Allows you to import an image file using the Import Lifecycle Image dialog box.
	Allows you to delete an associated image for the lifecycle.
	Allows you to set the selected image as the default for the lifecycle selected in the navigation pane.

- A table listing details for each associated image in the product:

Field	Description
Check Box	A check box to select the image.
Revision	The version of the image.
Created by	The user who created the image.
Creation Date	The date and time the image was created.
Last Imported by	The user who last imported the image file.
Last Import Date	The date and time the image file was last imported.
Is Default	Whether the image is the default for the selected lifecycle.

Lifecycle images section



Adding a New Lifecycle

Purpose

Follow this procedure when you want to create a new lifecycle to be used by one or more types of item, request, or baseline. The created lifecycle will subsequently be available for use in *all* products within the base database.

You can ether create a new lifecycle from scratch, or you can base it on an existing lifecycle. When you copy an existing lifecycle, its states and transitions are copied to the new lifecycle.

PRIVILEGES Manage Lifecycles

To create a new lifecycle:

1 From the Lifecycles main window, do one of the following

If you want to...	Then...
create a new lifecycle from scratch	from the Lifecycles main window, click the New:  button.
copy an existing lifecycle	from the Lifecycles main window, select the lifecycle you want to copy and click the Copy:  button.

The New Lifecycle dialog box or Copy Lifecycle dialog box appears.

2 Enter the name of the lifecycle in the **Name** field, and optionally a description in the **Description** field.

3 Click OK to add the lifecycle.

To define further details for the lifecycle, such as states, transitions, and attribute rules, see About Editing Lifecycles on page 187.

Deleting Lifecycles

Purpose Follow this procedure when you want to delete lifecycles that are no longer required, or were created in error.

PRIVILEGES Manage Lifecycles

Constraints A lifecycle can be deleted only if no object types in any product currently specify it, and if all its state transitions have first been deleted.

To delete one or more lifecycles:

1 In the Lifecycles main window, select the lifecycle(s) you want to delete.

2 Click the Delete button:  You will be prompted with a dialog box asking you to confirm the deletion.

3 Click OK. Dimensions CM deletes the lifecycle(s) and removes them from the list.

Assigning Object Types to a Lifecycle

Purpose Follow this procedure when you want to assign one or more object types to the selected lifecycle. This specifies the object type as being processed according to this lifecycle's states and transitions.

Note that you can also assign a lifecycle to a specific object type in the corresponding dialog boxes in Object Type Definitions. See About Assigning Lifecycles to Object Types, described on page 148.

PRIVILEGES Manage Lifecycles

Constraints You cannot assign a lifecycle to an object type that already has a lifecycle assigned to it.

To assign object types to a lifecycle:

1. In the Lifecycles main window, select the lifecycle to which you want to assign object types.
2. Click the assign button:  The Assign Object Types to Lifecycle dialog box appears.
3. To assign object types, select them from the **Available object types** list and click Relate.
4. When you have finished, click OK to associate the object types.

Unassigning Object Types from a Lifecycle

Purpose Follow this procedure when you want to remove one or more object type(s) from the list of object types that use a particular lifecycle.

Note that you can also unassign a lifecycle from a specific object type in the corresponding dialog boxes in Object Type Definitions. See Assigned Lifecycle Details, described on page 148.

PRIVILEGES Manage Lifecycles

Constraints See About Lifecycle Management on page 178.

A different lifecycle ID may be specified for an object type, or the lifecycle relationship may be deleted only if there are no objects of that type already existing.

To unassign object types from a lifecycle:

1. In the Lifecycles main window, select the lifecycle from which you want to unassign object types.
2. Click the Assign button:  The Assign Object Types to Lifecycle dialog box appears.
3. Select the object type(s) you want to unassign in the **Object types associated with this lifecycle** list and click Unrelate.
4. Click OK to confirm the changes.

Editing the Description of a Lifecycle

Purpose Follow this procedure when you want to edit the description for a lifecycle.

PRIVILEGES Manage Lifecycles

To edit the description of a lifecycle:

1. In the Lifecycles main window, select the lifecycle to which you want to assign object types.
2. Click the Edit:  button. The Edit Lifecycle Properties dialog box appears.
3. Edit the description as required.
4. Click OK to confirm the change.

Viewing the Image of a Lifecycle

Purpose Follow this procedure when you want to view the graphical image associated with a lifecycle:

To view the image of a lifecycle:

1. In the Lifecycles main window, select the lifecycle for which you want to view the image in the navigation pane. This will display its details in the content pane.
2. Select the revision for the image you want to view in the Lifecycle Images section of the content pane.
3. Click the  button in the content pane. The image will be displayed in a new browser window.

Associating an Image File with a Lifecycle

Purpose Follow this procedure when you want to associate an image file with a lifecycle or add a new revision of an existing image file.

PRIVILEGES Manage Lifecycles

To associate an image file with a lifecycle:

1. From the Lifecycles main window, select the lifecycle in the navigation pane.
2. Click the  button in the Lifecycle Images section of the content pane. The New Lifecycle Image dialog box appears.
3. Enter the version number of the image file in the **Revision** field.
4. Enter or browse to the filename and path of the image in the **Filename** field.
5. Optionally, select the **Default Image** check box to make this image the default for the lifecycle.
6. When you have finished, click OK. The details of the new image appear in the content pane.

Setting a Lifecycle Image as the Default

Purpose Follow this procedure when you want to make an associated image file the default for a lifecycle.

Note that you can also do this when you first create a new image or revision of an image.

PRIVILEGES Manage Lifecycles

To make a lifecycle image the default:

1. From the Lifecycles main window, select the lifecycle for which you want to set the default image in the navigation pane.
2. Select the revision you want to make the default in the Lifecycle Images section of the content pane.
3. Click the  button. The Set Default Image dialog box is displayed asking you if you are sure you want to set the template as default.

4 Click Yes to confirm the change.

Deleting an Associated Lifecycle Image

Purpose Follow this procedure when you want to delete an associated image file for a lifecycle.

PRIVILEGES Manage Lifecycles

To delete an associated lifecycle image:

1 From the Lifecycles main window, select the lifecycle for which you want to delete the associated image in the navigation pane.

2 Select the revision you want to delete in the Lifecycle Images section of the content pane.

3 Click the  button in the content pane. A dialog box is displayed asking you if you are sure you want to delete the image.

4 Click Yes to confirm the deletion.

Importing an Image File

Purpose Follow this procedure when you want to import a new graphic file for a lifecycle without changing the revision number of the associated image in Dimensions.

PRIVILEGES Manage Lifecycles

To import an image file:

1 From the Lifecycles main window, select the lifecycle for which you want to import the associated image in the navigation pane.

2 Select the revision for which you want to import the file in the Lifecycle Images section of the content pane.

3 Click the  button in the content pane. The Import Lifecycle Image appears.

4 Enter or browse to the filename and path of the image you want to import in the **Filename** field.

5 Click OK to commit the details.

Exporting an Image File

Purpose Follow this procedure when you want to export a copy of the associated image file for a lifecycle to your work area.

To export an image file:

1 From the Lifecycles main window, select the lifecycle for which you want to export the associated image in the navigation pane.

2 Select the revision that you want to export in the Lifecycle Images section of the content pane.

3 Click the  button in the content pane. The image will be displayed in a new browser window.

4 Use the *Save As* function of the browser to save the image to your work area.

5 Click OK to copy the file.

About Editing Lifecycles

Editing a lifecycle involves the following:

- Deleting a state from a lifecycle.
- Renaming a lifecycle state.
- Defining or editing the available transitions for lifecycle states.
- Defining or editing the attribute rules for lifecycle transitions.
- For items or requests, defining the Change Management rules for lifecycle states.
- For items and requests, defining whether a lifecycle state is sensitive.
- For items, associating a build stage with a lifecycle state.

  NOTE You create a new state only in the New Transition dialog box when adding a new transition. This is because you cannot add a state to a lifecycle without there being a possible transition for that state.

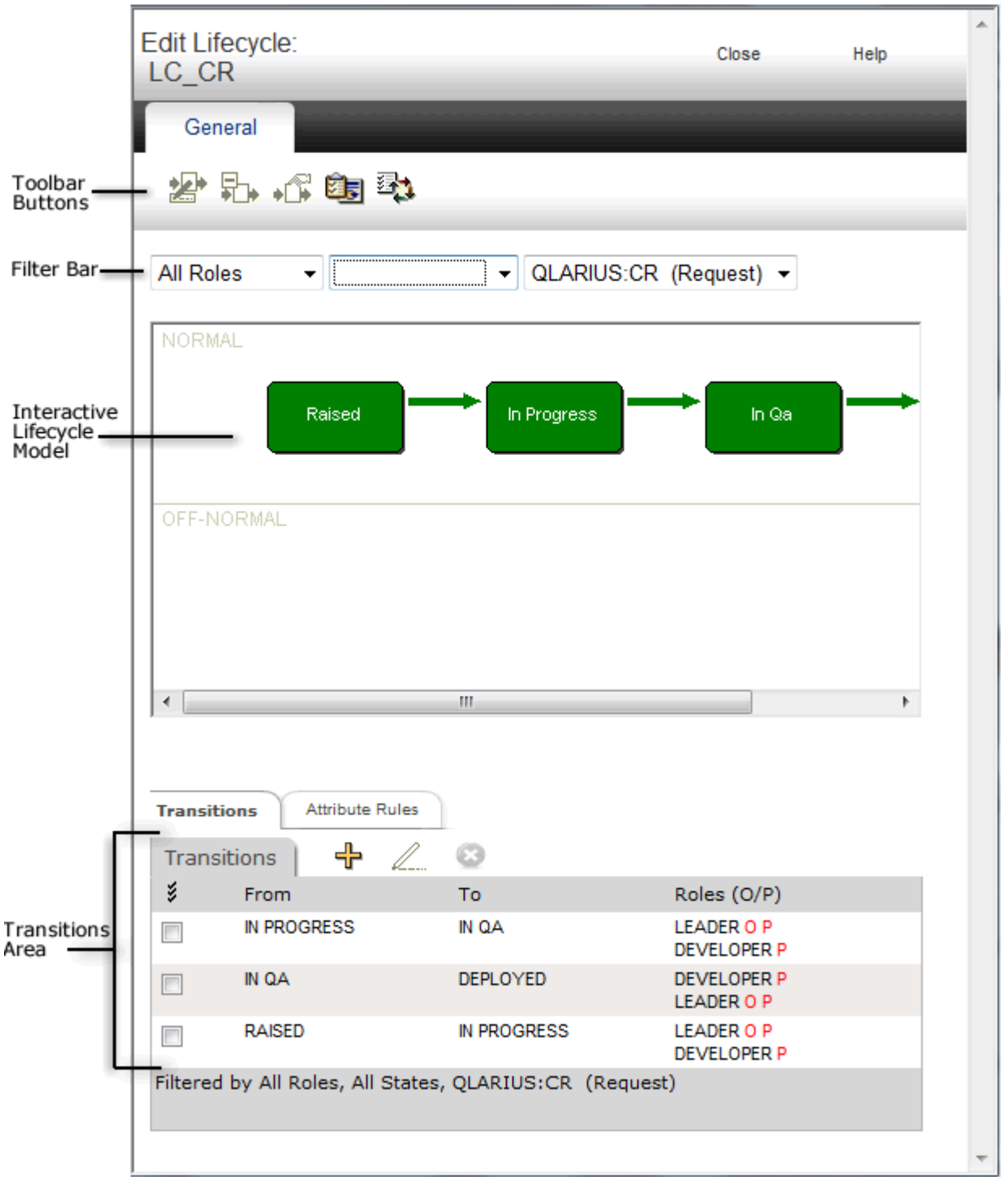
The Edit Lifecycle Dialog Box

The Edit Lifecycle dialog box allows you to perform a number of functions. From within it, you can edit the details of the states, transitions, and rules for the selected lifecycle. You access this editor from the Lifecycles main window by clicking the name link for the lifecycle or one of its related object types in the content pane. You can also access it from the Object Type Definitions options of the Administration Console.

The Edit Lifecycles dialog box consists of:

- A menu area. This is described in The Edit Lifecycle Menu Area on page 188.
- An Interactive Lifecycle Model, a diagram which allows you to select and control what appears in the Transitions and Attribute Rules areas. This is described in The Interactive Lifecycle Model on page 189.
- The Transitions area, a list of possible transitions and the roles required for those transitions. This is described in Transitions Area on page 194.
- The Attribute Rules area. This is described in The Attribute Rules Area on page 196.

Example Edit Lifecycle Dialog Box:



The Edit Lifecycle Menu Area

The menu area consists of:

- A Toolbar containing buttons to perform the following functions:

Button	Function
	Allows you to delete a normal state. For details, see Deleting a Normal State on page 191.
	Allows you to rename a state. For details, see Renaming a State on page 192.
	Allows you to set up or edit the CM rules for the selected state. This is only appears if a request type has been selected in the object type filter. For details, see Defining CM Rules on page 191.
	Allows you to configure a lifecycle state as a stage in Dimensions Build or to define the state as sensitive. This is only appears if an item or request type has been selected in the object type filter. *For details, see Setting the State Properties for an Object Type on page 192.*
	Allows you to assign object types to a lifecycle. For details, see *Assigning Object Types to a Lifecycle on page 183*.

- A filter bar that has three filters:
  - **Role:** When you select a role from the list, all the states in that role become highlighted. When you select a state and a role, only those transitions in or out of that state that are assigned to the selected role are displayed in the Transitions area.
  - **State:** When you select a state from the list, that state becomes highlighted in the lifecycle model. It is an alternative to selecting the state in the lifecycle model. When you select a state in the lifecycle model this filter updates to reflect your selection.
  - **Object Type:** If you have selected an object type from this list, then what is shown in the Transitions area and Attribute Rules area reflect this specific object type.

The Interactive Lifecycle Model

The Interactive Lifecycle Model has two rows showing transition states. You can scroll this diagram horizontally if it is wider than the window. The first row shows the normal transition states, in the path in which they are followed. They are connected by gray arrows that indicate the normal transitions. The second row shows the off normal transition states.

You can select a state by clicking it or by choosing it in the State filter. The arrows representing the possible transitions into that state become highlighted in light blue, and the transitions out of it become highlighted in dark blue. When a state is selected, the Transitions area only shows the possible transitions to and from that state, whereas when no state is selected all the transitions for the lifecycle are shown.

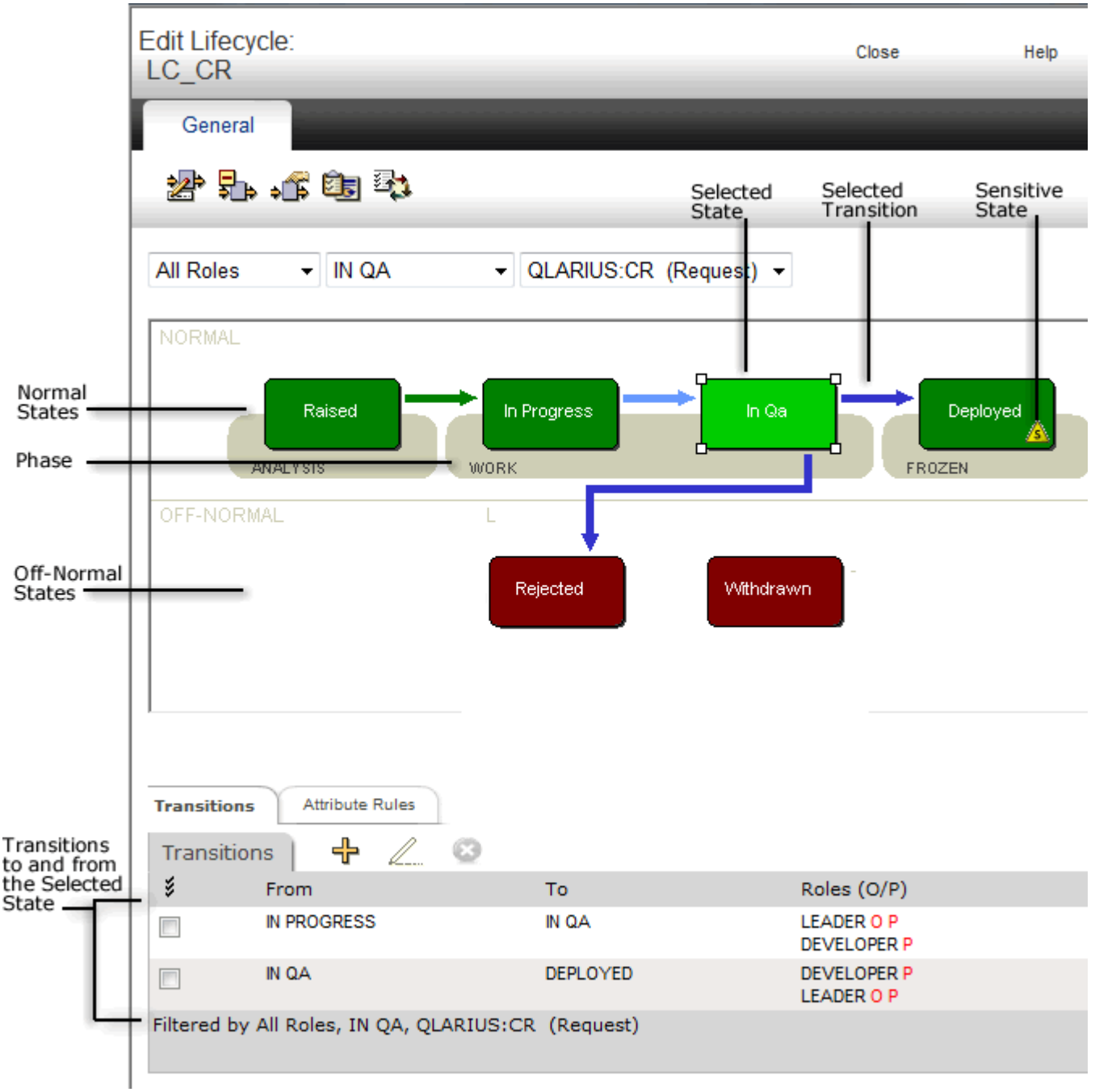
When a request type is selected in the **Object Type** filter, and phases have been defined for that request type, the phases are shown as a gray background against the lifecycle states to which they apply.

When an item type is selected in the **Object Type** filter, and build stages have been configured for Dimensions Build for that item type, the build stages are shown as a gray background against the lifecycle states to which they apply.

If a lifecycle state has been defined as sensitive, this is indicated by the following icon:

You can create a new transition by clicking in one state and dragging the mouse pointer into another state. This displays New Transition dialog box.

Lifecycle Model with a State Selected



Defining CM Rules

Purpose Follow this procedure when you want to define the change management rules for controlling the development of request types for a particular product.

PRIVILEGES Manage Object Types

Constraints This function is available for requests only.

A request cannot be related to another request if rules are in force for one of their types and not for the other.

To define CM rules for a Request Type:

1. From the Lifecycles main window, click the top level node: 

in the navigation pane.
2. Click the name link of the lifecycle associated with the request type in the content pane. The Edit Lifecycle dialog box appears.
3. Select the request type for which you want to define the rules in the filter bar.
4. Do one of the following:
   - Click the state for which you want to define the rule in the Lifecycle Model
   - Choose the state from the list in the filter bar.
5. Click the Phases: 

button. This will display the Edit CM Rules dialog box.
6. To enforce CM rules, select Yes for the **Enable CM Rules** field.
7. Map lifecycle states to phases in the **Lifecycle Phase Rules** section, and set the **Dependent Request Rule**.
8. When you have finished, click OK.

Deleting a Normal State

Purpose Follow this procedure when you want to delete a normal state from a lifecycle.

Note that you do not need to delete off normal states as they are removed when there are no transitions that involve them.

PRIVILEGES Manage Lifecycles

Constraints See About Lifecycle Management on page 178.

To delete a normal lifecycle state:

1. In the Lifecycles main window, display the list of lifecycles in the content pane by clicking the top node: 

in the navigation pane.
2. Click the name link of the lifecycle associated with the object type in the content pane. The Edit Lifecycle dialog box appears.
3. Select the state you want to delete in the upper row of the Lifecycle Model.
4. Click the Delete button: 

You will be prompted with a dialog box asking you to confirm the deletion.

5 Click Yes to delete the state.

Renaming a State

Purpose Follow this procedure when you want to change the name of a state for a lifecycle.

PRIVILEGES Manage Lifecycles

Constraints See About Lifecycle Management on page 178.

To rename a lifecycle state:

1 In the Lifecycles main window, display the list of lifecycles in the content pane by clicking the top node: Lifecycles in the navigation pane.

2 Click the name link of the lifecycle in the content pane. The Edit Lifecycle dialog box appears.

3 Select the state you want to rename in the Lifecycle Model.

4 Click the Rename button: Rename

5 Enter the name in the **New Name** field.

6 Click OK to confirm the change.

Setting the State Properties for an Object Type

Purpose Follow this procedure when you want to define a lifecycle state for an item or request type as sensitive. This means that an electronic signature is required by presenting the user with an Authentication Point dialog box when they action the object type to or from this lifecycle state.

Also follow this procedure to associate an item type with a stage in the Global Stage Lifecycle.

PRIVILEGES Manage Object Types

Constraints This function is available for items and requests only.

You can only assign a deployment stage to a state if no other deployment stage is assigned to it.

You cannot assign or unassign a deployment stage other than DEVELOPMENT to or from a lifecycle state if there are item revisions in the base database that:

- have item types following this lifecycle

  and

- are currently at this lifecycle state

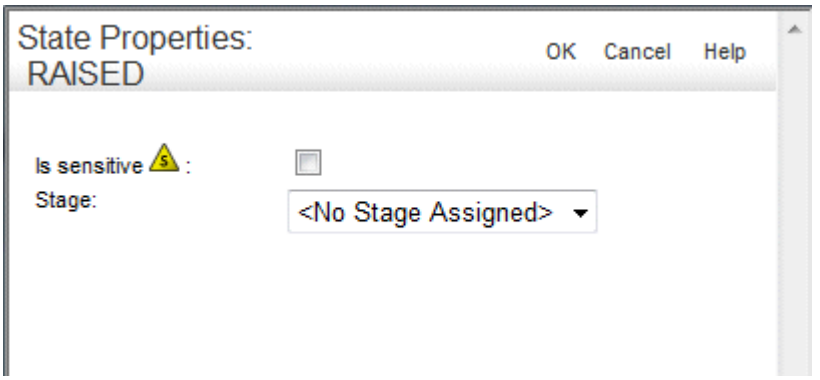
To set the state properties for an item or request type:

1 In the Lifecycles main window, display the list of lifecycles in the content pane by clicking the top node: Lifecycles in the navigation pane.

2 Click the name link of the lifecycle associated with the object type in the content pane. The Edit Lifecycle dialog box appears.

3 Select the item or request type whose state properties you want to set in the filter bar.

4 Select the state for which you want to set the properties and click the Properties 



5 If you want to define the lifecycle state as sensitive, select *Is sensitive*.

6 For an item, request, or baseline type, select the stage you want to associate with the lifecycle state from the **Stage** list. If you want to remove the association, select *<No Stage Assigned>*.

7 Click OK.

NOTE If you have changed the *Is Sensitive* option, you will be presented with an Authentication Point dialog box. Enter your Dimensions password and click OK.

Managing Transitions

About Transitions

A transition is a pair of lifecycle states between which it is possible to action the object. For each possible transition, you specify one or more roles. A user must have one of these roles in order to action the object from the *from* state to the *to* state.

In addition to this, there are two options associated with each role for a transition:

- **Optional.** If this option is set, this means that actioning an object to the transition's from state does not require there to be a user holding that role.

  If this option is not set, the actioning will be disallowed if there are no users holding that role.

- **Pending.** If this option is set, the object will appear in the user's inbox, and the user will receive an email notification when the object is actioned to the from state for the transition.

  If this option is not set, it will not appear in the user's inbox and the user will not receive email notification, but the user will still be able to perform the action.

For further details and an example, see "Assigning Pending and Optional Roles" on page 52.

Transitions Area

The Transitions area of the Edit Lifecycle dialog box displays the possible transitions in or out of the state you have currently selected in the Lifecycle Model. If you have selected a specific object type in the Filter bar, it also lists each attribute for which a role has a defined rule. If no state has been defined, it displays all transactions for all states of the lifecycle.

It consists of:

- A toolbar
- A table of transitions and roles.

The toolbar contains buttons to perform the following actions.

Button	Function
	Allows you to add a new transition using the New Transition dialog box.
	Allows you to edit a transition using the Edit Transition Lifecycle dialog box.
	Allows you to delete one or more transitions.

The table of transitions and roles contains a row for each combination of transition with each role that is authorized to action the object type(s) through this state. If a role is selected in the role filter, only that role will appear in the list.

Next to each role, there is an icon indicating whether the role is optional or pending.

Icon	Meaning
O	Optional. This means that actioning an object to the transition's *from* state does not require there to be a user holding that role. If the role is **not** optional, the actioning will be disallowed if there are no users holding that role.
P	Pending. The object will appear in the user's inbox and the user will receive email notification when the object is actioned to the *from* state for the transition. If the role is **not** pending, it will not appear in the user's inbox and the user will not receive email notification, but the user will still be able to perform the action.

Creating a New Transition

Purpose Follow this procedure when you want to define a new transition for a lifecycle state and its associated roles.

PRIVILEGES Manage Lifecycles

To create a new transition:

1. In the Lifecycles main window, display the list of lifecycles in the content pane by clicking the top node: Lifecycles in the navigation pane.
2. Click the name link of the lifecycle in the content pane. The Edit Lifecycle dialog box appears.
3. Do one of the following:
   - Optionally, select the state to which you want to add the transition in the Lifecycle Model and click the New Transition button:
   - In the Lifecycle Model, click the state from which you want to create the new transition, drag the mouse pointer into the state to which the transition is to go, and release the mouse button.

   The New Transition dialog box appears.
4. Select the states in the transition in the From State and To State fields. To create new states, enter them in the fields.

   NOTE An existing normal transition can be split by creating a normal transition that inserts a new state within the lifecycle path. If this is the case, the existing transition will be adjusted, but you will be prompted with a message asking you to confirm this before the new transition is created.
5. If the transition is a normal one, select the **Is a Normal Transition** check box.
6. Select the roles for the transition from the Roles list and specify if you want the roles to be optional and/or pending.
7. If you want the dialog box to remain open, select the **Keep Open** check box.
8. Click OK to create the transition.

Editing a Transition

Purpose Follow this procedure when you want to edit a transition for a lifecycle state.

PRIVILEGES Manage Lifecycles

Constraints See About Lifecycle Management on page 178.

To edit a transition:

1. In the Lifecycles main window, display the list of lifecycles in the content pane by clicking the top node: Lifecycles in the navigation pane.
2. Click the name link of the lifecycle in the content pane. The Edit Lifecycle dialog box appears.
3. Select the state for the transition you want to edit in the Lifecycle Model.
4. Select the transition in the Transitions area and click the Edit Transition button:
5. Update the roles on the transition as necessary.
6. Click OK to update the transition.

Deleting Transitions

Purpose Follow this procedure when you want to delete one or more transitions for a lifecycle state.

PRIVILEGES Manage Lifecycles

Constraints See **About Lifecycle Management on page 178.**

To delete transitions:

1. In the Lifecycles main window, display the list of lifecycles in the content pane by clicking the top node: 

in the navigation pane.
2. Click the name link of the lifecycle in the content pane. The Edit Lifecycle dialog box appears.
3. Select the state for the transition you want to delete in the Lifecycle Model.
4. Select the transition(s) you want to delete in the Transitions area and click the Delete Transition button:  You will be prompted with a dialog box asking you to confirm the deletion.
5. Click OK to delete the transition(s). Any off normal states that no longer have any transitions to or from them will also be removed.

Managing Attribute Rules

The Attribute Rules Area

The Attribute Rules area displays the attributes and their rules for the state that is currently selected in the Lifecycle Model. It consists of:

- A Toolbar
- A table of attributes and roles
- A status bar.

The Toolbar contains buttons to perform the following functions:

	Allows you to add a new rule. For details, see Creating a New Attribute Rule on page 197.
	Allows you to edit a rule. For details, see Editing an Attribute Rule on page 198.
	Allows you to delete one or more rules. For details, see Deleting Attribute Rules on page 199.

The table of attributes and roles contains a row for each combination of attribute and role that has a rule. If a role is selected in the role filter, only rows for that role will appear in the table.

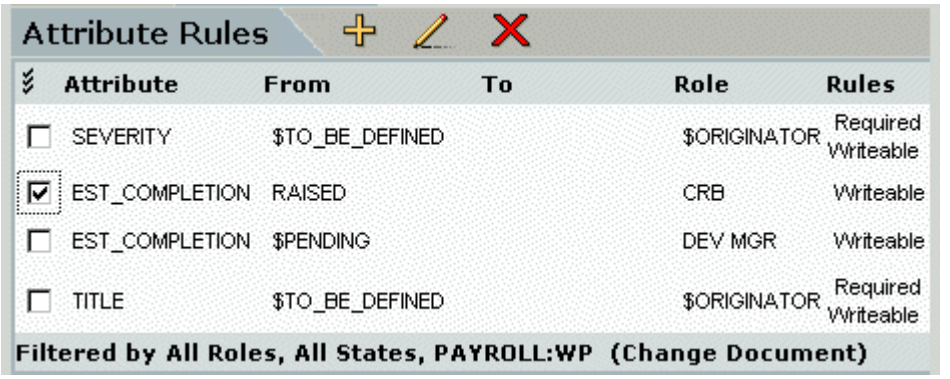
Each row consists of:

- A selection check box to select a rule on which to perform an action

- The name of the attribute
- The From state for which the rule applies
- The To state for which the rule applies
- The role for which the rule applies. If a role is not selected in the role filter, this row is repeated for each role for which there is a rule
- The rules; whether the attribute is required or updatable for the role:

Required	Means required. The attribute must have a value before the object type can be actioned to a new state.
Writeable	Means updatable. The attribute can be updated by this role when the object type is at this state.

The Attribute Rules tab



Creating a New Attribute Rule

Purpose Follow this procedure when you want to create a new attribute rule for a lifecycle state.

PRIVILEGES Manage Object Types

To create a new attribute rule:

1. In the Lifecycles main window, display the list of lifecycles in the content pane by clicking the top node: Lifecycles in the navigation pane.
2. Click the name link of the lifecycle in the content pane. The Edit Lifecycle dialog box appears.
3. Select the Object type for which the rule is to apply in the **Object Type** field in the filter bar.
4. Optionally, select the state to which you want to add the attribute rule in the Lifecycle Model.
5. Click the Add button: The New Attribute Rule dialog box appears.
6. Select the attribute for which you want to create a rule from the Attribute name list.
7. Select a state from the **From state** list. If you want the rule to apply to a transition, select a state from the **To state** list.

NOTE The From state and To state must belong to the same transition. An attribute rule only applies to a single transition.

TIP

- $PENDING: The object is in the users pending list. The actual state is not relevant.
- $TO_BE_DEFINED: The object is checked out (item) or held (request).

8 Select a role from the **Role** list.

9 Select the check boxes as necessary:

- **Required when actioned to the To state**: Specifies that the attribute must have a value before actioning to a new state.
- **Writeable at the From state**: Specifies that the attribute can be updated by this role.
- **Display in this role section**: Specifies that the attribute is to be displayed for this role and state or transition.

NOTE If you are defining a rule for an attribute within a multi-field, multi-value block, you need to ensure that, for a given role section and transition, all of the preceding attributes in the multi-field block will also be visible to the user. This means that for all the preceding attributes in the block, if you have not selected **Required when actioned to the To state** or **Writeable at the From state**, you will need to select **Display in this role section.** This is so that the multi-field multi-value attribute will be displayed correctly in the desktop and web clients.

10 If you want the dialog box to remain open, select the **Keep Open** check box.

11 Click OK to create the attribute rule.

Editing an Attribute Rule

Purpose Follow this procedure when you want to edit an attribute rule for a lifecycle state.

You can create an attribute rule from scratch or you can copy the details from an existing attribute rule.

PRIVILEGES Manage Object Types

To edit an attribute rule:

1 In the Lifecycles main window, display the list of lifecycles in the content pane by clicking the top node:  Lifecycles in the navigation pane.

2 Click the name link of the lifecycle in the content pane. The Edit Lifecycle dialog box appears.

3 Select the Object type for which the rule is to apply in the **Object Type** field in the filter bar.

4 Optionally, select the state for which you want to edit the attribute rule in the Lifecycle Model.

5 Click the Edit button:  The Edit Attribute Rule dialog box appears.

6 Update the fields as necessary.

7 Click OK to update the attribute rule.

Deleting Attribute Rules

Purpose Follow this procedure when you want to delete one or more attribute rules for a lifecycle state.

PRIVILEGES Manage Object Types

To delete an attribute rule:

1. In the Lifecycles main window, display the list of lifecycles in the content pane by clicking the top node: 

in the navigation pane.
2. Click the name link of the lifecycle in the content pane. The Edit Lifecycle dialog box appears.
3. Select the state for which you want to delete the rule in the Lifecycle Model.
4. Select the attribute/rule row(s) you want to delete in the Attribute Rules area and click the Delete button:  . You will be prompted with a dialog box asking you to confirm the deletion.
5. Click Yes to delete the attribute rule(s).

Chapter 10

Baseline and Release Templates

In this Chapter

About Baseline and Release Templates

Purpose

The management of Baseline and Release Templates within the Dimensions® CM Administration Console enables you to define the criteria for the items that are included in baselines or releases.

Managing baseline and release templates enable you to:

- List (query) existing baseline and release templates in the Dimensions base database.
- Create a baseline template to determine the items to be selected for inclusion when creating a baseline.
- Create a release template to determine which items are included in a release and the sub-directories into which they are copied.
- Optionally assign group names to collections of item types which may then be specified for a release template rule.
- Edit or delete existing baseline and release templates.

PRIVILEGES Manage Baseline and Release Templates

Constraints

- The template ID must be unique within the base database.
- Users must have a role on the product in order to restrict the list of item types when defining or editing a baseline or release template. The template itself is **not** product specific.
- A baseline template that has been used to make a release baseline cannot be modified or deleted unless the release baseline has first been deleted.
- A release template cannot be modified or deleted if it is used to make a release.
- An item type group cannot be modified or deleted if it is specified in any release template.

Invocation

Dimensions Administration Console | Product Administration | Baseline and Release Templates

What is a Baseline Template?

A ***Baseline Template*** is a set of rules which determines which Dimensions CM items are included and excluded from a baseline. There are two types of baseline template, item and request.

For an item baseline template, you specify the criteria for inclusion and exclusion of item revisions by defining a set of rules based on the item's type, revision, status or build stage, and item relationships.

For a request baseline template, you specify the criteria for inclusion of item revisions based on a specified group of requests to which the item revisions are related as *In Response To*, or optionally *Info*. You do this by defining a set of rules for the selection of these requests.

Release and **Archive** baselines are defined by their associated templates; whereas a **Design** baseline is essentially defined by the absence of an associated template in that the baseline will include all revisions of the items from the top-level design part and its subordinate design parts (regardless of their status). Baseline templates are unique with respect to the base database; they are not restricted to particular products. Release

baselines only include one revision of each item. Baselines based on a request baseline template are always release baselines.

For a full discussion of the concepts underlying baseline and release templates, please refer to "Baseline Templates" on page 302.

About Item Baseline Template Selection Criteria

An Item baseline template enables you to specify selection criteria for the item types to be included in a baseline. It consists of a default criterion for all item types, and selection criteria for specific item types that override the default for those item types for which they have been defined. The baseline template thus consists of a list, or set of these criteria that determine which item revisions to include.

Selection criteria can either be defined as *implicit*, or can be for a specific normal state or build stage in the lifecycle.

The options for implicit states are:

Description
All revisions
Latest edit revision
Latest edit revision at Final State in Lifecycle
Latest edited revision at the most progressed state
Revision built from selected inputs
Revision that makes selected outputs

NOTE Latest edit revision in this context means the last item revision that was created.

If you choose one of these implicit states as the value for the **Lifecycle State** field, you do not require any additional rules, since these are sufficient to determine which revisions of an item to include. The **State Selector** field can therefore be left blank.

If you specify a specific lifecycle state for the **Lifecycle State** field, an additional status rule will need to be provided in the **State Selector** field. The options for these additional criteria are:

Additional status rule
Latest from state
Most progressed state above specified state OR specified state
Specified state OR most progressed state
Specified state OR next existing state upward
Specified state only

If you specify a specific build stage for the **Lifecycle State** field, an additional stage related rule will need to be provided in the **State Selector** field. The options for these additional criteria are:

Additional build stage rule
Specified build stage and all next existing build stages upward
Specified build stage only.

Dimensions CM uses these additional rules together with a selected normal lifecycle state or build stage to determine a grouping and order of preference for the item revisions to be included in baselines using this template.

Please refer to "Baseline Templates" on page 302 for further details.

About Request Baseline Template Selection Criteria

Request baseline templates enable you to specify rules for selecting requests that will be used as input for creating a baseline from their related items. They comprise one or more rules that are made up from the following:

- Request type.
- Request status.
- Baseline status code, which itself comprises one of the following keys:
  - EQS – specified state only.
  - SUP – specified state and upward.

When a baseline is created specifying a request baseline template and a set of starting parent requests, then all the requests that:

- are related to those parent requests, and
- match the template rules

will be collected together for processing.

The template rules will be processed in exactly the same way they are for item templates, that is, requests will be selected based on the type, status, and the baseline status code that was specified. For example, if a template had a rule that specified:

- all requests of type PR,
- at status ACCEPTED,
- with the baseline status code EQS

then all the requests of type PR, at the status ACCEPTED only, would be used for inclusion into the baseline.

Once this list of requests has been determined, then **only** those items that are related to those requests with either an In Response To or, optionally, an Info relationship will be included in the baseline. However, because the baseline that is being created is a release baseline, only **one** revision of each item will be included in the baseline (not all revisions, as would be the case for a design baseline). This means, that even though the requests being selected may contain multiple revisions of the same item, the final baseline can only contain one revision of all these possible items.

To ensure that only one revision of an item is included in the final baseline in circumstances where multiple item revisions are related to requests, only the latest item revision will be selected using that item's pedigree.

When the baseline has been created, the requests that were used to create it will be related as In Response To that new baseline.

What is a Release Template?

A *Release Template* defines or modifies the rules by which baselined items are selected for inclusion in a release to an operating system directory. The set of rules specifies:

- Which parts of the product structure are to be included.
- Which item types are to be included.
- How the selected items are to be structured in the operating system release directory.

Refer to "Release Templates" on page 316 for further details.

About Release Template Selection Criteria

A release template may consist of any number of selection criteria. Each criterion:

- Optionally specifies a design part. This rule is then applied to that design part and to any part subordinate to it in the design part structure tree, unless and until another criterion specifies a subordinate design part, in which case that other rule overrides for that sub-tree, and so on. Alternatively, a criterion may be applied to ALL design parts of the product.
- Specifies either one or all item types, or a group of item types identified by an item type group you can create for this purpose.
- Optionally specifies a subdirectory of the release directory. The items of all types specified by the criterion, for all design parts selected by the criterion, are placed in the subdirectory if specified, or otherwise in the main release directory.

About Release Template Subdirectories

The purpose of subdirectories is to simplify subsequent handling of the release data. For example, a release template can be used to specify that executable code, source-code modules, user documentation and system-specification documents are each to be grouped in different subdirectories. This field specifies a sub-directory to be *added* to the release directory path name specified by users when they make a release. If a release template **is** specified, the items will be placed in the sub-directory as specified in the release template with the leaf node portion of their project filename; whereas, if a release template is **not** utilized or the **Release Sub-directory** field is left blank, the items' project filenames will be used, relative to the operating system release directory the user requests for the release.

What is an Item Type Group?

An item type group is a set of item types you set up in order to help you specify which ones are to be included in a release baseline. Dimensions CM allows you to assign a group name to a number of otherwise unrelated item types. The group name may then be

specified in a template rule instead of a specific item type, thereby bringing all the associated item types within the scope of that rule.

Baseline and Release Templates Main Window

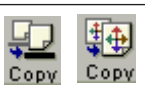
The Baseline and Release Templates main window consists of the following:

- **Menu area**: Displays a toolbar with buttons to carry out various functions. See Baseline and Release Templates Main Window Toolbar on page 206.
- **Navigation pane**: Displays a list of all the baseline templates, release templates, or item type groups that have been defined for the base database depending on the selection you have made. See Baseline and Release Templates Navigation Pane on page 207.
- **Content pane**: If no templates or item type groups are selected, this contains a table summarizing details of all those objects in the navigation pane. If one of those objects is selected in the navigation pane, the content pane displays various details for that object. See Baseline and Release Templates Content Pane on page 207.
- **Status area:** Displays login details and the number of objects selected.For details of the Administration Console status area, see "The Status Area" on page 73.

For the location of the parts of the Administration Console main window, see "The Specific Function Window" on page 71

Baseline and Release Templates Main Window Toolbar

The main window toolbar displays the following buttons. The appearance of the buttons depends on whether baseline templates, release templates, or item type groups have been selected in the navigation pane.

Button	Function
	Allows you to create a new template, or item type group using one of the following: ■ New Baseline Template dialog box ■ New Release Template dialog box ■ New Item Type Group dialog box
	Allows you to create a new template by copying the details from a selected one using one of the following: ■ Copy Baseline Template dialog box ■ Copy Release Template dialog box
	Allows you to delete the selected template, or item type group.

Baseline and Release Templates Navigation Pane

The navigation pane contains the following:

- **Selection:** A selection from which you choose what appears in the Baseline and Release Templates main window. If you made a specific selection under Baseline and Release Templates from the main Dimensions Administration Console, this field will be set to that selection. The selections are:
  - Baseline Templates
  - Release Templates
  - Item Type Groups
- **Base Database Node:** This shows the name of the base database and the type of object selected.
- **List:** A list of all the objects (baseline/release templates or item type groups) as determined by the **List Selection** above. Clicking on the base database node of this list displays a list of all those objects in the content pane. Clicking on a specific object displays more comprehensive details for that object in the content pane.

Baseline and Release Templates Content Pane

When you select a template or item type group in the navigation pane, the content pane displays details for that specific object. These details are grouped into sections which are displayed in the content pane according to the selection made. These details are discussed in:

- "Content Pane for a Single Baseline Template" on page 208.
- "Content Pane for a Single Release Template" on page 209
- "Content Pane for an Item Type Group" on page 211

When the top level node: Baseline Templates is selected in the navigation pane, the content pane displays a table listing all of the baseline templates, release templates, or item type groups in the base database depending on the selection in the navigation pane. It contains the following:

- A button to display the list of templates as a separate HTML page to print or save.
- A button to display the list of templates as comma-separated values to save as a text file.
- Column headings for the fields. Clicking the icon selects or deselects all the objects in the list. You can change the sort order of the list by clicking the column headings.
- A list of objects:

  For baseline and release templates, this contains the following details:

Field	Description
Check Box	A check box to select or deselect the template(s) for the operations performed by the toolbar buttons.
Template ID	The name that identifies the template.
Creation Date	The date the template was created.

Field	Description
Originator	The user ID of the user who created the template
Scope	For a baseline template, whether it is an item or request baseline template.

For item type groups it contains the following details:

Field	Description
Check Box	A check box to select or deselect the template(s) for the operations performed by the toolbar buttons.
Item Type Group Name	The name that identifies the item type group.

Content Pane for a Single Baseline Template

The content pane with a specific baseline template selected consists of:

- A **General** section displaying the following fields:

Field	Description
Name	The name that identifies the template.
Creation Date	The date the template was created.
Created By	The user ID of the user who created the template
Scope	Whether this is an item or request baseline template.

- A **Selection Criteria** section consisting of the following:
  - Toolbar buttons to perform the following functions:

Button	Function
	Allows you to add a selection criterion using the New Selection Criterion dialog box.
	Allows you to edit the selected selection criterion using the Edit Selection Criterion dialog box.
	Allows you to delete a selected selection criterion.

  - A table of details for the selection criteria:
    - For an Item baseline template

Field	Description
Check Box	A check box to select or deselect the criterion for the operations performed by the toolbar buttons.
Item Type	The item type for which the selection applies. "*" means this applies as the default for all item types for which no specific item type criteria have been defined.
Associated Lifecycle	The lifecycle associated with the item type. This is blank if **Item Type** is "*".

Field	Description
Lifecycle State	The state(s) for which the selection criterion applies. This may be either a specific normal state or one of the implicit values.
Additional Criteria	The criteria to be used when a specific lifecycle state has been specified in the Lifecycle State field.

For a further explanation of these fields, see About Item Baseline Template Selection Criteria on page 203.

- For a request baseline template

Field	Description
Check Box	A check box to select or deselect the criterion for the operations performed by the toolbar buttons.
Request Type	The item type for which the selection applies. "*" means this applies as the default for all item types for which no specific item type criteria have been defined.
Request Status	The status on which the selection criterion is based.
Baseline Status Code	The rule that determines which requests are selected in conjunction with the specified status.

For a further explanation of these fields, see "About Request Baseline Template Selection Criteria" on page 204.

- A **Baselines Using Above Template** area, containing a table of baselines created using the selected template. It displays the following for each baseline:

Field	Description
Product Id	The product to which the baseline belongs.
Baseline Id	The ID of the baseline that uses this template.
Creation Date	The date the baseline was created.
Created By	The user ID of the user who created the baseline.

Content Pane for a Single Release Template

The content pane with a specific release template selected consists of:

- A **Release** section displaying the following fields:

Field	Description
Name	The name that identifies the template.
Creation Date	The date the template was created
Created By	The user ID of the user who created the template

- A **Selection Criteria** section consisting of the following:
  - Toolbar buttons to perform the following functions:

Button	Function
	Allows you to add a selection criterion using the New Selection Criterion dialog box.
	Allows you to edit the selected selection criterion using the Edit Selection Criterion dialog box.
	Allows you to delete a selected selection criterion.

  - A table of details for the selection criteria:

Field	Description
Check Box	A check box to select or deselect the criterion for the operations performed by the toolbar buttons.
Part ID	The highest level design part to which the criterion applies. The same criterion applies to all its subordinate child parts, unless overridden by a more specific criterion. "*" means this is the default for all design parts for which no specific item type criteria exist.
Part Variant	This field limits the criterion to one variant of the design parts specified by the **Part ID** field (plus subordinate design part variants). "*" means all variants of the design part will be subjected to the criterion
Item Type/ Group	The item type(s) for which the selection applies. This will either be a specific item type or the name of an item type group that has been defined. "*" means all items will be subjected to the criterion. Hyphen "–" means that the selection criterion will be **not** be applied to any items (i.e. items will not be selected for this criterion).
Release Subdirectory	If specified, all item types selected by this criterion will be placed in this subdirectory.

For a further explanation of these fields, see "What is a Release Template?" on page 205.

- A **Releases Using Above Template** area, containing a table of releases created using the selected template. It displays the following for each release:

Field	Description
Product Id	The product to which the release belongs.
Release Id	The ID of the release that uses this template.
Creation Date	The date the release was created.

Content Pane for an Item Type Group

The content pane with a specific item type group selected consists of:

- An **Item Type Group** section consisting of the following:

Field	Description
Name	The name that identifies the item type group.

- An **Item Types in Above Group** section consisting of the following:
  - Toolbar buttons to perform the following functions:

Button	Function
	Allows you to add an item type to the group.
	Allows you to delete a selected item type from the group.

  - A list of the item types in the group:

Field	Description
Check Box	A check box to select the item type(s) to be deleted from the group.
Name	The name of the item type.

- A **Release Templates Using Above Item Types Group** section displaying the following fields for each release in which the item type group has been used:

Field	Description
Template ID	The name of the release template that uses the item type group.
Created By	The user ID of the user who created the template.
Creation Date	The date the template was created

Creating a New Baseline Template

Purpose Follow this procedure when you want to create a new template to specify which items are included in a baseline. Once you have created the template, you can then define its selection criteria.

PRIVILEGES Manage Baseline and Release Templates

To create a new baseline template:

1 From the Baseline and Release Templates main window, select Baseline Templates from the list in the navigation pane.

2 Do one of the following:

If you want to...	Then...
Create a new template from scratch	Click the New button:  in the main window.
Create a new template based on an existing template	Select the template you want to copy in the navigation pane and click the Copy button:  in the main window.

The New Baseline Template dialog box or Copy Baseline Template dialog box appears.

3 Enter a name for the new template in the **Template ID** field.

4 For a new template, select item or request for the **Scope**.

5 Click OK to create the template.

Once the template has been created, you can subsequently define the selection criteria, as described in Defining the Selection Criteria for a Baseline or Release Template on page 214.

Creating a New Release Template

Purpose Follow this procedure when you want to create a new template to specify which items are included in a release. Once you have created the template, you can then define its selection criteria.

PRIVILEGES Manage Baseline and Release Templates

To create a new release template:

1 From the Baseline and Release Templates main window, select Baseline Templates from the list in the navigation pane.

2 Do one of the following

If you want to...	Then...
Create a new template from scratch	Click the New button:  in the main window.
Create a new template based on an existing template	Select the template you want to copy in the navigation pane and click the Copy button:  in the main window.

The New Release Template dialog box or Copy Release Template dialog box appears.

3 Enter a name for the new template in the **Template ID** field.

4 Click OK to create the template.

Once the template has been created, you can subsequently define the selection criteria, as described in "Defining the Selection Criteria for a Baseline or Release Template" on page 214.

Creating a New Item Type Group

Purpose Follow this procedure when you want to create a new group of item types to use in a release template.

PRIVILEGES Manage Baseline and Release Templates

To create a new item type group:

1. From the Baseline and Release Templates main window, select Item Type Groups from the list in the navigation pane.
2. Click the 

button in the main window. The New Item Type Group dialog box appears.
3. Enter a name for the item type group in the **Item Type Group Name** field.
4. Select the first item type to include in the group from the **Item Type** list. You can include additional item types later.
5. Click OK to create the item type group.

Deleting a Template or Item Type Group

Purpose Follow this procedure when you want to delete a baseline template, release template, or item type group that you created in error or is no longer required.

PRIVILEGES Manage Baseline and Release Templates

To delete a template or item type group:

1. In the Baseline and Release Templates main window, select the object that you want to delete in the navigation pane or the content pane.
2. Click the Delete button: 

or 

This displays a dialog box asking you if you are sure you want to delete the object.
3. Click the 

button. The object disappears from the list.

Specifying the Item Types in an Item Type Group

Purpose Follow this procedure when you want to add to or amend the list of item types to be included in an item type list.

PRIVILEGES Manage Baseline and Release Templates

To specify the item types in an item type group:

1. From the Baseline and Release Templates main window, select Item Type Groups from the list in the navigation pane.
2. Select the item type group in the navigation pane. This will display the details in the content pane.

3 Do one of the following:

If you want to...	Then...
add a new item type to the list	Click the Add: button. The Add Item Type to Item Types Group dialog box appears. Select an item type from the list. If you want the dialog box to remain open to add more item types, select the **Keep Open** check box.
remove an item type from the list	Select the item type in the *Item Types in Above Group* section in the content pane and click the Delete: button. This will display a dialog box asking you to confirm the deletion.

4 Click OK to update the details.

Defining the Selection Criteria for a Baseline or Release Template

Purpose Follow this procedure when you want to add or amend the selection criteria that specify which items are included in a baseline or release.

PRIVILEGES Manage Baseline and Release Templates

To define the selection criteria for a baseline or release template:

1 From the Baseline and Release Templates main window, select the baseline or release template for which you want to define the selection criteria in the navigation pane. This will display the template details in the content pane.

2 Do one of the following:

If you want to...	Then...
add a new selection criterion	Click the Add button: The Add Selection Criterion dialog box appears.
edit an existing selection criterion	Select the criterion in the Selection Criteria section of the content pane and click the Edit button: The Edit Selection Criterion dialog box appears.
delete a selection criterion	Select the criterion in the Selection Criteria section in the content pane and click the Delete button: This will display a dialog box asking you to confirm the deletion.

3 For item baseline templates, do the following:

a From the **Item Type** list, select the item type to which the criteria applies.

b From the Lifecycle State list, select the lifecycle state or build stage to which the criteria applies. Choose an actual state or one of the implicit values. See About Item Baseline Template Selection Criteria on page 203.

c If you chose an actual state or stage from the **Lifecycle State** list, select a value from the **State Selector** list. See About Item Baseline Template Selection Criteria on page 203.

4 For request baseline templates, do the following:

a From the **Request Type** list, select the request type to which the criteria applies.

b From the **Request Status** list, select the lifecycle state to which the criteria applies.

c From the **Baseline Status Code** list choose the option that applies to the Status selected. See "About Request Baseline Template Selection Criteria" on page 204.

5 For release templates, do the following:

a From the **Part Id** list, select the highest-level design part to which the criteria applies.

b In the **Variant** field, enter the design part variant.

c From the **Item Type/Group** list, select an item type or group.

d Optionally, specify a subdirectory name for the items selected by the criteria in the **Release Subdirectory** field.

6 Click OK.

Chapter 11

Upload Rules

In this Chapter

About Upload Rules Management

Purpose

The management of upload rules in the Dimensions® CM Administration Console allows you to:

- Specify rules that determine which files can be added to the Dimensions database and which files should be excluded.
- Determine which item types are assigned to which types of file.
- View, edit, and delete rules for Dimensions clients, products, IDEs, and IDE projects.

PRIVILEGES Manage Project Upload Inclusions/Exclusions

Constraints

You must be the Tool Manager to modify default rules for Dimensions or IDEs. You must be the Product Manager to modify rules for a specific IDE project.

Invocation

Dimensions Administration Console | Configuration Object Management | Upload Rules

NOTE The Upload Rules function in the Dimensions Administration Console was previously known as IDE Setup in releases prior to Dimensions 8.0.

What are Upload Rules?

Upload rules map file name patterns to Dimensions file formats and item types. These rules determine whether files that match a certain file name pattern can be added to the database using a Dimensions client or an IDE. These file name patterns can also restrict a rule to items within a specific project folder. Upload rules must exist in the base database before you can start adding files.

Default upload rules are automatically included when you create a database. The default rules apply to Dimensions clients and each supported IDE. You can modify the default rules, as well as create and modify rules for specific IDE projects and Dimensions CM products.

If you modify the:

- **Default rules for Dimensions Clients:** The modified rules will be used for all files added to Dimensions CM through the desktop client, the web client, and the Project Merge tool.
- **Rules for a specific product for Dimensions Clients:** The modified rules will override the default Dimensions CM rules for all files added to Dimensions through the desktop client, the web client, and the Project Merge tool within that product.
- **Default rules for an IDE:** All subsequent projects created by that IDE in the same base database will use a copy of the new default rules unless there are default product-specific rules for the product to which that project belongs.
- **Default product rules for an IDE:** All subsequent projects created by that IDE in the specified product will use a copy of the new product-specific default rules.
- **Rules for a specific project:** The modified rules will be used for all subsequent operations in that project.

For more information

Upload rules determine the attributes of new items that are created in Dimensions. See About Attributes on page 42 for more information on how attributes work.

Assigning Files of the Same Data Format a Different Item Type

You may want to assign more than one item type for files of the same data format. For example, a team lead wants to have a file to keep notes on the ".project". The file the notes are kept in is a MS Word document and the item type is PN, with a lifecycle of Open and Closed. Other MS Word and `.txt` documents in the project all have the same item type, DOC, and so the same lifecycle, Draft, Review, and Approved.

When adding a single item to a project or stream the item type can be changed.

When adding multiple files of the same data format but different item type, this can be achieved by using filename pattern matching:

- Make sure the project note files contain the same string in their names, for example, are all named `<project>_pn.doc`.
- Update the upload rules for the client and product so that the filename pattern, `_pn.doc`, is assigned the PN item type.

Using Wildcards in Upload Rules

Upload rules support the use of multiple wildcards and Ant style pattern matches in the filenames and paths. Ant patterns help to simplify the organization of separate design part rules. For example, you can define design part rules by a directory pattern match while maintaining the existing item type to file extension mapping. The patterns are similar to those used in DOS and UNIX:

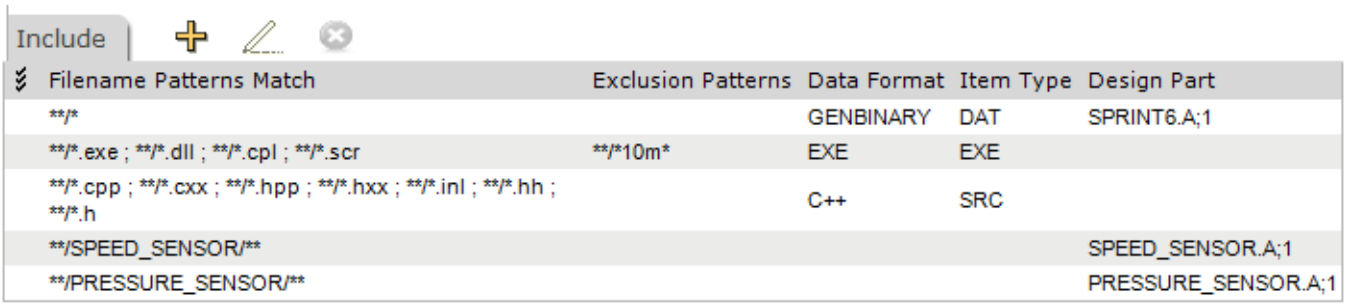
- `'*'` matches zero or more characters.
- `'**'` matches zero or more directories.
- `'?'` matches one character.

Combinations of * and ? are allowed.

For more information see:
http://ant.apache.org/manual/dirtasks.html#patterns

Adding an Independent Design Part Rule

You can add an independent design part rule for a specific directory. In the example below, the pattern `**/SPEED_SENSOR/**` is an independent design part rule and does not define the data format or item type lifecycle as these are not defined in the rule. For the upload to succeed there must be another rule that matches and defines the data format and the item type.



Filename Patterns Match	Exclusion Patterns	Data Format	Item Type	Design Part
**/*		GENBINARY	DAT	SPRINT6.A;1
**/*.exe ; **/*.dll ; **/*.cpl ; **/*.scr	**/*10m*	EXE	EXE	
**/*.cpp ; **/*.cxx ; **/*.hpp ; **/*.hxx ; **/*.inl ; **/*.hh ; **/*.h		C++	SRC	
/SPEED_SENSOR/				SPEED_SENSOR.A;1
/PRESSURE_SENSOR/				PRESSURE_SENSOR.A;1

The file SPEED_SENSOR/x3534.exe activates two upload rules:

- This rule defines the design part of the item:

  ```
  **/SPEED_SENSOR/**  SPEED_SENSOR.A;1
  ```

- This rule defines the EXE data format and the EXE item type lifecycle:

  ```
  **/*.exe ; **/*.dll ; **/*.cpl ; **/*.scr **/*10m* EXE EXE
  ```

Using Multiple Wildcards in a File Pattern

You can use multiple wildcards in a file pattern. This enables you to specify an upload rule for items that share the middle part of a filename, but have a different beginning and end. For example, abc_s_functionality and xyx_s_morefunctionality have filenames with the same middle part: _s_

An administrator adds a new upload rule specifying the file pattern that contains the middle part of a filename, but starts and ends with wildcard characters. For example:

```
*_s_*
```

A developer creates several items where the middle part of the filename is the same. The new upload rule applies to all of the items. Items are created with a common middle part of the filename but with different beginnings and ends.

Combining Multiple Inclusion and Exclusion Patterns

You can combine multiple inclusion and exclusion patterns in a single rule. For example, an administrator changes an existing upload rule and appends another inclusion pattern to match the filename extension expected by the developers. The number of upload rules remains the same. When items are created a rule is activated if there is a match from any pattern from the "Include filename patterns match" list and no matches from any pattern from the "Exclude filename patterns match" list.

Upload Rules Main Window

The Upload Rules main window consists of the following:

- **Menu Area:** Displays toolbar buttons to enable you to carry out various tasks. See Upload Rules Menu Area on page 221.
- **Navigation pane:** Displays a list of IDEs and IDE projects, as well as the Dimensions Default project. See Upload Rules Navigation Pane on page 221.
- **Content pane:** Displays upload rule details for the currently selected IDE, IDE project, Dimensions Default project, or product. See Upload Rules Content Pane on page 221.
- **Status area:** Displays log in details. See The Status Area on page 73.

For the location of the parts of the Administration Console main window, see "The Specific Function Window" on page 71

Upload Rules Menu Area

The menu area displays the following tool bar buttons:

Button	Function
	Create a new set of upload rules for an IDE for a product using the selected upload rules.
	Delete the selected product-specific upload rules for the IDE.

Upload Rules Navigation Pane

The navigation pane displays the following:

- **Base Database:** The current base database for which these upload rules apply.
- **Dimensions Clients:** Represents the default upload rules used by Dimensions when files are added. (This was known as the Project Merge Tool IDE environment in releases prior to Dimensions 8.0.) Beneath this section there is a Default Project folder, representing the default upload rules for the base database, and a folder for each product within the base database representing any product-specific rules.
- ***IDE Names*:** Lists the names of the supported IDEs. Beneath each IDE, the Default Project folder represents the default upload rules for the IDE. Any specific IDE projects are also listed. These folders represent any project-specific rules for the IDE.

Upload Rules Content Pane

When you select the Dimensions default project, a product, an IDE default project, or an IDE specific project, the content pane displays the following sections:

- **General:** Displays product and project information about the current selection.

- **Include:** Displays file inclusions, which determine the files that can be added, checked in, or uploaded to Dimensions. From this section you can add, edit, and delete file inclusions.
- **Exclude**: Displays file exclusions, which determine the files that are not normally added, checked in, or uploaded to Dimensions. From this section you can add, edit, and delete file exclusions.

NOTE You can drag and drop the rules to reorder them.

Defining Upload Rules

Creating Product-Specific Upload Rules for an IDE

Purpose Follow this procedure when you want to create a set of inclusion/exclusion rules for a specific product for an IDE

To create product-specific upload rules:

1. From the Upload Rules main window, in the Navigation pane, expand the node for the IDE for which you want to create product-specific upload rules.
2. Do one of the following:
   - If you want to base the product-specific upload rules on the rules for a specific project, select the node for the project.
   - If you want to base the product-specific upload rules on the default rules for the IDE, select the node for the default project.
3. Click the Create a Default Product Upload Rule button:  . The New Product Upload Rule dialog box appears.
4. From the Product list, select the product whose default rules you want to create.
5. Click OK.

To delete product-specific upload rules:

1. From the Upload Rules main window, expand the node for the IDE for which you want to delete the upload rules for a product.
2. Select the product whose rules you want to delete.
3. Click the Delete a Product Upload Rule button:  in the tool bar, and click **Yes** to confirm that you want to delete the upload rules.

Copying Upload Rules

Purpose Follow this procedure when you want to copy a set of inclusion/exclusion rules from an existing IDE or Client project/product to the current one.

PRIVILEGES Manage Project Upload Inclusions/Exclusions

1. From the Upload Rules main window, select an upload project in the navigation pane:
   - **Dimensions Clients | Default Project** for the Dimensions default rule set.
   - **Dimensions Clients | product name** for the rule set for a specific product.
   - ***IDE Name* | Default Project** for an IDE default rule set.
   - ***IDE Name* | *Project Name*** for an IDE project rule set.
2. In the General section in the content pane, click the Copy button:  . The Copy Inclusions/Exclusions dialog box appears.
3. From the **Source project** list, select the project whose rules you want to copy.
4. Click OK.

Including Files for Upload

Purpose Follow this procedure when you want to add a file inclusion to the upload rules for Dimensions, an IDE, or an IDE project. A file inclusion allows files that match a certain file name pattern to be added, checked in, or uploaded to Dimensions.

PRIVILEGES Manage Project Upload Inclusions/Exclusions

NOTE The order in which the inclusions are displayed in the list is significant. General inclusions should appear above more specific inclusions, ensuring that the specific inclusions are attempted for pattern matching first.

To add a file inclusion:

1. From the Upload Rules main window, select one of the following from the navigation pane:
   - **Dimensions Clients | Default Project** to add the file inclusion to the Dimensions default rule set.
   - **Dimensions Clients | product name** to add the file inclusion to the rule set for a specific product.
   - ***IDE Name* | Default Project** to add the file inclusion to an IDE default rule set.
   - ***IDE Name* | *Project Name*** to add the file inclusion to an IDE project rule set.
2. From the Include section, select an existing inclusion if you want the new inclusion to be added above it in the list. Otherwise, the new inclusion will appear at the bottom of the list.
3. Click the New button:  . The New Upload Inclusion dialog box appears.
4. Enter the file name pattern for the files you want to include in the **File Name Inclusion Patterns Match** field.
5. Optionally, complete the rest of the fields in the dialog box.

6 Click OK.

NOTE Where there is a conflict, the list of files to exclude supersedes the list of files to include.

Excluding Files from Upload

Purpose Follow this procedure when you want to add a file exclusion to the upload rules for Dimensions, an IDE, or an IDE project. A file exclusion prevents files that match a certain file name pattern from being added, checked in, or uploaded to Dimensions.

PRIVILEGES Manage Upload Rules

To add a file exclusion:

1 From the Upload Rules main window, select one of the following from the navigation pane:

- **Dimensions Clients | Default Project** to add a file exclusion to the Dimensions default rule set.
- **Dimensions Clients | *product name*** to add a file exclusion to the rule set for a specific product.
- ***IDE Name* | Default Project** to add a file exclusion to an IDE default rule set.
- ***IDE Name* | *Project Name*** to add a file exclusion to an IDE project rule set.

2 From the Exclude section, click the New button:  . The New Upload Exclusion dialog box appears.

3 Enter the file name pattern of the files you want to exclude from Dimensions in the **File Name Pattern Match** field.

4 Click OK.

Modifying Rules

Purpose Follow this procedure when you want to edit or delete file inclusions or exclusions from an upload rule set.

PRIVILEGES Manage Project Upload Inclusions/Exclusions

To edit or delete an inclusion or exclusion:

1 From the Upload Rules main window, select one of the following from the navigation pane:

- **Dimensions Clients | Default Project** to add an inclusion or exclusion to the Dimensions default rule set.
- **Dimensions Clients | *product name*** to add an inclusion or exclusion to the rule set for a specific product.
- ***IDE Name* | Default Project** to edit or delete an inclusion or exclusion from an IDE default rule set.

- ***IDE Name | Project Name*** to edit or delete an inclusion or exclusion from an IDE project rule set.

2 Select the file inclusion from the Include section that you want to edit or delete, or select the file exclusion from the Exclude section.

You can select more than one inclusion or exclusion to delete.

3 Do one of the following:

- Click the Edit button:  and change the fields as necessary in the Edit Upload Inclusion or Edit Upload Exclusion dialog box.
- Click the Delete button:  and click Yes to confirm that you want to delete the file inclusion(s) or exclusion(s).

Chapter 12

Preservation Policies

In this Chapter

About Preservation Policies

A preservation policy is a set of rules that determine how the output files from a build process are represented in Dimensions CM. A preservation policy is associated with a stage for a project/stream, and each type of target, or output file resulting from a build, is preserved according to these preservation rules when a build is run for the project/stream for that stage.

For further information on Build see the Build User's Guide.

There are three types of preservation rule that determine the type of item revision that can be created for a build output:

- **Normal**: An output file from a build is stored as a normal item revision with the file stored in a Dimensions item library.
- **External**: An output file from a build (such as a listing) that is stored outside a build area. It is represented as an item revision but the item file is not stored in a Dimensions item library. You can, however, get or check out the item revision, in which case the external file is copied to your work area.
- **Placeholder**: An output file from a build that can be inside or outside a build area. It is represented as an item revision but the item file is not stored in a Dimensions item library and you cannot get or check out the item file.

Note that external and placeholder item revisions do not get promoted from one build area to another.

When you define a preservation policy, you specify the default preservation rule for all item types for build outputs by choosing one of these three options. You can then optionally add one or more preservation rules to override the default one for any specific item types that you want to be treated differently.

Once defined, you can associate different preservation policies with different stages of a build project. If you do not specify a preservation policy for a build stage the default is to preserve all build outputs as normal Dimensions items.

Preservation Policies Main Window

The Preservation Policies main window consists of the following:

- **Menu area:** Displays the toolbar buttons to help you carry out various tasks. See Preservation Policies Menu Area on page 229.
- **Navigation pane:** Allows you to view and select from a list of preservation policies. See Preservation Policies Navigation Pane on page 229.
- **Content pane:** Displays a list of valid preservation policies in the base database, or the details of a selected version branch. See Preservation Policies Content Pane on page 229.
- **Status area:** Displays login details and the number of objects selected.For details of the Administration Console status area, see "The Status Area" on page 73.

For the location of the parts of the Administration Console main window, see "The Specific Function Window" on page 71

Preservation Policies Menu Area

The menu area displays the following toolbar buttons:

Button	Function
	Define a new preservation policy using the New Preservation Policy dialog box.
	Create a preservation policy by copying the details of an existing preservation policy.
	Delete the selected preservation policy.

Preservation Policies Navigation Pane

The navigation pane allows you to:

- Select the top-level icon: PAYROLL - Preservation Policies to view a list of all the preservation policies in the product.
- Select a preservation policy icon: to view and edit its details in the content pane.

Preservation Policies Content Pane

When you select a preservation policy in the navigation pane, you view and edit its details in the content pane.

When you select the product icon: PAYROLL - Preservation Policies in the navigation pane, you can do the following in the content pane:

- Select a preservation policy to edit, copy, or delete it. Click to select or deselect all of the preservation policies in the list.
- Print the list of preservation policies by clicking
- Save the list of preservation policies as comma-separated values by clicking
- Sort the preservation policies by clicking the column headings.

When you select a preserving policy icon: in the navigation pane, you can do the following in the content pane:

- Add: edit: or delete: a preservation rule in the **Preservation Rules** area.
- Add: or delete: stages for a project in the **Used By** area.

Managing Preservation Policies

When you create a new preservation policy you define the default rule to be used for the items for the output files. See "Creating a New Preservation Policy" on page 230

You can then:

- Set up one or more preservation rules within the preservation policy for specific item types that override the default rule. See "Adding, Editing, or Deleting Preservation Rules" on page 231.
- Associate this preservation policy with one or more stages for one or more projects. See "Adding or Deleting Project Stages" on page 232

Creating a New Preservation Policy

Purpose Create a preservation policy when you want to determine how outputs from a build are to be preserved in Dimensions. Optionally, you can set up one or more preservation rules within the preservation policy for specific item types that override the default rule. You can subsequently associate this preservation policy with one or more stages in a project.

To create a new preservation policy:

1. In the Build Administration main window, click the Preservation Policies tab.
2. In the status area, select the product to which the preservation policy is to belong.
3. Click the New button: 

   The New Preservation Policy dialog box appears
4. Enter a name and, optionally, a description for the preservation policy.
5. Select the **Default Rule** for the preservation of items.
6. Click OK. The new preservation policy is added to the product.

Editing, Copying, or Deleting Preservation Policies

Purpose Edit a preservation policy when you want to change its description or default rule.

Copy a preservation policy when you want to create a new preservation policy based on an existing policy and its rules.

Delete a preservation policy when it is not required or has been created in error.

Constraints
- When you edit a preservation policy, you cannot change its name.
- You can only delete a project whose status is SUSPENDED.

To edit, copy or delete a preservation policy:

1. Make sure the product to which the preservation policy belongs is selected in the status area
2. In the Build Administration main window, click the Preservation Policies tab and select the preservation policy in the navigation pane.

3 Do one of the following:

If you want to...	Then...
Edit a preservation policy	**1** In the General area, click Edit:  **2** In the Edit Preservation Policy dialog box, change the description or Default Rule, as required. **3** Click OK to confirm the changes.
Copy a preservation policy	**1** Click the Copy button:  **2** In the Copy Preservation Policy dialog box, enter the Name, and optionally, select a new Product, Default Rule, or Description **3** Click OK to add the new preservation policy.
Delete a preservation policy	**1** Click the Delete button:  **2** Click Yes to confirm that you want to delete the preservation policy.

Adding, Editing, or Deleting Preservation Rules

Purpose Add, edit, or delete the preservation rules within a preservation policy when you want to determine how build outputs of specific item types are to be preserved in Dimensions CM.

To edit or delete preservation rules:

1 In the Build Administration main window, click the Preservation Policies tab and select the preservation policy whose rules you want to change in the navigation pane.

2 In the Preservation Rules area, do one of the following:

If you want to...	Then...
Add a rule	**1** Click Add:  **2** In the New Preservation Rule dialog box, select the item type for which you want to add a rule. **3** Select the rule you want to apply to this item type.
Edit a rule	**1** Select the Type whose rule you want to edit. **2** Click Edit:  **3** In the Edit Preservation Rule dialog box, select the rule you want to apply to this item type. **4** Click OK to confirm the changes. **NOTE** You cannot change the type. You must delete the rule for a type and add a new one to change it.
Delete a rule	**1** Select the Type whose rule you want to delete. **2** Click Delete:  **3** Click Yes to confirm that you want to delete the rule.

Adding or Deleting Project Stages

Purpose Add, or delete the project stages assigned to a preservation policy when you want to specify the projects, and the stages for those projects, to which the preservation policy will apply.

To edit or delete project stages:

1 In the Preservation Policies main window, click the Preservation Policies tab and select the preservation policy whose project stages you want to change in the navigation pane.

2 In the Used By section of the content pane, do one of the following:

If you want to...	Then...
Add a project stage	**1** Click Add:  **2** In the Add Project and Stages dialog box, select the project that you want to use this preservation policy. **3** Select the stage(s) for this project for which you want the preservation policy to apply.
Delete a stage for a project	**1** Select the Project and stage you want to delete. **2** Click Delete:  **3** Click **Yes** to confirm that you want to delete the rule.

Chapter 13

Data Formats and MIME Types

In this Chapter

About Defining Data Formats and MIME Types

Purpose

Defining item and request data formats and MIME types and subsequently assigning particular object types to them, is used by Dimensions® CM:

- To compel users to select from a list of defined and assigned formats when creating items of a particular type.
- To determine how the Dimensions desktop client and the Dimensions web client applications display requests or item files.
- To specify MIME types for the Dimensions web client.

If a data format is *not* assigned to a particular item type, you can use *any* format when you create an item, including formats that have not been defined. If a data format is not registered, then it is assumed by Dimensions to be of type BINARY. Binary files do not undergo end of line translation when moving from one operating system to another. Therefore it is important to plan and manage this information.

From the Data Format and MIME Types main window you can:

- List the data formats and Multipurpose Internet Mail Extension (MIME) types currently registered in the base database.
- Filter the list of data formats and MIME types according to user-defined criteria. See "Filtering Data Formats" on page 239.
- Create (register) a new data format and MIME type. See "Creating a New Data Format" on page 239.
- Edit an existing data format and MIME type. See "Editing a Data Format" on page 240.
- Delete a data format and MIME type. See "Deleting a Data Format" on page 240.
- Assign item types to a data format and list the current assignments, see "About Assigning Item or Request Types" on page 240.

PRIVILEGES Manage File Format Definition

Constraints

Only the Tool Manager can define data formats and MIME types, and only a Product Manager can assign them to item types.

About File Classes

What are File Classes?

The File Class specifies the way in which the data of a file is encoded. For example, on Dimensions desktop client this value determines the mode in which item files are transferred between the client and the server. If you do not intend to use this client, you may accept the default value of "ASCII". However, in anticipation of future developments, Dimensions CM includes a number of other values for file classes and it is recommended that you make a relevant choice from the following values:

- ASCII
- BINARY
- MAC (Macintosh)
- VMS
- Windows.

These classes are hard coded into Dimensions CM and cannot be modified. At present, all items that do not have a class of "ASCII" will be transferred between Dimensions CM clients and servers in binary mode.

About MIME Types

What are MIME Types?

A MIME (Multipurpose Internet Mail Extension) type is an internationally defined data format. These types (and subtypes) are allocated by the Internet Assigned Number Authority (IANA) and a fuller description is given below. These values are used by the Dimensions web client to determine how to display a request or item's content within the user's Internet browser. If you do not intend to use Dimensions web client, you may leave this field blank, in which case a default MIME type will be used, but in anticipation of future developments, it is recommended that you make a relevant choice.

MIME types comprise several broad categories, with each category also having subtypes (see About the MIME Subtypes for full details) defined by using a separator. Common examples of MIME types are "text/plain", "text/html", "application/msword" and "application/pdf".

The broad categories are:

text	Used to describe text of various types
multipart	Indicates that the file contains multiple sections with potentially more than one MIME type
message	Used for various types of messages.
application	A catchall description allowing for miscellaneous types of files
image	Used for graphic files
audio	Used for audio files
video	Used for video files
model	Multi-media modeling

To find out current IANA media types, visit the web site

http://www.iana.org/assignments/media-types/

About the MIME Subtypes

The MIME subtypes are subdivisions of the basic categories. A list of the various subtypes is given in the following table.

Type	Subtypes		
text	enriched	plain	html
	richtext	sgml	tab-separated-values
	vnd.fmi.flexstor	vnd.latex-z	
multipart	alternative	appledouble	byteranges
	digest	encrypted	form-data
	header-set	mixed	parallel

Type	Subtypes		
	related	report	signed
	voice-message		
message	external-body	http	news
	partial	rfc822	
application	activemessage	andrew-inset	applefile
	atomicmail	cals-1840	commonground
	cybercash	dca-rft	dec-dx
	eshop	hyperstudio	iges
	mac-binhex40	macwriteii	mathematica
	msword	news-message-id	news-transmission
	octet-stream	oda	pdf
	pgp-encrypted	pgp-keys	pgp-signature
	postscript	prs.alvestrand. titrax-sheet	prs.nprend
	remote-printing	riscos	rtf
	set-payment	set-payment-initiation	set-registration
	set-registration-initiation	sgml	sgml-open-catalog
	slate	vemmi	vnd. $commerce_ battelle
	vnd.businessobjects	vnd.ecdis-update	vnd.enliven
	vnd.fdf	vnd.FloGraphIt	vnd. framemaker
	vnd.hp-HPGL	vnd.hp-PCL	vnd.hp-PCLXL
	vnd.ibm .MiniPay	vnd.intercon .formnet	vnd.intertrust. digibox
	vnd.intertrust .nncp	vnd.japannet-directory-service	vnd.japannet-jpnstore-wakeup
	vnd.japannet-payment-wakeup	vnd.japannet-registration	vnd.japannet-registration-wakeup
	vnd.japannet-setstore-wakeup	vnd.japannet-verification	vnd.japannet-verification-wakeup
	vnd.koan	vnd.lotus-1-2-3	vnd.lotus-approach
	vnd.lotus-freelance	vnd.lotus-organizer	vnd.lotus-screencam
	vnd. lotus-wordpro	vnd.meridian-slingshot	vnd.mif
	vnd.ms-artgalry	vnd.ms-asf	vnd.ms-excel
	vnd.ms-powerpoint	vnd.ms-project	vnd.ms-tnef
	vnd.ms-works	vnd.musician	vnd.music-niff

Type	Subtypes		
	vnd.noblenet-directory	vnd.noblenet-sealer	vnd.noblenet-web
	vnd.osa.netdeploy	vnd.powerbuilder6	vnd.powerbuilder6-s
	vnd.rapid	vnd.seemail	vnd.shana.informed.formdata
	vnd.shana.informed.formtemplate	vnd.shana.informed.interchange	vnd.shana.informed.package
	vnd.street-stream	vnd.svd	vnd.truedoc
	vnd.webturbo	vnd.xara	wita
	wordperfect5.1	x400-bp	zip
image	cgm	g3fax	gif
	ief	jpeg	naplps
	png	tiff	vnd.dwg
	vnd.dxf	vnd.fpx	vnd.net-fpx
	vnd.svf		
audio	32kadpcm	basic	vnd.qcelp
video	mpeg	quicktime	vnd.motorola.video
	vnd.motorola.videop	vnd.vivo	
model	iges	mesh	vrml

Data Formats and MIME Types Main Window

The Data Format and MIME Types main window consists of the following:

- **Menu area**: Contains a toolbar enabling you to carry out various operations on the data formats in the content pane. See Data Formats and MIME Types Menu Area on page 238.
- **Content pane**: Displays a list of data formats and a filter to restrict what appears in the list. See Data Formats and MIME Types Content Pane on page 238.
- **Status area:** Displays log in details. See The Status Area on page 73.

For the location of the parts of the Administration Console main window, see "The Specific Function Window" on page 71

Data Formats and MIME Types Menu Area

The menu area contains a toolbar with buttons to perform operations on the data formats selected in the content pane. These are:

Button	Description
	Create a new data format using the new Data Format dialog box.
	Edit a data format using the Edit Data Format dialog box.
	Delete a data format.

Data Formats and MIME Types Content Pane

The content pane contains a list of all the data formats created for the base database to which you are connected, optionally restricted by the selection you have made in the filter. It consists of the following:

- A page navigator allowing you to select which page of the list to display, or to go to the previous or next page.
- A filter link:  enabling you to filter the list of data formats.
- A  button to display the list of data formats as a separate HTML page to print or save.
- A  button to display the list of data formats as comma-separated values to save as a text file.
- Column headings for the data format fields. Clicking the  icon selects or deselects all the data formats in the list. You can change the sort order of the list by clicking the column headings.
- A table with a row for each data format, containing:
  - A selection check box
  - Data Format
  - Class
  - MIME Type
  - Description.

Working with Data Formats and MIME Types

Working with Data Formats and MIME Types includes the following topics:

- Filtering Data Formats
- Creating a New Data Format
- Editing a Data Format
- Deleting a Data Format

Filtering Data Formats

Purpose Filter data formats when you want to restrict the list in the Data Formats and Mime Types main window to those in which you are interested.

PRIVILEGES Manage File Format Definitions

To filter data formats:

1 In the Data Format and MIME Types main window, click the Filter link: Filter

2 Set up your filtering criteria in the Data Format Filter dialog box.

3 Click OK. The data formats listed in the main window will be filtered on the criteria entered.

To reset the filter:

1 In the Data Format and MIME Types main window, click the Filter link: Filter The Data Format Filter dialog box appears.

2 Click the Reset Filter button. This will clear all the fields in the dialog box.

3 Click OK. The main window will now contain all the data formats in the base database.

Creating a New Data Format

Purpose Follow this procedure when you want to create a new data format to associate with one or more item types.

PRIVILEGES Manage File Format Definitions

To create a new data format:

1 From the Data Format and Mime Types main window, click the New button:  The New Data Format dialog box appears.

2 Enter the name of the new format in the **Data Format** field.

3 Select the class of the item type from the **Class** list.

4 Select the **Compression level**. This is the level of compression to be used when getting item revisions that are assigned this data format. Choose between:

- **No Compression**.
- **Fast Compression** —This performs the compression quickly and using less memory but is the least compressed.
- **Normal Compression**—a normal degree of compression. This is a happy medium between Fast and Best.
- **Best Compression**—This is the slowest but gives the best compression.

5 data format.

6 Enter a valid **MIME** type in the MIME Type field.

7 Optionally, enter a description of the format in the **Description** field.

8 Click OK.

Editing a Data Format

Purpose Follow this procedure when you want to edit the properties of a data format, including the name, description, file class, or MIME type, of a data format.

PRIVILEGES Manage File Format Definitions

To edit a data format:

1. From the Data Format and Mime Types main window, click the name link of the data format you want to edit, or select it and click the Edit button: 
2. Update the fields in the Edit Data Format dialog box as necessary. For details of these fields, see "Creating a New Data Format" on page 239.
3. Click the Used By tab if you want to assign or unassign item types. See Assigning Item or Request Types to a Data Format on page 241 for details.
4. Click OK.

Deleting a Data Format

Purpose Follow this procedure when you want to delete a data format you no longer require, or have created in error.

PRIVILEGES Manage File Format Definitions

To delete one or more data formats:

1. In the Data Format and Mime Types main window, select the data format(s) you want to delete.
2. Click the Delete button:  You will be prompted with a dialog box asking you to confirm the deletion.
3. Click OK.

Assigning Object Types to Data Formats

Assigning Item Types to Data Formats and MIME Types includes the following topics:

About Assigning Item or Request Types

Assign data formats to item types to ensure that items are added in the correct format. For example, C source files are assigned as source code, Word document files as design documents, and so on. The assignment mappings here are in the direction of *item type list -> data format*; that is, a data format is assigned to one or more item types.

NOTE If a data format is *not* assigned to a particular item type, then *any* format can be used at the time of item creation *even ones that have not been defined*.

Assign Data formats to request types to determine how they are displayed in Dimensions CM. Once assigned, the format and MIME type are used by the Dimensions CM client applications (web client, desktop client, and IDE) to correctly choose an application/viewer to display the request.

Assigning Item or Request Types to a Data Format

Purpose Follow this procedure when you want to assign one or more item or request types to a data format.

PRIVILEGES Manage File Format Definitions

To assign item or request types to a data format:

1. From the Data Format and Mime Types main window, click the name link of the data format you want to edit, or select it and click the Edit button: The Edit Data Format dialog box appears showing the General tab.
2. Click the Relationships tab.
3. Click the Assign button:
4. Select **To Item Type** or **To Request Type** according to the type of assignment to want to make.

   The Assign dialog box appears.
5. Select the product that owns the item or request type you want to assign from the **Product Id** list.
6. Select the item or request type you want to assign from the **Item/Request Type Name** list.
7. Click OK. You will be returned to the Edit Data Format dialog box.
8. Repeat steps 3, 4, and 5 for each item or request type you want to assign.
9. When you have finished, click OK.

Unassigning Item Types from a Data Format

Purpose Follow this procedure when you want to remove the assignment of item types from a data format.

PRIVILEGES Manage File Format Definitions

To unassign item or request types from a data format:

1. From the Data Format and Mime Types main window, click the name link of the data format you want to edit, or select it and click the Edit button: The Edit data Format dialog box appears showing the General tab.
2. Click the Used By tab.
3. Select the Item or request type(s) you want to unassign from the data format.
4. Click the Unassign button: The Unassign dialog box appears asking you if you are sure you want to unassign the item or request type.
5. Click Yes. You will be returned to the Edit Data Format dialog box.
6. Repeat steps 3, 4, and 5 for each item or request type you want to unassign.
7. When you have finished, click OK.

Chapter 14

Valid Version Branch Names

In this Chapter

About Version Branch Management

Purpose

The management of version branches in the Dimensions® CM Administration Console allows you to view, define, edit, and delete valid version branch names in a base database.

PRIVILEGES Manage Version Branch Definitions

Constraints

- This function can only be run by the ADMIN group or the Product Manager.
- You cannot delete a version branch name if it is currently used to label any item pedigree trees.

  An option to delete a currently used version branch name is available using the Dimensions Remove Version Branch (RMVB) command with the /FORCE qualifier. Please refer to the *Command-Line Reference Guide* for further details.

Invocation

Dimensions Administration Console | Configuration Object Management | Version Branch Names

For more information

If you have installed the separately licensed Dimensions CM Replicator, refer to the *Dimensions CM System Administration Guide* for branch details specific to Replicator.

What is a Valid Version Branch Name?

A valid version branch name is a permitted branch that users can choose when performing a command that assigns a version to a new item revision. Using the Administration Console, you can define valid version branches, and then make them available to a specific project using the desktop client, the web client, the command-line interface, or the ISPF client. When users add, check out, revise, or edit a revision from that project, they are allowed to choose from the list of available version branches.

In the case of a stream, a single branch name is assigned that is used for all new item revisions in that stream. When you create a new stream, you can either assign an existing branch name or create a new one as a part of the Create Stream process.

The use of version branch names within a project is optional, but a stream must have a branch name.

Version Branch Names Main Window

The Version Branch Names main window consists of the following:

- **Menu area:** Displays the toolbar buttons to help you carry out various tasks. See Version Branch Names Menu Area on page 245.
- **Navigation pane:** Allows you to view and select from a list of valid version branches. See Version Branch Names Navigation Pane on page 245.
- **Content pane:** Displays a list of valid version branch names in the base database, or the details of a selected version branch. See Version Branch Names Content Pane on page 245.
- **Status area:** Displays login details and the number of objects selected.See "The Status Area" on page 73.

For the location of the parts of the Administration Console main window, see "The Specific Function Window" on page 71

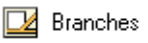
Version Branch Names Menu Area

The menu area displays the following toolbar buttons:

Button	Function
	Define a new version branch name using the New Version Branch dialog box.
	Delete the version branch name(s).

Version Branch Names Navigation Pane

The navigation pane allows you to:

- Select the top-level Branches icon:  to view a list of all the version branches in the base database:
- Select any version branch to view and edit its details in the content pane.

Version Branch Names Content Pane

When you select a version branch name in the navigation pane, you view and edit its details in the content pane.

When you select the Branches icon:  in the navigation pane, you can do the following in the content pane:

- Select a branch name to edit or delete it. Click  to select or deselect all of the version names in the list.
- Print the list of branch names by clicking 
- Save the list of branch names as comma-separated values by clicking 
- Sort the branch names by clicking the column headings.

Defining Version Branch Names

Defining Version Branch Names

Purpose Follow this procedure to define a valid version branch name in the base database. Defining valid version branches enables users to select specific development branches for their item operations within a project.

PRIVILEGES Manage Version Branch Definitions

To define a valid version branch name:

1 From the Version Branch Names window, click the New button: 

The New Version Branch dialog box appears.

2 Enter a name for the branch in the **Version Branch Name** field.

3 Optionally, complete the rest of the fields in the dialog box.

4 Click OK to add the new version branch name.

Editing or Deleting Version Branch Names

Purpose Follow this procedure to modify or remove a valid version branch name from the base database.

PRIVILEGES Manage Version Branch Definitions

Constraint You cannot delete a version branch name if it is currently used to label any item pedigree trees.

To edit or delete a valid version branch name:

Do one of the following from the Version Branch Names window:

- Click a version branch name to edit it, or select a version branch name and click the Edit button: 

  Change the fields in the Edit Version Branch dialog box as necessary.
- Select one or more version branch names and click the Delete button: 

Chapter 15

Area Definitions and the Global Stage Lifecycle

In this Chapter

About Areas and Deployment

Purpose Areas and Deployment allows you to:

- Create and modify work areas, deployment areas, and library cache areas.
- Assign work areas to project/streams.
- Edit the Global State Lifecycle.
- Create and modify area filters.

Invocation Dimensions Administration Console | Distributed Development | Areas and Deployment

About Areas

Areas are locations that are defined in order to contain item files for project/streams when certain item operations are performed. You define a network node and folder for the area so that the files are held under that "root" folder in the same hierarchical folder structure as the items in the project/stream. There are three types of area:

- **Work Area**: An area that is defined for one or more users or groups, so that their file operations such as check in, check out, and so on, automatically use this area to contain the item files for those operations.
- **Deployment Area**: An area that is defined to contain files whose items have reached a particular stage in the lifecycle for a project. This means that item files are moved to that area when the item revisions are deployed to the corresponding stage in the Global Stage Lifecycle.

  Deployment areas can also have a **Filter** specified. A filter is a template file containing a set of rules that determine which files from the project/stream are held in the file area. You can specify that only the items matching a certain filename pattern, or that belong to a certain design part, are transferred to that area.

  You can also specify **Transfer Scripts** for a deployment area. These are scripts that are performed:

  - **Before** files are transferred to the area.
  - **After** files have been transferred to the area.
  - **On Fail**. The script is performed if, for any reason, the transfer of the files has failed to complete successfully.
- **Library Cache Area**: An area that is defined in order to contain copies of files whose items are located on a database on a remote server. The purpose is to improve file get performance. When a user obtains a copy of a file using an operation such as get or check out, Dimensions CM first looks in the library cache area to see whether that file is already there and uses that copy instead of transferring the file from the item library. This makes the processing more efficient if, for example the files are being transferred via the network from another country, and a number of users are accessing them. (The administrator could also set up a batch process to populate the library cache area with the latest file versions overnight, when users are not accessing it.)

  The use of library cache areas is supported for the following clients:

- Desktop client
- DMCLI
- DMPMCLI
- PowerBuilder
- Eclipse plug-in
- .NET integration
- Web client

NOTE A library cache area can be used in the web client provided that the user sets the work area to a work area that has been defined in Dimensions as opposed to a folder location on their local machine. A library cache area is assigned for your user for a given project/stream using the Project/Stream Properties dialog box in the web client or desktop client. If the working location is not set to a Dimensions-defined work area, the library cache area will not be used for item file operations in the web client although it will be used in the desktop client.

Once you have defined a file area you relate it to one or more projects/streams so that it is used to contain item files for those projects/streams

Area Definitions Tab

The Area Definitions tab consists of the following:

- **Menu area:** Displays the program's toolbar to help you carry out various tasks. See Area Definitions Menu Area on page 249.
- **Navigation pane:** Allows you to view and select from the areas in the base database. See Area Definitions Navigation Pane on page 250.
- **Content pane:** Displays details about the selected area(s). See Area Definitions Content Pane on page 250.
- **Status area:** Displays log in details. See The Status Area on page 73.

For the location of the parts of the Administration Console main window, see The Specific Function Window on page 71.

Area Definitions Menu Area

The menu area displays the following toolbar buttons:

Button	Function
	Create a new area using one of the following: ■ New Work Area dialog box ■ New Deployment Area dialog box ■ New Library Cache Area dialog box.

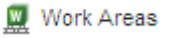
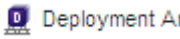
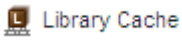
Button	Function
	Delete the selected area(s).
	Edit the Global Stage Lifecycle.

Area Definitions Navigation Pane

The navigation pane displays the areas in the base database. It contains a tree structure in which the second-level nodes represent the types of area:

- Work
- Deployment
- Library Cache.

In the navigation pane, you can:

- Select the top-level icon:  to view a summary of all areas in the database in the content pane.
- Select a second-level node to view a summary of all the areas of that type in the content pane:
  - 
  - 
  - 
- Expand the Deployment Areas node and select the node for a stage in the Global Stage Lifecycle:  to view a summary of the deployment areas associated with that stage.
- Select an area icon:
  - Work area: 
  - Deployment area: 
  - Library Cache area: 

  to view or modify that area's associated details in the content pane.

Area Definitions Content Pane

The information that displays in the content pane varies, depending on what you have selected in the navigation pane. The tables below describe what you'll see in the content pane based on your selection.

Area Definitions Tab: Content Pane

If you select...	Then the content pane displays:
A single Work area:	■ **General:** Section that displays details about the selected Work area. Click the Edit: button to edit the area details and assign users or groups. ■ **Assigned Users and Groups**: Section that displays the users and/or groups that have been assigned to the work area.
A single Deployment area:	■ **General:** Section that displays details about the selected Deployment area. Click the Edit: button to edit the area details. ■ **Global Stage Lifecycle**: A graphical representation of the Global Stage Lifecycle. Click the Edit: button to edit the lifecycle. ■ **Projects**: Section that displays the project/streams that have been related to the deployment area. Click the button to assign a project/stream the area using the Assign Area to Project dialog box. Click the button to unassign a project/stream from the area.
A single library cache area:	■ **General:** Section that displays details about the selected library cache area. Click the Edit: button to edit the area details.
Top-level node: or second-level node: ■ Work Areas ■ Deployment Areas ■ Library Cache Areas or Stage node:	■ **Areas:** Section that displays a detailed summary of all areas in the base database, or of the selected type. ■ Click the Name link to modify details for the selected area. ■ : Click to print the summary of areas. ■ : Click to save the summary of areas as comma-separated values.

Defining Areas

Creating Areas

Purpose Follow this procedure to create a new area.

PRIVILEGES
Create Work Areas for work areas
Create Deployment Areas for deployment areas
Create Library Cache Areas for library cache areas

To create a new area:

1. In Areas and Deployment click **Areas**.
2. From the Areas tab, click the New button:  on the toolbar, and select an option for the type of area you want to create from the list; **Work Area**, **Deployment Area**, or **Library Cache Area**. The New Area dialog box appears.
3. Enter a name for the area in the **ID** field. Optionally enter a **Description**.
4. Select a **Network node** from the list and enter a **Directory** to be used for the area on that node or use the browse button to select it.

   NOTE This is a logical network node that needs to have been defined in the Administration Console Network Administration. See the *System Administration Guide* for details.

   NOTE You cannot assign the same location (the same network node and directory) to more than one area. An error will occur if this location has already been assigned.
5. Enter the **Node user ID** and **Node password** to use for the operating system on the network node. You can also enter a credential set, in which case the **Node password** is not required. For details on credential sets, see the System Administration Guide.

   NOTE This user ID and password need to have been registered using the *dmpasswd* utility. See the *System Administration Guide* for details.
6. If you have selected Deployment Area, select a **Stage ID** to be associated with this area from the list.
7. If you have selected Deployment Area, optionally enter the name of an **Area Filter** used to control the types of item files that are copied to the area.

   CAUTION! Audit and area filters can easily be confused. See "Correct Use of Area and Audit Filters" on page 263.
8. Enter the user ID for the **Area owner**, or accept your current user as the default. The area owner is the user that is able to edit the area details using the Edit Area dialog box.
9. If you have selected Deployment Area, optionally enter the name(s) of any files containing **Transfer Scripts** you want to be performed in relation to the transfer of files to the area, **Pre-event**, **Post-event**, or **Fail-event**.

   For details of configuring and using these scripts, see "Deployment Area Scripts" in the *Developer's Reference*.
10. If you have selected **Deployment Area** or **Library Cache Area**, and you want to exclude the area from all transfer operations at the present time, select **Is this area currently offline?**
11. If you have selected **Work Area**, include the users/groups you want to have the ability to assign the area:
    - To add a user/group, select the name in the **Available Users** list, and click the **>>** link to move it to the **Assigned Users** list

- To remove a user/group, select the name in the **Assigned Users** list, and click the **<<** link.

12 Click **OK** to create the new area.

NOTE When assigning the area to a project/stream in the GUI clients, you must specify the name of the area and not the path.

Modifying Areas

Purpose Follow this procedure when you want to edit the details of an area.

To edit an area:

PRIVILEGES If you are not the area owner, the privileges required are:
Update Work Areas for work areas
Update Deployment Areas for deployment areas
Update Library Cache Areas for library cache areas

1 In Areas and Deployment click **Areas**.

2 On the Area Definitions tab, in the navigation pane, select a area.

3 In the content pane, in the **General** section, click the Edit button:  . The Edit Work Area, Edit Deployment Area, or Edit Library Cache Area dialog box appears.

4 Edit the fields as described in "Creating Areas" on page 251.

5 Click **OK** to commit the changes.

To delete an area:

PRIVILEGES If you are not the area owner, the privileges required are:
Delete Work Areas for work areas
Delete Deployment Areas for deployment areas
Delete Library Cache Areas for library cache areas

1 On the Area Definitions tab, select the top-level node:  or an area type node: in the navigation pane.

2 Select the area in the navigation pane and Click the Delete button:  .

3 Click **Yes** to delete the area.

Assigning Areas to Project/Streams

Purpose Follow this procedure when you want to assign an area to a project/stream in order to assign that area to item files for that project/stream.

PRIVILEGES
Assign Deployment Areas to Project for deployment areas

To assign an area to a project/stream:

1 On the Area Definitions tab, in the navigation pane, select a deployment area.

2 In the navigation pane, in the **Projects** section, click the Add button:  . The Assign Area to Project dialog box appears.

3 Select a project/stream from the **Projects** list.

4 If you want to use a folder that is offset, or located in another folder relative to the Area's directory, enter the relative path to be used in **File path relative to Area Directory**.

5 Select **Deploy by default** if you want files to automatically be deployed to this area when the item revisions are promoted or demoted to this stage.

6 Select an **Audit filter** from the list, or leave this as *Default* to use the default filter for the area.

CAUTION! Audit and area filters can easily be confused. For information see "Correct Use of Area and Audit Filters" on page 263.

7 Enter a **Sequence order** if you want deployments to this area to occur in a particular order relative to other deployment areas when there is more that one deployment area for this project/stream. See the Deployment Guide for the rules for deployment sequences.

8 If you want the area to be populated with item files as soon as you relate the project/stream, select **Populate Area with Project contents**.

9 Click OK to commit the changes.

To edit an area assignment:

1 On the Area Definitions tab, in the navigation pane, select a deployment area.

2 In the navigation pane, in the **Projects** section, select a project and click the  button. The Edit Assignment dialog box appears.

3 If you want to use a folder that is offset, or located in another folder relative to the Area's directory, enter the relative path to be used in **File path relative to Area Directory**.

4 If you want the area to be populated with item files as soon as you relate the project/stream, select **Populate Area with Project contents**.

5 If you want the item files to remain where they are if the relative path changes, select **Do not delete files from old relative path if relative path changes**.

To unassign an area from a project/stream:

1 On the Area Definitions tab, in the navigation pane, select a deployment area.

2 In the navigation pane, in the **Projects** section, select a project/stream and click the button. The Unassign Project from Area dialog box appears.

3 Click **Yes** to unassign the area.

About the Global Stage Lifecycle

The Global Stage Lifecycle

The Global Stage Lifecycle (GSL) is the lifecycle that items follow that controls which versions are included in configurations and builds of the project. Item revisions are moved to the next stage in this lifecycle when they have reached the appropriate stage of approval (a process called *deployment*). This lifecycle is defined for the base database.

Deployment areas for a project can be associated with these stages so that item files are copied to those areas when they are deployed to the corresponding stage.

If you are using Deployment Automation (DA) you can map GSL stages to DA pipelines and environments. For information about using DA see the *Deployment Guide*.

The Global Stage Lifecycle Tab

The Global Stage Lifecycle tab allows you to:

- Rename a stage
- Delete a stage
- Import, export, or delete an associated lifecycle image
- Add a valid transition
- Map stages to DA pipelines and environments
- Delete a transition
- Assign roles to a transition

  NOTE You create a new stage only in the New Transition dialog box when adding a new transition. This is because you cannot add a stage to a lifecycle without there being a possible transition for that stage.

The tab consists of:

- A menu area. This is described in "The Global Stage Lifecycle Menu Area" on page 255.
- An Interactive Lifecycle Model, a diagram which allows you to select and control what appears in the Transitions area. This is described in "The Interactive Lifecycle" on page 256.
- The Transitions area, a list of possible transitions and the roles required for those transitions. This is described in "Transitions Area" on page 259.

The Global Stage Lifecycle Menu Area

The menu area consists of:

- A Toolbar containing buttons to perform the following functions:

Button	Function
	Allows you to delete a stage, see "Deleting a Stage" on page 258.
	Allows you to rename a stage, see "Modifying a Stage" on page 257.
	Allows you to edit the description of a stage.
	Allows you to import an image file using the Import Lifecycle Image dialog box.

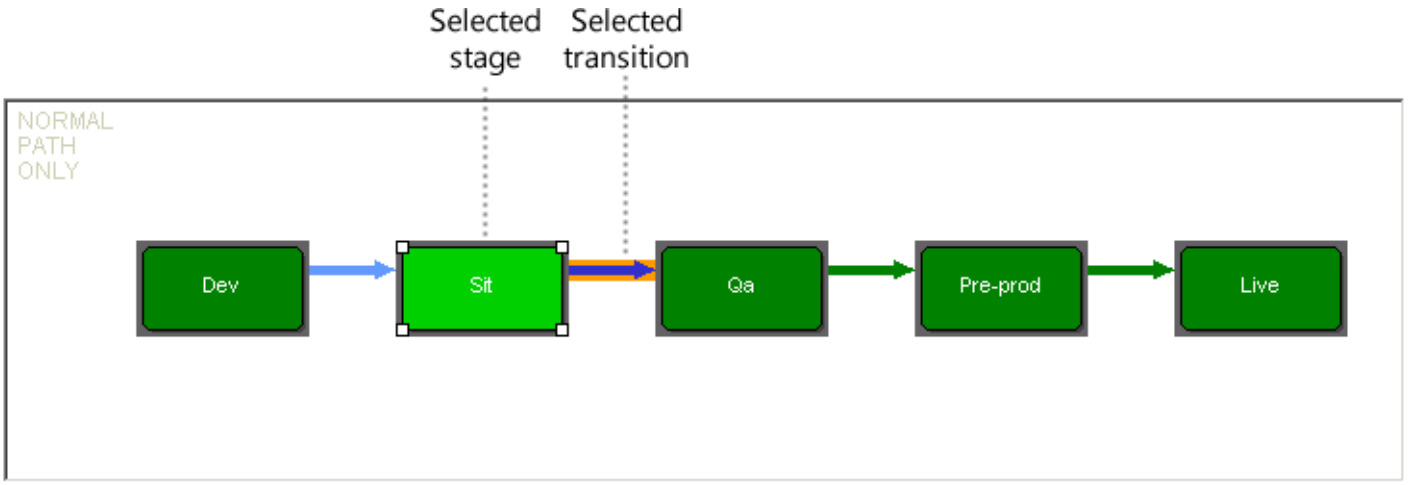
Button	Function
	Allows you to export an image file associated with a lifecycle to your work area.
	Allows you to delete an associated image for the lifecycle.

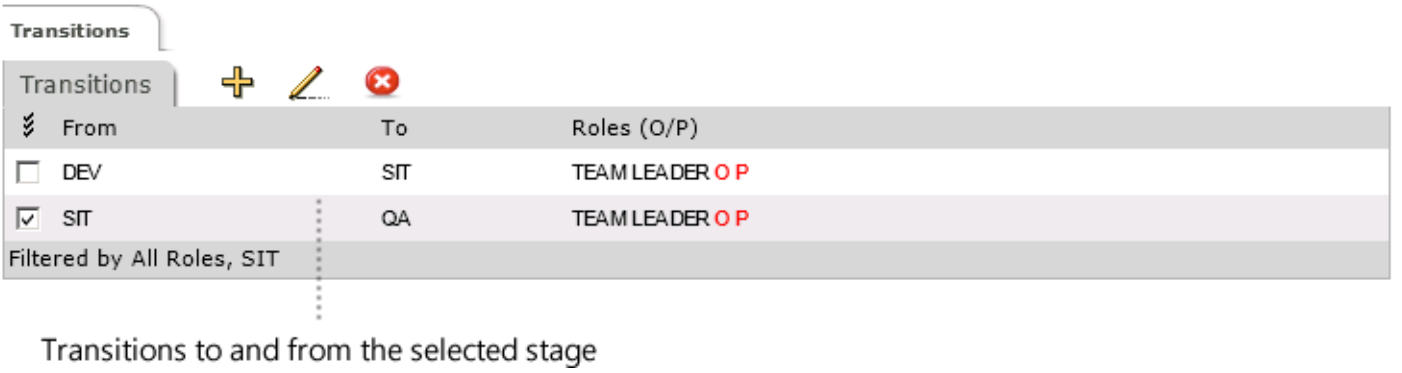
- A filter bar that has filters allowing you to control what appears in the Transitions area:
  - **Role:** When you select a role from the list, all the stages available to that role become highlighted. When you select a stage and a role, only those transitions in or out of that stage that are assigned to the selected role are displayed in the Transitions area.
  - **Stage:** When you select a stage from the list, that stage becomes highlighted in the lifecycle model. It is an alternative to selecting the stage in the lifecycle model. When you select a stage in the lifecycle model this filter updates to reflect your selection.

The Interactive Lifecycle

The interactive lifecycle displays the stages and transition. Select a stage or choose one from the Stage filter. The arrows representing the possible transitions into that stage are highlighted in light blue, and the transitions out of it are highlighted in dark blue. When a stage is selected, the Transitions area only shows the possible transitions to and from that stage. When no stage is selected all the transitions for the lifecycle are shown.

For details about creating a new transition see "Creating a New Transition" on page 260.





Modifying a Stage

Follow these procedure to modify a stage for a lifecycle.

PRIVILEGES Manage Lifecycles (ADMIN_LIFECYCLEMAN)

To rename a lifecycle stage:

1 In Areas and Deployment click **GSL**.

2 Select the stage you want to rename.

3 On the toolbar click **Rename**.

4 Enter the new name and click **OK**.

NOTE Renaming the stage does not update the description.

To change the description of a stage:

1 In Areas and Deployment click **GSL**.

2 Select the stage whose description you want to edit.

3 On the toolbar click **Properties**.

4 Edit the description and click **OK**.

Mapping GSL Stages to Deployment Automation

If you are using Deployment Automation (DA) you can map GSL stages to DA pipelines and environments. For information about using DA see the *Deployment Guide*.

PRIVILEGES Manage Lifecycles (ADMIN_LIFECYCLEMAN)

1 In Areas and Deployment click **GSL**.

2 Select the stage you want to map to DA.

3 On the toolbar click **Properties**.

4 In **Deployment automation to** select one of the following:

- **No Automation**: do not map this stage to DA.
- **Pipeline**: select an DA pipeline. A *pipeline* is a sequence of environments where an application process request is propagated.
- **Environment**: select an DA environment. An *environment* represents logical deployment locations. Your deployment processes must run against at least one environment. Environments and their resources are used by applications and components at runtime.

5 Click **OK**.

Deleting a Stage

Purpose Follow this procedure when you want to delete a stage from the Global Stage Lifecycle.

PRIVILEGES Manage Lifecycles (ADMIN_LIFECYCLEMAN)

To delete a stage:

1 In Areas and Deployment click **GSL**.

2 Select the stage you want to delete.

3 Click **Delete** and confirm.

Importing an Image File

Purpose Follow this procedure when you want to import a new graphic file for the Global Stage Lifecycle.

PRIVILEGES Manage Lifecycles

To import an image file:

1 In Areas and Deployment click **GSL**.

2 Select a lifecycle stage.

3 Click **Import Image** on the toolbar.

4 Browse to an image you want to import.

5 Click **OK**.

Exporting an Image File

Purpose Follow this procedure when you want to export a copy of the associated image file for the Global Stage Lifecycle to your work area.

To export an image file:

1 In Areas and Deployment click **GSL**.

2 Select a lifecycle stage.

3 Click **Export Image** on the toolbar.

4 Right-click the image and save it to a work area.

Deleting an Associated Lifecycle Image

Purpose Follow this procedure when you want to delete an associated image file for the Global Stage Lifecycle.

PRIVILEGES Manage Lifecycles

To delete an associated lifecycle image:

1 In Areas and Deployment click **GSL**.

2 Select a lifecycle stage.

3 Click **Delete Image** on the toolbar.

4 Click **Yes** to confirm the deletion.

Managing Transitions in the Global Stage Lifecycle

Transitions Area

The Transitions area of the Edit Global Stage Lifecycle dialog box displays the possible transitions in or out of the stage you have currently selected in the Lifecycle Model. If no stage has been selected, it displays all transactions for all stages of the lifecycle.

It consists of:

- A toolbar
- A table of transitions and roles.

The toolbar contains buttons to perform the following actions.

Button	Function
	Allows you to add a new transition using the New Transition dialog box.
	Allows you to edit a transition using the Edit Transition Lifecycle dialog box.
	Allows you to delete one or more transitions.

The table of transitions and roles contains a row for each combination of transition with each role that is authorized to promote item revisions through this stage. If a role is selected in the role filter, only that role will appear in the list.

Next to each role, there is an icon indicating whether the role is optional or pending.

Icon	Meaning
O	Optional. This means that promoting an object to the transition's *from* stage does not require there to be a user holding that role. If the role is **not** optional, the promoting will be disallowed if there are no users holding that role.
P	Pending. The object will appear in the user's inbox and the user will receive email notification when the item is promoted to the *from* stage for the transition. If the role is **not** pending, it will not appear in the user's inbox and the user will not receive email notification, but the user will still be able to perform the promotion.

Creating a New Transition

Purpose Follow this procedure when you want to define a new transition for the Global Stage Lifecycle and its associated roles.

PRIVILEGES
Manage Lifecycles to create transitions

To create a new transition:

1 In Areas and Deployment click **GSL**.

2 Do one of the following:

- Select the stage to which you want to add the transition in the Lifecycle Model and click the New Transition button: 
- Select the stage from where you want to start the new transition and drag the mouse to the stage that is the target of the transition.

The New Transition dialog box appears.

3 Select the states in the transition in the From State and To State fields. To create new states, enter them in the fields.

NOTE An existing transition can be split by creating a transition that inserts a new stage within the lifecycle path. If this is the case, the existing transition will be adjusted, but you will be prompted with a message asking you to confirm this before the new transition is created.

4 Select the roles for the transition from the Roles list and specify whether you want the roles to be optional and/or pending.

5 If you want the dialog box to remain open, select the **Keep Open** check box.

6 Click OK to create the transition.

Editing a Transition

Purpose Follow this procedure when you want to edit a transition for a stage in the Global Stage Lifecycle.

PRIVILEGES
Manage Lifecycles to edit transitions

Constraints
- The role titles defined in each lifecycle transition must have been previously defined (except for $ORIGINATOR). See "Adding Roles" on page 90 for details.

To edit a transition:

1 In Areas and Deployment click **GSL**.

2 Select the stage for the transition you want to edit.

3 Select the transition in the **Transitions** area and click the Edit Transition button: 

4 Update the roles on the transition as necessary.

5 Click OK to update the transition.

Deleting Transitions

Purpose Follow this procedure when you want to delete one or more transitions for a lifecycle stage.

PRIVILEGES Manage Lifecycles

To delete transitions:

1 In Areas and Deployment click **GSL**.

2 Select the stage for the transition you want to delete.

3 Select the transition(s) you want to delete in the Transitions area and click the Delete Transition button:  Confirm the deletion.

4 Click OK to delete the transition(s). Any off normal stages that no longer have any transitions to or from them will also be removed.

Managing Area Filters

About Area Filters

Purpose

You define an area filter to specify which types of item file are copied to a deployment or library cache area. For more information about defining areas and associating them with a project/stream, see "About Areas" on page 248. Defining an area filter and associating it with the deployment or library cache area enables you to specify that only certain types of item file are automatically copied to that area.

You can define a series of filename matching patterns and specify that files matching those patterns are either to be included in the filter, or to be excluded from the filter.

The management of Area Filters in the Administration Console enables you to:

- List (query) existing area filters in the Dimensions CM base database.
- Create an area filter in order to add a series of file selection criteria for that filter.
- Add, edit, or delete a file selection criteria for the items to be included in a deployment.
- Add, edit, or delete a file selection criteria for the items to be excluded from a deployment.
- Create a new area filter by copying the details of an existing one.
- Delete an area filter.

You associate an area filter with a deployment or library cache area using the Administration Console Area Definitions option. See "Defining Areas" on page 251.

Audit Filters

You can also use filters that you define to control an audit process. An audit filter selects the files to be fixed that are flagged as errors during an audit operation. For example, if an audit filter has the pattern `*.txt` then only files with the suffix `.txt` " are re-fetched to the area to fix the problems. All other file types are not fixed.

NOTE Audit filters do not alter the list of files to be examined or affect the success or failure of an audit.

You can specify audit filters in several ways:

- On the AUDIT command use the /FILTER qualifier to specify a filter. This only affects this specific audit.
- On the RAWS command use the /FILTER qualifier to attach a filter to a specific project or stream and area. This filter will always be used until it is changed with another RAWS command.
- In the web and desktop clients attach a deployment area to a project or stream and select an audit filter. For details see the section *Assigning Deployment Areas* in the *Managing Deployment* chapter the *Deployment Guide.*

NOTE The filter on the AUDIT command replaces any filter specified on the RAWS command.

Correct Use of Area and Audit Filters

Area and audit filters can easily be confused. The correct use is as follows:

- **Area filters**: used to exclude files from deployment areas and to delete files from areas when they are audited.
- **Audit filters**: used to exclude files from the AUDIT command /FIX process.

CAUTION! Do not attach an audit filter by accident to an area (as an area filter) as this may have unintended consequences. For example, to prevent the audit process from fixing `*.exe` files use an audit filter. However, if by mistake you attach this audit filter as an area filter, the result is that the `*.exe` files are deleted from the area during an audit.

How are the Area Filter Criteria Evaluated?

When Dimensions CM is about to copy item files to a deployment or library cache area, it first checks to see whether there is an area filter associated with it. For a given item revision and area filter, the rules are evaluated according to the following criteria (in order):

1 If the area has no area filter defined, the item revision is transferred into the area.

2 Otherwise, if the item revision matches an exclusion rule, the revision is not transferred into the area.

3 Otherwise, if there are no inclusion rules, the revision is transferred into the area.

4 Otherwise, if the item revision matches an inclusion rule, the revision is transferred into the area.

5 Otherwise, the item revision is not transferred into the area.

NOTE When you assign a project or stream to an area, you specify a folder offset for the location of the files for that project/stream. Any file matching pattern will need to Any file matching pattern will need to take account of the fact that different projects/streams are likely to have different folder offsets relative to the base directory for the area.

About File Pattern Matching

Patterns can be used for the inclusion and exclusion of files. For example:

- '*' matches zero or more characters
- '?' matches one character
- '/' is used to separate folders/directories

Patterns are considered in terms of relative paths, relative to a base directory. So a pattern like `../foo.java` will not match anything when applied since the base directory's parent is never included.

NOTE When you assign a project or stream to an area, you specify a folder offset for the location of the files for that project/stream. Any file matching pattern will need to take account of the fact that different projects/streams are likely to have different folder offsets relative to the base directory for the area.

Examples:

- `*.java` matches `.java`, `x.java` and `Foo.java`, but not `Foo.xml`

- `?.java` matches `Q.java`, `x.java`, but not `.java` or `abc.java`

Combinations of * and ? are allowed.

Matching is done per-directory. First the first directory in the pattern is matched against the first directory in the path to match, then the second directory is matched, and so on. For example, when we have the pattern `?abc/*/*.java` and the path `/xabc/foobar/test.java`, the first `?abc` is matched with `xabc`, then * is matched with `foobar`, and finally `*.java` is matched with `test.java`. They all match, so the path matches the pattern.

It is also possible to match multiple directory levels. This can be used to match a complete directory tree, or a file anywhere in the directory tree. To do this, ** is used as the name of a directory. When ** is used, it matches zero or more directories. For example, `test/**` matches all files and directories under `/test/`, such as `/test/x.java`, or `/test/foo/bar/xyz.html`, but not `/xyz.xml`.

There is one other convention. if a pattern ends with / or \, then ** is assumed to be appended. For example, `mypackage/test/` is interpreted as if it were `mypackage/test/**`.

Further examples:

- `**/xyz/**` Matches all files that have a string `xyz` in their path, including `xyz` as a filename.
- `**/CVS/*` Matches all files in CVS directories that can be located anywhere in the directory tree.

  Matches:

  ```
  CVS/Repository
  org/apache/CVS/Entries
  org/apache/jakarta/tools/ant/CVS/Entries
  ```

  But not:

  ```
  org/apache/CVS/foo/bar/Entries
  ```

- `org/apache/jakarta/**` Matches all files in the org/apache/jakarta directory tree.

  Matches:

  ```
  org/apache/jakarta/tools/ant/docs/index.html
  org/apache/jakarta/test.xml
  ```

  But not:

  ```
  org/apache/xyz.java
  ```

Area Filters Tab

The Area Filters tab consists of the following:

- **Menu area:** Displays the program's toolbar to help you carry out various tasks. See Area Filters Toolbar on page 265.
- **Navigation pane:** Allows you to view and select from the file filters in the base database. See Area Filters Navigation Pane on page 265.
- **Content pane:** Displays details about the selected file filter(s). See Area Filters Content Pane on page 265.
- **Status area:** Displays log in details. See The Status Area on page 73.

For the location of the parts of the Administration Console main window, see "The Specific Function Window" on page 71

Area Filters Toolbar

The toolbar displays the following buttons:

Button	Function
	Create a new file filter using the New Area Filter dialog box
	Allows you to create a new area filter by copying the details from a selected one using one of the Copy Area Filter dialog box.
	Delete the selected area filter(s).

Area Filters Navigation Pane

The navigation pane displays the area filters in the base database. It contains:

- **Base Database Node:**  This shows the name of the base database. Clicking on this node displays a list of all the area filters in the content pane.
- **Area filter nodes:**  Clicking on an area filter node displays more comprehensive details for that area filter in the content pane.

Area Filters Content Pane

The content pane of the Area Filters tab with the base database node  selected contains the following details:

Field	Description
Check Box	A check box to select or deselect the area filter(s) for the operations performed by the toolbar buttons.
Filter ID	The name that identifies the area filter.

Field	Description
Creation Date	The date the filter was created.
Originator	The user ID of the user who created the filter.

The content pane of the Area Filters tab with a specific area filter selected consists of:

- A **General** section displaying the following fields:

Field	Description
Filter ID	The name that identifies the area filter.
Creation Date	The date the filter was created.
Created By	The user ID of the user who created the filter.

- An **Excluded Selection Criteria** section consisting of the following:
  - Toolbar buttons to perform the following functions:

Button	Function
	Allows you to add a selection criterion using the New Selection Criterion dialog box.
	Allows you to edit the selected selection criterion using the Edit Selection Criterion dialog box.
	Allows you to delete a selected selection criterion.

  - A table of details for the excluded selection criteria:
- An **Included Selection Criteria** section consisting of the following:
  - Toolbar buttons to perform the following functions:

Button	Function
	Allows you to add a selection criterion using the New Selection Criterion dialog box.
	Allows you to edit the selected selection criterion using the Edit Selection Criterion dialog box.
	Allows you to delete a selected selection criterion.

  - A table of details for the included selection criteria:

Creating a New Area Filter

Purpose Follow this procedure when you want to create a new area filter to specify which item files are included when items are deployed to a new stage. Once you have created the area filter, you can then define its selection criteria.

To create a new area filter:

1 In Areas and Deployment click **Filters**.

2 Do one of the following:

If you want to...	Then...
Create a new area filter from scratch	Click the New button:  in the main window.
Create a new area filter based on an existing filter	Select the area filter you want to copy in the navigation pane and click the Copy button:  in the main window.

The New Area Filter dialog box or Copy Area Filter dialog box appears.

3 Enter a name for the new template in the **Template ID** field.

4 Click OK to create the template.

Once the template has been created, you can subsequently define the selection criteria, as described in Defining the Selection Criteria for an Area Filter on page 267.

Defining the Selection Criteria for an Area Filter

Purpose Follow this procedure when you want to add or amend the selection criteria that specify which items are included or excluded for an area filter.

To define the selection criteria for an area filter:

1 In Areas and Deployment click **Filters**.

2 Select the area filter for which you want to define the selection criteria in the navigation pane. This will display the template details in the content pane.

3 Do one of the following:

If you want to...	Then...
add a new excluded selection criterion	In the Excluded Selection Criteria section, click the Add button: The Add Selection Criterion dialog box appears.
add a new included selection criterion	In the Included Selection Criteria section, click the Add button: The Add Selection Criterion dialog box appears.
edit an existing selection criterion	Select the criterion in the Excluded Selection Criteria or the Included Selection Criteria section of the content pane and click the Edit button:  The Edit Selection Criterion dialog box appears.
delete a selection criterion	Select the criterion in the Excluded Selection Criteria or the Included Selection Criteria section of the content pane and click the Delete button:  This will display a dialog box asking you to confirm the deletion.

4 If required, enter the file path pattern to use for matching the files that you want to exclude/include in the **File Path Pattern** field. For details, see "About File Pattern Matching" on page 263.

5 If you want to only exclude/include items belonging to a particular design part, enter the design part specification in the **Design Part** field, or use the browse button  to select it.

6 If you want to include the child design parts belonging to the selected **Design Part**, select **Recurse Part Structure**.

7 If required, from the **Data Format list,** select a data format for the items you want to exclude/include.

8 If required, from the **Item Type** list, select an item type for the items you want to exclude/include.

9 Click OK.

Chapter 16

Miscellaneous Database Options

In this Chapter

About IDM Tools Configuration

Purpose IDM Tools Configuration in the Dimensions® CM Administration Console allows you to determine the provider for issues/requests to be used for tracking item updates in Dimensions CM.

PRIVILEGES Manage IDM Tools

Invocation Dimensions Administration Console | Database Management | IDM Tools

What is an IDM Tool?

An IDM (Issue Defect Management) tool is a tool that is used to raise and manage issues, defects, or change requests. Dimensions provides the capability to use the issues or change requests provided from an external IDM tool to track item revisions as an alternative to using internal Dimensions requests.

At the present release, the only type of external request provider supported is SBM. This external request provider is supported in the Visual Studio and Eclipse Integrations, and the desktop client.

An IDM tool instance is a distinctly named definition for a particular installation of the IDM Tool, in this case SBM. The instance is identified by its name and specifies the URL of the Web Interface, the URL for Web Services and a SBM Solution from which issues will be available for use.

In order to use an external request provider, you need to:

1. Create an IDM tool Instance
2. Set this IDM tool instance as the Default.

The use of an external IDM tool is optional, and the default is to use CM requests.

IDM Tools Configuration Main Window

The IDM Tools main window consists of the following:

- **Menu area:** Displays the toolbar buttons to help you carry out various tasks. See IDM Tools Menu Area on page 271.
- **Navigation pane:** Allows you to view and select from a list of IDM tool instances. See IDM Tools Navigation Pane on page 271.
- **Content pane:** Displays a list of IDM tool instances, or the details of a selected IDM tool instance. See IDM Tools Content Pane on page 271.
- **Status area:** Displays login details and the number of objects selected. See "The Status Area" on page 73.

For the location of the parts of the Administration Console main window, see "The Specific Function Window" on page 71

IDM Tools Menu Area

The menu area displays the following toolbar buttons:

Button	Function
	Set an IDM tool instance as the default.
	Create a new IDM tool instance.
	Edit an IDM tool instance.

IDM Tools Navigation Pane

The navigation pane allows you to:

- Select the top-level base database icon:  to view a list of all the IDM Tool types in the base database (currently only SBM).
- Select an icon for an IDM Tool type:  to view a list of all the IDM Tool instances of a particular type.
- Select an icon for an IDM tool instance:  to view and edit its details in the content pane.

IDM Tools Content Pane

When you select a node for an IDM Tool type:  in the navigation pane, you can use the toolbar buttons to perform the following functions:

Button	Function
	Allows you to add an IDM tool instance using the New IDM Tool Instance dialog box.
	Allows you to edit a selected IDM tool instance using the Edit IDM Tool Instance dialog box.
	Allows you to delete a selected IDM tool instance.

When you select an IDM tool instance in the navigation pane, you can view the details, or edit them using the  button.

Configuring IDM Tools

Choosing the Active IDM Provider

Purpose Follow this procedure to determine the use of requests in Dimensions CM clients. You can choose to use:

- Dimensions CM internal requests
- No requests at all
- An external request provider (IDM tool instance).

In order to use an external request provider, you need to:

1 Create an IDM tool Instance. See "Creating an IDM Tool Instance" on page 273.

2 Set this IDM tool instance as the Default. (See below.)

NOTE

If you change the default IDM tool instance or provider, user's working lists and project default requests will be cleared.

When changing the default IDM tool instance or provider, you should make sure that no clients are connected to the server when the change is being made.

PRIVILEGES Manage IDM Tools

To set the active IDM provider:

1 Click the Default button: 

The Set Default IDM Tool Instance dialog box appears.

2 Select a **Default IDM Tool Instance** from the list:

- Select *DIMENSIONS CM* to use Dimensions CM internal requests
- Select *NONE* if you do not want to use requests
- Select the name of an IDM tool instance if you want to use an external IDM provider.

3 Click OK.

If you have changed the default IDM instance then you will see the following warning message:

"Warning: if the default IDM tool instance is changed the working lists and project default requests of ALL users will be erased."

Click OK again to change the default IDM tool instance or press Cancel.

Creating an IDM Tool Instance

Purpose Follow this procedure to create an instance of an IDM tool in the base database. Defining an IDM tool instance enables you to use it to provide issues/requests to relate to item revisions when using the desktop client, Eclipse, or Visual Studio integrations.

PRIVILEGES Manage IDM Tools

To create an IDM tool instance:

1. From the IDM Tools Configuration window, click the New button: 

   The New IDM Tool Instance dialog box appears.
2. Enter a name for the **IDM Tool Instance Name**.
3. Select the **Attributes** tab.
4. For **Web GUI URL**, enter the URL of the GUI for the SBM service. For example:

   http://<hostname>/tmtrack/tmtrack.dll?

   where hostname is the host name of the SBM server.
5. For **Web Services URL**, enter the URL of the Web services application for the SBM service. For example:

   http://<hostname>/gsoap/gsoap_ssl.dll?aewebservices71

   where hostname is the host name of the SBM server.
6. For **Application**, enter the name of the application in SBM that contains the issues to be provided, for example:

   ISSUE_DEFECT_MANAGEMENT
7. Click OK to add the new IDM tool instance.

Editing an IDM Tool Instance

Purpose Follow this procedure to modify or remove an IDM tool instance from the base database.

NOTE

If you change the default IDM tool instance, user's working lists and project default requests will be cleared.

When changing the default IDM tool instance, you should make sure that no clients are connected to the server when the change is being made.

PRIVILEGES Manage IDM Tools

Constraint You cannot delete an IDM tool instance if it is currently used to label any item pedigree trees.

To edit an IDM tool instance:

1. From the IDM Tools Configuration window, expand the tree in the navigation pane and select the node:  for the IDM tool instance.
2. click the Edit button: 

The Edit IDM Tool Instance dialog box appears.

3 If required, edit the name in the **IDM Tool Instance Name**.

4 If required, select the **Attributes** tab and edit the required fields:

- For **Web GUI URL**, edit the URL of the GUI for the IDM service.
- For **Web Services URL**, edit the URL of the Web services application for the IDM service.
- For **Solution**, edit the name of the solution in IDM that contains the issues to be provided.

5 Click OK.

To delete an IDM tool instance:

1 From the IDM Tools Configuration window, expand the tree in the navigation pane and select the node: for the IDM tool type.

2 In the content pane under the IDM Tool Instances tab, select the IDM tool instance.

3 Click the button and click Yes to confirm that you want to delete the instance.

About Database Options

Purpose Database Options in the Dimensions® CM Administration Console allows you to configure various options at the base database level.

Note that at present, the only such option available is Project/Stream Options.

PRIVILEGES Manage Database Options

Invocation Dimensions Administration Console | Database Management | Database Options

What are the Project/Stream Options?

Dimensions CM provides two types of container for the files used in software development, projects and streams. You can set the following options in relation to them:

- **Both projects and streams are active**: This means that users in the base database can create and work with both streams and projects and perform operations on the items contained within them.
- **Only projects are active**: This means that users can only create projects and perform those operations available for projects. Any streams previously created in the database will not be visible.
- **Only streams are active**: This means that users can only create streams and perform those operations available for streams. Any projects previously created in the database will not be visible.

For details of the differences between streams and projects and the operations that can be used with them, see the *User's Guide*.

Database Options Main Window

The IDM Tools main window consists of the following:

- **Menu area:** Displays the toolbar buttons to help you carry out various tasks. See "Database Options Menu Area" on page 275.
- **Navigation pane:** Allows you to view and select from a list of IDM tool instances. See "Database Options Navigation Pane" on page 275.
- **Content pane:** Displays a list of DM tool instances, or the details of a selected DM tool instance. See "Database Options Content Pane" on page 275.
- **Status area:** Displays login details and the number of objects selected. See "The Status Area" on page 73.

For the location of the parts of the Administration Console main window, see "The Specific Function Window" on page 71

Database Options Menu Area

The menu area displays the following toolbar buttons:

Button	Function
	Edit a database option.

Database Options Navigation Pane

The navigation pane allows you to:

- Select the top-level base database icon:  to view a list of all the database options in the base database.
- Select an icon for a specific database option:  to view and edit its details in the content pane.

Database Options Content Pane

The content pane displays the following:

- Database Option: The name of the option
- Value: The current value that is set for this option.

Editing the Project/Stream Options

Purpose Follow this procedure to modify the Project/Stream options for the base database.

PRIVILEGES Update Database Options

To edit the Project/Stream Options:

1. From the Database Options window, expand the tree in the navigation pane and select the node:  for Project/Stream Options.
2. Click the Edit button: 

   The Edit Project/Streams Options dialog box appears.
3. For **Projects/Streams**, select an option from the list:
   - **Both projects and streams are active**
   - **Only projects are active**
   - **Only streams are active**,

   For details of these options, see "What are the Project/Stream Options?" on page 274
4. Click OK.

Part 3

Appendixes

Part 3: Appendixes contains the following appendixes.

Appendix A

Item and Request Templates

In this Appendix

About Item and Request Templates

This appendix contains reference information on item and request format templates in Dimensions® CM. These format templates specify the content and layout for an item or request type.

Use this appendix to determine what variables you can use in a template. Once you've defined a template, you associate it with the appropriate item or request type using the Administration Console *Object Type Definitions* | *Templates* function. You can easily edit the template and store it back in the Dimensions CM using the export/import feature.

Item Format Templates

An *item format template* is a user-defined text file that specifies the content of an item when you create it without any content. If you create an item without specifying an initial filename, Dimensions CM creates the new item and uses the item format template defined for the item type as the initial content.

If you specify a filename at item creation, Dimensions CM does not insert the item header template into the file. To apply the substitution variables, you need to manually insert them at the desired position in the item file before creating the item.

For details of how to specify item header substitution variables, see *Item Header Substitution* in the *User's Guide*.

Request Format Templates

A *request format template* is a user-defined text file that specifies the content and layout of a request when it is browsed or printed. Format templates are required for every request type.

The format template contains substitution variables that are dynamically expanded when you browse a request. Variables can represent both system-defined (Dimensions CM) and user-defined attributes that are single or multi-valued.

A template might include:

- The Dimensions CM identity of the request, including the product ID, request type, and serial number.
- The status of the request.
- The values for one or more user-defined attributes.
- The date of the last update.

You create request templates outside of Dimensions CM, usually using a standard text editor. Though you can use the same template for multiple types, Micro Focus recommends that you create a template for each request type so you can manage each template independently.

After creating a format template, you associate it to a request type. At this point, the format template is placed under revision control by Dimensions CM and is stored in a manner similar to items.

Rules and recommendations

- You can use either lower-case or upper-case text when you define variables in a request format template.
- When using RTF as a request format, any backslash (\) characters included in attributes are removed because the backslash is treated as a qualifier in RTF.

  Also, any word immediately following a backslash without a space is completely removed.

Example of Request Template

Next is an example of a Change Request (CR) request template, followed by an example of an expanded request.

Change Request (CR) Template

<table>
<tr><td colspan="3">CHANGE REQUEST</td></tr>
<tr><td>CR No:%ch_doc_id%
Originator:%originator%</td><td>Create Date:%create_date%
Severity:%severity%</td><td>Status:%current_status%
Customer Name:%customers%</td></tr>
<tr><td colspan="3">Reported by:%cust_contact%</td></tr>
<tr><td>Title:%title%</td><td></td><td></td></tr>
<tr><td><u>Affected Product Details</u>
Platform:%platform%
<u>Main Description:</u>
%description%
%description_end%</td><td colspan="2">Operating System:%op_sys%</td></tr>
<tr><td>Affected Item:
%affected_item%
Affected Parts:
%affected_part%</td><td></td><td></td></tr>
<tr><td><u>Solution Details:</u>
%solution%
Target Release Id:%target_release%</td><td>Expected effort:%exp_effort%</td><td>Actual effort:%actual_effort%</td></tr>
<tr><td><u>Reason for Deferral</u>
%reason_deferral%
Reason for Rejection:
%rejection%</td><td></td><td></td></tr>
<tr><td colspan="3"><u>Action Message</u>
%action%
Action:%action_number% by %user_name% on %date% %time%
%action_text%
%action_end%</td></tr>
</table>

Related Requests %related_doc%		
Action History %action_history%		

Expanded Change Request

```
CHANGE REQUEST


            CR No:PAYROLL_CR_1
            Originator:DMSYS
            Reported by:Mike Corelli
            Create Date:02-APR-2003
            Severity:1_critical
            Status:CLOSED
            Customer Name:Acme Inc


      Title:New field for staff form

            Affected Product Details
            Release:2.0.0
            Platform:Sun Sparc

            Operating System:SunOS 5.3


      Main Description:
 Staff commented that they were unable to state if they
 were ill during a holiday
```

```
Affected Item:
Affected
2     PAYROLL:SICK PAY REPORT.A-SRC;1 (Affected)  DMSYS
      (reports/src/rep_sick.c)
      Report Sick Pay details
Response
4     PAYROLL:SICK PAY REPORT.A-SRC;2 (Response)  DMSYS
      (reports/src/rep_sick.c)
Report Sick Pay details

Affected Parts:
0     PAYROLL:PAYROLL.A;1                         DMSYS
      (PRODUCT)
Payroll Product
2     PAYROLL:REPORTS.A;1                         DMSYS
      (SUB-SYSTEM)
      Offline statistics reports



Solution Details:


Target Release Id:Expected effort:Actual effort:


Reason for Deferral


Reason for Rejection:
```

```
     Action Message

   Related Requests



Action History
1     02-APR-2003 14:15:32                              DMSYS 010
      DMSYS                       DEPT
      Document created
2     02-APR-2003 14:33:37                              DMSYS 010
      DMSYS                       DEPT
      Actioned document from RAISED to ANALYSED
3     02-APR-2003 14:33:41                              DMSYS 010
      DMSYS                       DEPT
      Actioned document from ANALYSED to VERIFIED&CLOSED
4     02-APR-2003 14:42:59                              DMSYS 010
      DMSYS                       DEPT
      Actioned document from VERIFIED&CLOSED to IMPLEMENTED
5     02-APR-2003 14:44:17                              DMSYS 010
      DMSYS                       DEPT
      Actioned document from IMPLEMENTED to VERIFIED&CLOSED
6     02-APR-2003 14:47:20                              DMSYS 010
      DMSYS                       DEPT
      Actioned document from VERIFIED&CLOSED to IMPLEMENTED
7     02-APR-2003 14:48:13                              DMSYS 010
      DMSYS                       DEPT
      Actioned document from IMPLEMENTED to CLOSED
```

Substitution During Operations

When you browse or print a request, Dimensions CM formats the request using the currently defined request template. The request is generated with the current attribute values replacing the substitution variables.

Dimensions CM also allows any request to be retrieved with the attribute values as they were at any previous point that it was actioned. You can do this using the command-line interface BC command with the /ACTION_NO option. See the *Command-Line Reference Guide* for details.

Single-Valued Dimensions CM Variables

These single-valued variables are built into Dimensions CM and may appear anywhere in a request template. The table below contains variables that generally retain the same value throughout the life of the request.

Variable	Description
%ch_doc_id%	The request ID, which consists of the product ID, request type, and serial number. For example: PAYROLL_CR_1
%product%	The ID of the product to which the request belongs.
%base_db%	The base database to which the request belongs.
%dsn%	The database connection string for the Dimensions server to which the request belongs.

Variable	Description
%hostname%	The name of the machine hosting the Dimensions server to which the request belongs.
%type_head%	The description of the request type.
%create_date%	The date when the request was created.
%originator%	The originator's full name.
%department%	The originator's department.
%location%	The originator's site.
%telephone_no%	The originator's phone number.
%originator_id%	The originator's login ID.
%group%	The originator's user group.
%action_date%	The date the request was last actioned.

The remaining variables change frequently during the life of the request.

Variable	Description
%current_status%	The current lifecycle status of the request.
%lifecycle_phase%	The current phase of the request, which is determined by the current status and any applicable rules.
%action_number%	The number of the current (most recent) action.
%update_date%	The date the request was last updated.
%user_name%	The current user's full name.
%user_department%	The current user's department.
%user_location%	The current user' site.
%user_telephone_no%	The current user's phone number.
%user_group%	The current user's group.
%date%	The current date.
%time%	The current time.

Multi-Valued Dimensions CM Variables

These multi-valued variables are built into Dimensions CM and may appear anywhere in a request template. Unlike single-valued variables, they are replaced by a list of entries, each of which may occupy several lines.

Below are descriptions of the multi-valued Dimensions CM variables. For some multi-valued variables, column variables exist that allow you to access information in a particular column. This column information can duplicate the information provided by the multi-valued variables, or provide additional information. Column variables also enable you to have greater control over the formatting of each variable. See "Using Substitution Variable Syntax" on page 292 for help with formatting.

Multi-Valued Variable	Description	Column Variables
%action_history%	A list of all actions performed on the request. Each entry includes a short description of the action, the action number and date, plus the login user name, full name, phone number, and department of the user performing the action.	■ **%ah_action_number%:** The action number. ■ **%ah_date%:** The action date in format DD-MON-YYYY HH:MM:SS. ■ **%ah_status%:** The state to which this request was actioned. ■ **%ah_user_name%:** The full user name. ■ **%ah_user_phone%:** The user's phone number. ■ **%ah_user_dept%:** The user's department. ■ **%ah_action_note%:** The action note. ■ **%ah_user_group%:** The action history user group.
%audit_trail_ failure%	Lists the details of the audit records for failed authentication attempts on a request.	
%audit_trail_ success%	Lists the details of the audit records for successful authentication attempts on a request.	

Multi-Valued Variable	Description	Column Variables
%affected_ baseline%	A list of all baselines currently related to the request. Each entry includes details about the relationship, the current status of the baseline, and the user who created the relationship.	■ **%bln_action_number%:** Request action number when the relationship was created. ■ **%bln_baseline_id%:** The related baseline specification. ■ **%bln_baseline_type%:** The type of the baseline. ■ **%bln_brief_desc%:** A brief description of the baseline. ■ **%bln_creation_method%:** How the baseline was originally created: merged, revised, etc. ■ **%bln_relationship%:** The name of the relationship. ■ **%bln_status%:** The current status of the baseline. ■ **%bln_template%:** The template name that was used to create the baseline. ■ **%bln_user%:** The OS id of the user who created the relationship. ■ **%bln_user_name%:** The full user name of who created the relationship.
%affected_item%	A list of all item revisions currently related to the request. Each entry includes the full item specification, item library filename, the action number when the relationship was created, the type of relationship, and the full name of the user who created the relationship.	■ **%ai_action_number%:** The action number. ■ **%ai_item_id%:** The item's identity. ■ **%ai_user_name%:** The full user name. ■ **%ai_filename%:** The item's filename. ■ **%ai_brief_desc%:** The item's brief description.
%affected_part%	A list of all design parts currently related to the request. Each entry includes the full part specification, the design part category, the action number when the relationship was created, and the full name of the user who created the relationship.	■ **%ap_action_number%:** The action number. ■ **%ap_part_id%:** The part's identity. ■ **%ap_user_name%:** The full user name. ■ **%ap_user%:** The user. ■ **%ap_category%:** The design part category. ■ **%ap_brief_desc%:** The part's brief description.

Multi-Valued Variable	Description	Column Variables
%parent_doc%	A list of all higher-level requests to which this request (child) is currently related. Each entry includes the higher-level request ID, its title (attribute no. 1), the action number when the relationship was created, and the full name of the user who created the relationship.	■ **%pd_relationship%:** The relationship. ■ **%pd_doc_id%:** The document's identity. ■ **%pd_product%:** The document's owning product. ■ **%pd_status%:** The status. ■ **%pd_user_name%:** The full user name. ■ **%pd_user%:** The user. ■ **%pd_brief_desc%:** The brief description.
%related_doc%	A list of all lower-level requests currently related to this request (parent). Each entry includes the lower-level request ID, its title (attribute no. 1), the action number when the relationship was created, and the full name of the user who created the relationship.	■ **%rd_action_number%:** The action number. ■ **%rd_relationship%:** The relationship. ■ **%rd_doc_id%:** The document's identity. ■ **%rd_product%:** The document's owning product. ■ **%rd_status%:** The document's status. ■ **%rd_user_name%:** The full user name. ■ **%rd_user%:** The user. ■ **%rd_brief_desc%:** The document's brief description.
%update_history%	A list of all modifications other than actioning to the request. Each entry includes a short description of the modification, the action number and date, plus the login user name, full name, phone, and department of the user making the modification.	■ **%uh_action_number%:** The action number. ■ **%uh_date%:** The date in format DD-MON-YYYY HH:MM:SS. ■ **%uh_user_name%:** The full user name. ■ **%uh_user%:** The user. ■ **%uh_user_telephone_no%:** The user's telephone number. ■ **%uh_user_dept%:** The user's department. ■ **%uh_user_note%:** The user's note. ■ **%uh_user_group%:** The update history user group.

Multi-Valued Variable	Description	Column Variables
%attribute_update_history%	A list of attribute updates made to the request. Each entry includes the action number and date, plus the login user name, full name, phone, and department of the user making the modification, details of the attribute and the value before and after the change.	■ **%auh_action_number%**: The change document action number indicating when the update was done. ■ **%auh_user_name%**: The full user name of the user who did the update. ■ **%auh_date%**: The date the update was done. ■ **%auh_user%**: The OS name of the user who did the update. ■ **%auh_user_telephone_no%**: The telephone number of the user who did the update. ■ **%auh_user_dept%**: The department of the user who did the update. ■ **%auh_user_group%**: The group of the user who did the update. ■ **%auh_user_note%**: The user note of the user who did the update. ■ **%auh_attrno%**: The attribute number for which the change was made. ■ **%auh_attr_variable%**: The attribute variable for which the change was made. ■ **%auh_old_value%**: The old attribute value. ■ **%auh_new_value%**: The new attribute value.
%user_roles%	A list of the primary, secondary, and leader users for every role in the lifecycle.	
%description%, %description_end%	The detailed description of the request. These two substitution variables must appear on separate, successive lines in a request template.	

Multi-Valued Variable	Description	Column Variables
%action%, %action_text%, %action_end%	These three variables are used to define a subtemplate for action descriptions. The subtemplate is part of the overall request template and contains a list of all recorded action descriptions. This subtemplate starts with the variable **%action%** and ends with the variable **%action_end%**, both appearing on separate lines. Between these, the variable **%action_text%** *must appear on a separate line*, to define where the text of the action description will appear. You can also use the column variables described to the right, as well as the other substitution variables described above (including user-defined variables) in the subtemplate. **NOTE** This subtemplate is **required** for add action description operations to be recorded in the database, *regardless* of whether you want the data to show up in a browse template. **NOTE** The action descriptions are stored separately. If you add variables to the template they do not affect the action descriptions previously entered (the action description substitution variables are not substituted).	■ **%user_name%:** The full name of the user entering the action description. ■ **%user_department%:** The user's department. ■ **%user_location%:** The user's site. ■ **%user_telephone_no%:** The user's phone number. ■ **%date%:** The date of the action description was entered. ■ **%time%:** The time of the action description was entered.
%chdoc_attachments%	A list of files that are attached to the request.	
%primary_user%	A list of users who currently have the request in their inbox and have the primary role capability.	

Multi-Valued Variable	Description	Column Variables
%secondary_ user%	A list of users who currently have the request in their inbox and have the secondary role capability.	
%leader_user%	A list of users who currently have the request in their inbox and have the leader role capability.	

User-Defined Variables

User-defined variables correspond to the user-defined attributes set up for a request type. You can define single-valued attributes as well as multi-valued attributes in list or table form. See "Using Substitution Variable Syntax" on page 292 for information on formatting multi-valued attributes in tables.

Use the following variables to signify a user-defined attribute:

- **%<*variable*-name>%**

  where **<*variable-name*>** is the name of the user-defined attribute that you want to substitute.

- **%PRODUCT_<*variable-name*>%**

  where **<*variable-name*>** is the name of the user-defined product attribute that you want to substitute.

- **%<*variable-name*>_PROMPT%**

  where **<*variable-name*>** is the name of the user-defined attribute that you want to substitute, and **PROMPT** indicates that you want to display the value of the attribute's **Prompt** field.

- **%<*variable-name*>_BLOCKNAME%**

  where **<*variable-name*>** is the name of the user-defined attribute that you want to substitute, and **BLOCKNAME** indicates that you want to display the value of the attribute's block name. This substitution applies to block attributes only.

The example below uses each type of variable. The first row uses the attributes' block name to label the table of attributes. The second row uses the attribute prompt names to label the column headings. The third row substitutes the actual values of the attributes.

```
                          %CUSTOMERS_BLOCKNAME%
                          ---------------------

%CUSTOMER_PROMPT%         %PHONE_PROMPT%         %PRIORITY_PROMPT%
-----------------         --------------         -----------------

%CUSTOMER%                %PHONE%                %PRIORITY%
```

Block attributes

To define block attributes in a table, specify the block name as the table heading, as shown in the example above (%CUSTOMERS_BLOCKNAME%). Each attribute that you want to include in the table must have the same block name. Next, define each attribute name and value as a column heading and column. The order of the attributes must follow

the order of the columns specified in the block attribute, from left to right. See "Using Substitution Variable Syntax" on page 292 for information on how to format columns.

Note that if you update the block attribute name or values, you'll need to update the block in the template.

Displaying Multi-Valued Variables

To define a variable for a multi-valued attribute (user-defined or Dimensions CM-defined), you use the substitution variables or **%<*variable-name*>%** as explained earlier. This displays the values of the attribute in a list, with each of the values appearing on a new line. For example:

```
Attribute name
--------------
Value 1
Value 2
...
Value n
```

You can also display multi-valued attributes as a table. For example:

```
              Table Heading
              -------------
Attr name 1   Attr name 2     Attr name 3
------------  --------------  ------------
Value 1       first value     alpha
Value 2       second value    beta
...           ...             ...
Value n       nth value       nu
```

To define the precise layout of the columns, see "Substitution Variable Syntax," below.

Using Substitution Variable Syntax

For any Dimensions CM or user-defined variables, you can use substitution variable syntax for greater control over formatting. This syntax organizes the attributes into precise columns as an alternative to using manual spacing or tabs in a template as formatting.

The syntax is:

%<*variable-name*>:<c>:<w>:<s>%

where:

- **<c>** is an integer specifying where the column starts from the left of the template, in characters.

- **<w>** is an integer specifying the column width in characters. Specify a positive number to add padding to the right of the field, which ensures that if data doesn't fill the field, the text written to the right of the field will always start at the next column position. To eliminate padding, specify the column width as a negative number. Text following this field will then appear in the next column following the last character of data.
- **<s>** is a single character (upper or lower case) that specifies the column alignment. The available characters are:
  - **L**: Left-justified. Data will be aligned with the left-hand edge of the column. Excess data will be truncated at the column boundary.
  - **R**: Right-justified. Data will be aligned with the right-hand edge of the column. Excess data will be truncated at the right-hand column boundary.
  - **C**: Centered. Data will centered within the column. Excess data will be truncated at the right-hand column boundary.
  - **W**: Wrapped. Data will be aligned with the left-hand edge of the column. Excess data will be wrapped onto successive lines.

For example:

```
                        %CUSTOMERS_BLOCKNAME%
                        --------------------

%CUSTOMER_PROMPT:9:16:L%    %PHONE_PROMPT:27:7:L%       %PRIORITY_PROMPT:40:7:L%
------------------------    --------------------       ------------------------

%CUSTOMER:9:16:L%           %PHONE:27:7:R%              %PRIORITY:40:7:C%
```

Rules and recommendations

- You must either include all of the fields in the substitution variable syntax (**c:w:s**) or none of them.
- Generally, make the starting column equal to or greater than the previous column's starting column value plus its width. This ensures that the columns of data are separated by whitespace.
- We do not recommend using substitution variable syntax (**%<*variable-name*>:<c>:<w>:<s>%**) and plain substitution variables (**%<*variable-name*>%**) on the same line. This can lead to unpredictable formatting results.

  You can use a mixture of both variable types on different lines without any problem.
- Substitution variable syntax is **only** supported for plain text substitution.
- Substitution variable syntax is not supported for proportional fonts because positions are determined by the number of characters you specify.
- Do not use tab characters or insert text to the left or right of substitution variable syntax. These actions may lead to unpredictable formatting results.

Examples

If a template is specified as:

```
....:....1....:....2....:....3....:....4....:....5....:....6
Line prefix:%token1:25:15:L% %token2:42:7:w%%token3:50:9:r%
```

Notice that there is a space character between the "%" characters ending "token1" and beginning "token2", but there is no space between "token2" and "token3". This "extra" space will be forced into the output in some circumstances illustrated below. In practice, it may be better to include this space to make both the template and resulting output slightly more understandable. However, it is recommended that no characters, particularly **not** the "tab" character, or at most a single space character is inserted between the substitution variable declarations within the template file (later examples will illustrate the associated problems). If the data supplied is:

for token1	oneisawidestring, four, seven
for token2	two, fiveisawidestring, eight
for token3	three, six, nine

then the result will be:

```
....:....1....:....2....:....3....:....4....:....5....:....6
Line prefix:            oneisawidestrin  two          three
Line prefix:            four             fiveisa      six
Line prefix:                             widestr
Line prefix:                             ing
Line prefix:            seven            eight        nine
```

Notice that the first value "oneisawidestring" is truncated while "fiveisawidestring" is wrapped. In general, the columns should be specified in such a way that **they do not overlap.** If this rule is broken then the resulting table may not "look" as expected by the user. To illustrate this the template is changed so that the first column overlaps the "line prefix:" text within the template:

```
Line prefix:%token1:9:15:L% %token2:42:7:w%%token3:50:9:r%
```

The resulting substituted text will be:

```
....:....1....:....2....:....3....:....4....:....5....:....6
Line prefix:oneisawides                  two          three
Line prefix:four                         fiveisa      six
Line prefix:                             widestr
Line prefix:                             ing
Line prefix:seven                        eight        nine
```

On each row, the data within the first column has been "pushed" to the right by the fixed text, it should start in character column 9 and be 15 characters wide therefore the last character column in which data is displayed is 23. The first value "oneisawidestring" is truncated at this column. The other data in the first column ("four" and "seven") has also been "pushed" to the right by the fixed text, but because these strings are shorter than the remaining column width they are not truncated.

If we now allow the column specifications to overlap (**not** recommended):

```
Line prefix:%token1:9:15:L% %token2:22:18:w%%token3:35:9:r%
```

Then the result will be:

```
....:....1....:....2....:....3....:....4....:....5....:....6
Line prefix:oneisawides two            thre
Line prefix:four        fiveisawidestri six
Line prefix:seven       eight          nine
```

Remember that, with a positive column width, data within a column is padded with spaces to occupy the full column width specified.

- We know (from the previous example) that the last character column in which the first column of data should appear is 23.
- The padding characters "push" all of the second column of data to the right. The space character in the template (between token1 and token2) appears in column 24, and so all of the second column of data actually starts in character column 25. However, because it was specified to start in column 22 for 18 columns, the last character column in which the second column of data should be printed is 39.
- The first character column in which the third column of data may be printed is 40, similarly its last printing character column is 43.

Thus the strings "three" and "fiveisawidestring" are truncated to "thre" and "fiveisawidestri" respectively. One reason why the column specifications should not overlap is illustrated here, because the second column of data is specified as 18 characters wide and "w" (wrap). Data in this column will be wrapped at 18, 36, 54, etc., but the displayed column is only 15 characters wide, resulting in three characters being truncated from each wrapped line! Do not allow the column specifications to overlap.

Consider the implications of putting a "tab" character in the template between the "token1" and "token2" declarations (again **not** recommended), then the file will appear as follows:

```
....:....1....:....2....:....3....:....4....:....5....:....6
Line prefix:%token1:9:15:L%    %token2:22:18:w%%token3:35:9:r%
```

The "tab" has printed as blank space up to the next multiple of eight, in this case 32, and so the string "token2" starts at character column 33. The resulting output will be:

```
....:....1....:....2....:....3....:....4....:....5....:....6
Line prefix:oneisawides two            thre
Line prefix:four        fiveisawidestri six
Line prefix:seven       eight          nine
```

which is exactly the same as with the space character, because column 24 is also a multiple of eight. If we had put a space followed by a "tab" character in the template file, then when printed out the template file will appear exactly the same as above, but the printed output file would be as follows:

```
....:....1....:....2....:....3....:....4....:....5....:....6
Line prefix:oneisawides        two           thre
```

```
Line prefix:four              fiveisawidestr six
Line prefix:seven             eight         nine
```

The result is **very** confusing because the program has put out two characters (space and "tab") between the substitution variables "token1" and "token2", but this has resulted in the printed columns shifting to character column 33 instead of 26. Consequently the third column of data (three, six and nine) is also displaced by seven characters. You are advised **not** to use "tab" characters in the template file.

Going back to the original template file, containing a single space between the substitution variables "token1" and "token2", if we now change the column width specification for "token1" from "15" to "-15" then it will not force the first column of data to end in character column 23 (the minus sign requests no padding characters).

```
Line prefix:%token1:9:-15:L%  %token2:22:18:w%%token3:35:9:r%
```

The result will be:

```
....:....1....:....2....:....3....:....4....:....5....:....6
Line prefix:oneisawides two            thre
Line prefix:four      fiveisawidestring six
Line prefix:seven     eight            nine
```

It can be seen that the column one data finishes in character column 23, but the second data column should start in character column 22, thus the value "two" is "pushed" to the right, but "fiveisawidestring" and "eight" start in their correct character columns. The space character between the "token1" and "token2" declarations in the template file has been forced in at character column 24 before the string "two". By luck, "fiveisawidestring" has not "pushed" the value "six" because there was space to right-justify the three characters. However had the value "six" been "sixteen", then the result would be:

```
....:....1....:....2....:....3....:....4....:....5....:....6
Line prefix:oneisawides two            thre
Line prefix:four     fiveisawidestring sixt
Line prefix:seven    eight             nine
```

If we change the specification of "token2" width from "18" to "-18" then the result will be:

```
....:....1....:....2....:....3....:....4....:....5....:....6
Line prefix:oneisawides two           three
Line prefix:four     fiveisawidestringsixte
Line prefix:seven    eight             nine
```

There is no space separating the declarations for "token2" and "token3", so the two strings have run together at the data column boundary, but "three" is no longer "pushed" to the right and so this string is all displayed. If we now change the data again so that

"oneisawidestring" is shortened to "oneiswide" and "fiveisawidestring" becomes "fiveiswide" then the result is:

```
....:....1....:....2....:....3....:....4....:....5....:....6
Line prefix:oneiswide two             three
Line prefix:four     fiveiswide     sixteen
Line prefix:seven    eight             nine
```

The point to note here is that even if the columns are specified in such a way that they overlap, the data is only "pushed" to the right by non-white-space characters in the preceding column.

The next six examples illustrate that the user encounters while trying to create a "boxed" table by adding characters into the template. The behavior of any character (or string of characters) between the declaration of the substitution variables in the template file is exactly the same as we have already seen for the space character (between "token1" and "token2"). So if the line in the template was:

```
|%token1:9:-15:L%|%token2:22:-18:w%|%token3:35:9:r%|
```

Then the resulting substituted text is:

```
....:....1....:....2....:....3....:....4....:....5....:....6
|       oneiswide|   two|            three|
|       four|        fiveiswide|   sixteen|
|       seven|       eight|           nine|
```

With the wider strings the result would be:

```
....:....1....:....2....:....3....:....4....:....5....:....6
|       oneisawidestrin|two|          three|
|       four|        fiveisawidestring|sixt|
|       seven|       eight|            nine|
```

Either of which is probably not what was intended. The situation gets even more complex if data within the table is wrapped. The template line:

```
|%token1:9:-15:L%|%token2:22:-7:w%|%token3:35:-9:r%|
```

generates the table:

```
....:....1....:....2....:....3....:....4....:....5....:....6
|       oneisawidestrin|two|          three|
|       four|        fiveisa|       sixteen|
||                  widestr||
||                  ing||
|       seven|       eight|            nine|
```

which is clearly not what was wanted, but it does serve to illustrate what the substitution program will do with characters in between the token declarations. In general, it is recommended that no characters (except perhaps a single space) should be put between the substitution variable declarations in the template file.

If we allow padding characters to be printed then:

```
|%token1:9:15:L%|%token2:22:18:w%|%token3:35:9:r%|
```

results in substituted text:

```
....:....1....:....2....:....3....:....4....:....5....:....6
|       oneisawidestrin|two|           |thr|
|       four           |fiveisawidestri|six|
|       seven          |eight          |nin|
```

The template line:

```
|%token1:9:12:L%|%token2:22:7:w%|%token3:35:9:r%|
```

generates the table:

```
....:....1....:....2....:....3....:....4....:....5....:....6
|       oneisawidest|two    |         three|
|       four        |fiveisa|       sixteen|
|                   |widestr|              |
|                   |ing    |              |
|       seven       |eight  |          nine|
```

and the template line:

```
|%token1:2:18:L%|%token2:22:7:w%|%token3:35:9:r%|
```

generates the table:

```
....:....1....:....2....:....3....:....4....:....5....:....6
|oneisawidestring  | two    |         three|
|four              | fiveisa|       sixteen|
|                  | widestr|              |
|                  | ing    |              |
|       seven      | eight  |          nine|
```

For completeness, the last example has the formatting data removed from the line in the template file, but uses the same data:

```
%token1% %token2% %token3%
```

Note that there are now spaces between the substitution variable declarations, resulting in the following output text:

```
....:....1....:....2....:....3....:....4....:....5....:....6
oneisawidestring two three
four fiveisawidestring sixteen
seven eight nine
```

As expected, there is now no formatting of the data into columns. Substitutions occur one after another across the output line, and are separated by the spaces specified between the substitution variables in the template file.

Appendix B

Baseline and Release Templates

In this Appendix

Baseline Templates

In Chapter 2, "Process Modeling Concepts", the concept of a baseline is introduced in the topic "About Object Classes" on page 30. A **Baseline Template** consists of a set of rules which are used as criteria for inclusion/exclusion of items under Dimensions® CM control into a baseline. There are two types of baseline template. These are described in:

- "Item Baseline Templates" on page 302
- "Request Baseline Templates" on page 314.

Release and **Archive** baselines are defined by their associated templates; whereas a **Design** baseline is essentially defined by the absence of an associated template in that the baseline will include all revisions of the items from the top level design part and its subordinate design parts (regardless of their status). Baseline templates are unique with respect to the base database i.e. they are not restricted to particular products. Baseline templates are defined using the Administration Console *Baseline and Release Templates* function.

Item Baseline Templates

An item baseline template consists of a set of rules for selecting item revisions based on the item's type, revision, status or build stage.

There can be one rule for default item types (specified as item type *); this **default rule** is applied to all relevant-items which are not covered by the other rules. The inclusion of a default rule in a template ensures that all relevant-items receive consideration for inclusion in a baseline.

The simplest baseline template could select the latest revision of all the items in the current project/stream i.e. item type * and **latest edit revision** (*LATEST). This baseline template allows manual selection of item revisions to be included in a baseline.

Baseline rules divide into two classes:

- Rules that **only** operate on a specified normal lifecycle state or build stage. One of these must be chosen for the selected normal lifecycle state or stage.
- Rules that do **not** require a normal lifecycle state to be specified. These correspond to implicit states. These do not apply to build stages.

For brevity in the following pages the shortcut codes shown in the table below are used.

Lifecycle State	Shortcut
Baseline Rules that Operate on a Specified Normal Lifecycle State	
Latest from state	LFS
Most progressed state above specified state or specified state	MPS
Specified state or most progressed state	SMP
Specified state or next existing state upward	SUP
Specified state only	EQS

Baseline Rules that Operate on a Specified Build Stage	
Specified build stage and all next existing build stages upward	BUP
Specified build stage only	EQB
Baseline Rules that do not Require a Normal Lifecycle State to be Specified	
*All revisions	*ALL
*Latest edit revision	*LATEST
*Latest edit revision at final state in lifecycle	*FINAL
*Latest edited revision at the most progressed state	*HIGHEST
*Revision built from selected inputs	*BUILT
*Revision that makes selected outputs	*MADE OF

Rule Operation

Only one rule may be defined for each item type. The rule is used to determine, for relevant-items of that type, what status (lifecycle state) or build stage is acceptable, or to be preferred, for any revision to be included in a baseline. To specify the rule, any one of the states or build stages in the normal lifecycle associated with that type may be chosen, together with one of the rules below, which determines how that state and others in the normal lifecycle are to be grouped and arranged in an order of preference. (A normal lifecycle consists of a single chain of states, starting with the **first state** at the beginning of the chain, and progressing **upward** through **later states** to the **final state** at the end of the chain.)

Then for each relevant item which is of the type concerned within this rule, we look at each grouping of normal lifecycle states in the order of preference indicated by the rule, until we find the first group which is not empty, i.e. where there is at least one revision of the relevant-item in that group. If we then find that there are two or more revisions in that group, we always select the revision which has been created/updated most recently; this is often, but need not necessarily be, the one with the highest entry in the revision field. If all groups in the order of preference are empty, then that item will not be included in any baseline specified using this template.

Note that, except when a ***ALL** rule is used (this is detailed later), no more than one revision of any item is selected for inclusion in a release-baseline.

Rule	Grouping and Order of Preference
Latest from state (LFS)	Latest from specified state and upward: there is just one group – it consists of all revisions which are at **any** of the normal lifecycle states between the specified state and the final state, both of these states inclusive. (Conversely, if a relevant-item of the given type has all its revisions at states which are **not** in the normal lifecycle, and/or at states which *are* in the normal lifecycle but are below (earlier than) the specified state, then such a relevant-item is not selected using this template.)
Most progressed state above specified state or specified state (MPS)	Most progressed state not below specified state: there are several groups, each of which includes revisions (if any) at just one lifecycle state. The order of preference for the groups is then: **1** the group for the final state in the normal lifecycle **2** the group for the next-to-final state then so on downwards, ending with the group for the specified lifecycle state.
Specified state or most progressed state (SMP)	Specified state or most progressed state: there are several groups, each of which includes revisions (if any) at just one lifecycle state. The order of preference for the groups is then: **1** the group for the specified lifecycle state **2** the group for the final state in the normal lifecycle **3** that for the next-to-final state then so on downwards to earlier states, ending with the group for the next state upwards from the specified state.
Specified state or next existing state upward (SUP)	Specified state or upward: there are several groups, each of which includes revisions (if any) at just one lifecycle state. The order of preference for the groups is then: **1** the group for the specified lifecycle state **2** the group for the next later normal lifecycle state upward from this then so on, ending with the group for the final state.
Specified state only (EQS)	Equal to specified state: there is just one group – it consists of revisions which are at the specified lifecycle state only. (A relevant-item of the given type, which has no revision at this state, will not be selected using this template.)

Rule	Grouping and Order of Preference
Specified build stage and all next existing build stages upward (BUP)	Latest from specified build stage and upward: there are several groups, each of which includes revisions (if any) at just one build stage. (Conversely, if a relevant-item of the given type has all its revisions at stages which are below the specified build stage, then such a relevant-item is not selected using this template.) The order of preference is then: **1** the group for the specified build stage **2** the group for the next existing higher build stage upward from this then so on, ending with the group for the final stage.
Specified build stage only (EQB)	Equal to specified build stage: there is just one group – it consists of revisions which are at the specified build stage only. (A relevant-item of the given type, which has no revision at this stage, will not be selected using this template.)

Illustration of Rule Operation

The precise meaning of the different rules is perhaps best illustrated by considering this rather abstract example. Suppose a normal lifecycle is defined as:

STATE 1 → STATE 2 → STATE 3 → STATE 4 → STATE 5

and suppose a rule is specified as STATE 2, along with the rule shown in the left-hand column below. Then, for a relevant-item of the type concerned within this rule, the revision which is selected for inclusion in the baseline is the most recently updated revision chosen from those whose status is as shown in the right-hand column.

Rule	Selects Latest Revision from those:
Latest from state (LFS)	at *any* of the states: STATE 2, STATE 3, STATE 4, STATE 5.
Most progressed state above specified state or specified state (MPS)	at STATE 5; *or else* at STATE 4; *or else* at STATE 3; *or else* at STATE 2.
Specified state or most progressed state (SMP)	at STATE 2; *or else* at STATE 5; *or else* at STATE 4; *or else* at STATE 3.
Specified state or next existing state upward (SUP)	at STATE 2; *or else* at STATE 3; *or else* at STATE 4; *or else* at STATE 5.
Specified state only (EQS)	at STATE 2 only.

In the first and last cases the item is not included in the baseline, if no revision meets the given requirement. In the middle three cases, we look at each "*or else*" in the list, only if no revision has been found which meets the requirements up to that point; the item is not included in the baseline, only if we reach the end of the list, still without success.

Rules Using Implicit-States

Instead of specifying a particular state in the normal lifecycle, one of the following six implicit-states may be chosen. These do not apply to build stages. None of the rules discussed in the previous sections are required in such cases, as each such title implies its own order-of-preference grouping:

Implicit State	Grouping and Order of Preference
*All revisions (*ALL)	All revisions of each relevant-item of the type specified, regardless of their lifecycle status, are selected for inclusion in a baseline specified using this template. This is for use with Dimensions Archive, Retrieval, and Transfer facilities (ART). If a baseline is made using a template which contains one or more rules with *ALL, such a baseline *cannot* be used for configuration-build or release-control purposes.
*Latest edit revision (*LATEST)	There is just one group, consisting of all revisions which are at *any* of the normal lifecycle states. It is equivalent to choosing the first normal lifecycle state specifically, with a LFS rule.
*Latest edit revision at final state in lifecycle (*FINAL)	There is just one group, consisting of revisions at the final normal lifecycle state only. It is equivalent to choosing that state specifically, with an EQS rule.
*Latest edited revision at the most progressed state (*HIGHEST)	The order of preference of the groups is: first the group of revisions (if any) at the final normal lifecycle state, then the group of revisions at the next-to-final state, and so on downwards to the group at the first normal lifecycle state. It is equivalent to choosing that first state specifically, with a MPS rule. This implicit state is principally of value at an early stage in the development of a product, as it ensures that every relevant existing item (of the type that the rule is concerned with) gets selected at the latest revision of its most-progressed status.
*Revision built from selected inputs (*BUILT)	The selection procedure used here is entirely different from that specified above for other implicit states; and regardless of the order in which they were specified, rules with implicit states *BUILT (and *MADE OF – see below) are interpreted after all other rules have been used to select item revisions of other types. *All* relevant-items of the type(s) specified in *BUILT rule(s) are reviewed for potential inclusion in the baseline, and each relevant-item is inspected individually as follows. The lifecycle status of the relevant-item's revisions is **not** significant in this case, but each of the revisions is considered in turn, to see if one can be found which has been built, using as input(s) some of the item revisions already included in the baseline. For such a revision to qualify, it **must** have been the output of a Dimensions build process, and *all* the inputs used in that process *must* be found, *either* already included in the baseline (as a result of processing the template rules of other types – i.e. rules other than *BUILT), *or* included in the baseline as a result of earlier processing by this *BUILT procedure. (But it does not matter in which order *BUILT rules are specified, because this *BUILT processing is reiterative: the whole cycle will be repeated as often as necessary on all the relevant-items of types in *BUILT rules, until all possible orders and combinations have been dealt with.) If such a revision can be found to qualify, then this revision is also included in the baseline, and the item is considered to have been successfully selected.

Implicit State	Grouping and Order of Preference
*Revision built from selected inputs (*BUILT) (*Continued*)	However, if each of the item's revisions has been considered, and none has been found to meet these criteria (and further iterations of the entire cycle could not improve on this), then a warning message is issued to say that this item has not been included in the baseline; and the inspection process starts over again on the revisions of the next relevant-item.
*Revision that makes selected outputs (*MADE OF)	This selection procedure uses an inverse criterion to that used for *BUILT. In *MADE OF rules also, the lifecycle status of relevant-item revisions of the types specified is *not* significant in selecting them for inclusion in the baseline. In *MADE OF rules the selection is made by examining the made-of lists of built item revisions which either are already in the baseline or have been included in it as a result of previous iterations of this *MADE OF processing. Wherever such a made-of list contains any item revision which is of the type(s) specified in *MADE OF rule(s), as well as being a relevant-item (i.e. within the scope of the baseline), then that item revision is also included in the baseline, *provided that no other* made-of list so examined contains a *different* revision of the same item. If such a conflict of revisions does arise, then *no revision* of that item is included in the baseline, and an error message is issued to explain this.

For notes and guidelines on *using *BUILT and *MADE OF rules* refer to this sub-topic below.

The Default Rule and Other Uses of Implicit-States

The default rule (item type *) is applied to all item types not specified in other rules. These default types may be associated with several different item lifecycles, and therefore no specific lifecycle state may be used in the default rule, but instead one of the above implicit-states must be chosen. Therein lies the value of such titles as *FINAL, *LATEST and *HIGHEST, as they can be re-interpreted according to the different lifecycles associated with the types of item to be selected by the default rule.

The default rule is just a short alternative to specifying a separate rule (with the same implicit-state in each) for every item type used by all the relevant-items, except for the item types which have rules to themselves.

The implicit-states can, of course, be useful in other rules besides the default rule. If a template is to select items of several different types, and for each of them to pick, say, the latest revision at the final normal lifecycle state, it is probably more convenient (and clearer) to choose *FINAL for the rule for each of these item types, rather than look up and select the name of the actual final normal state in each case. (The template would function equally effectively, specified either way.) A further advantage of implicit states can be in defining general-purpose templates, for use by several different products. For example, several products may have items of type XYZ, but in each product a different lifecycle is specified for that item-type. A template consisting of the single rule: XYZ *LATEST could be used by any of these products to create a baseline which included the latest revision (at any normal lifecycle state) of each relevant-item of type XYZ.

Using *BUILT and *MADE OF Implicit States

A typical build configuration is:

- a compiler build tool which produces one item of type OBJECT from one input of type SOURCE; and
- a linker build tool which produces one output item of type EXE from several inputs of type OBJECT.

There will usually be other inputs in addition to these, but by considering this simple example of a configuration involving just three item-types, it will become apparent how *BUILT and *MADE OF can be used in any configuration.

Example 1

A suitable baseline template would be:

EXE	*BUILT
OBJECT	*BUILT
SOURCE	*MADE OF

(Note that the order in which template rules are specified makes no difference to the content of a baseline.) The effect of this template is:

- first (because *BUILT processing comes at the end) to include in the baseline SOURCE item revisions at the *FINAL (i.e. final normal) state;
- then to include those OBJECT item revisions which were built (compiled) from the SOURCE revisions already included;
- finally to include any EXE item revision(s) built (linked) from some combination of these included OBJECT revisions.

There must be a specified rule (or the default rule) to cover each stage in the build process: if the rule for OBJECT had been omitted, then the rule for EXE could not have included anything, because none of the build inputs for any EXE item would be present.

Example 2

Another suitable baseline template would be:

EXE	*HIGHEST
OBJECT	*MADE OF
SOURCE	*MADE OF

(Again note that the rules could be given in any order.) This template might be used to prepare a beta-test version of a product for release. The effect is:

- first (because *MADE OF processing, like that for *BUILT, comes at the end) to include in the baseline all the EXE items, choosing the revisions at the *HIGHEST state (i.e. the best revision available so far for each EXE item);
- then to include those revisions of OBJECT items which were used to build the revision of each EXE item already included, *unless* it is found that two or more different revisions of the same OBJECT item were used in building those EXE items (the EXE rule would have to have included at least two different items for such a conflict to possibly arise) – if such a conflict is found, it is flagged and reported, and *no* revision of that OBJECT item is included;

- then to include the SOURCE item revisions which were used to build each OBJECT item revision already included (and therefore not flagged) – which, provided that there is a one-to-one correspondence between SOURCE and OBJECT items, will not result in any further conflicts being found.

Again, there must be a rule to cover each stage in the build process: if the rule for OBJECT had been omitted, then the rule for SOURCE could not have included anything, because the SOURCE items do not themselves belong to the made-of list of any EXE item. (Although an EXE item is built ultimately from SOURCE items, this requires made-of lists to be used in combination.)

Example 3

Another example of how revision conflicts could arise would be in a template consisting of:

OBJECT	*LATEST
SOURCE	*MADE OF
ENV	*MADE OF

(where ENV items are environment-items, some of which may be used in the build compilations of several different OBJECT items), different revisions of one ENV item might have been used in the compilations which produced the OBJECT items already included.

Example 4

It is valid to have a template with both *BUILT and *MADE OF rules, such as:

EXE	*BUILT
OBJECT	*LATEST
SOURCE	*Revision that makes selected outputs

It is necessary that the other rule(s) in such a template include items which are in between in the build process. This means that the rules for *BUILT inclusion and those for *MADE OF inclusion will never conflict with each other. In this example OBJECT items have been included first, followed by both the EXE revisions built from them and the SOURCE revisions used to build them.

Example 5

It is *not* valid to have a template consisting solely of *BUILT and/or *MADE OF rules. For example a template of simply:

OBJECT	*BUILT
SOURCE	*MADE OF

would include nothing, because there is nothing selected to start off with, and so neither the *BUILT nor the *MADE OF processing could possibly find anything to qualify for inclusion.

Note 1: *MADE OF rules do not guarantee that the baseline will include the entire contents of any made-of list. If a complete made-of list is to be assured of being included, then:

either the template should be designed to use *BUILT; or else the *MADE OF rules must be sufficiently all-embracing to cover the made-of list. For example:

EXE	*HIGHEST
*	*MADE OF

could be used to include all the made-of lists for the included EXE items; *but only provided that* the scope of the baseline is large enough to include every configuration part which was specified or implied in each of the build processes used.

Note 2: *BUILT and *MADE OF rules are incompatible with *ALL rules: neither *BUILT nor *MADE OF may be used in any template which contains one or more *ALL rules.

Cross-Product Baselines

A cross-product baseline is one which includes design parts and/or items which belong to two or more products, because within the baselined design-structure USAGE relationships have been created (using the Dimensions functions Relate Design Part Usage and/or using Relate Item to Part) and these have specified design parts and/or items in other product(s).

The product whose design-structure (or a sub-tree of it) is being baselined, is referred to here as the **baselining product**, and all relevant-items which do not belong to this product, are called **foreign-items** belonging to **foreign products**.

To create a cross-product baseline successfully, the first requirement is that the item-types of *all* relevant-items, *including* foreign-items, *must* be defined in the process model for the baselining product.

The remaining requirements and limitations concern the lifecycles associated with these item types, in the various process models for all the products represented in the baseline. Obviously, the most straightforward case is where, for each item type concerned, taken in turn, that item type uses the same lifecycle in every process model. If this is indeed the case, then there are no further complications, and revisions of all relevant-items, including foreign-items, can be selected using the procedures already described.

However, it may be impracticable for different products to share identical lifecycles for the same item type. Therefore when the lifecycle used for an item type in a foreign product differs from that used for the same item type in the baselining product, the extent to which these two lifecycles need to be similar to each other needs to be examined in some detail.

First, suppose the normal lifecycles in the two lifecycles are identical: the same state names appear in the same order; and the lifecycles differ only in the number and/or arrangement of the lifecycle transitions which do not lie on the normal path. Then there is no further problem for relevant-items of that type in either product: as far as a baseline-template is concerned, the two lifecycles function in an identical manner, because their normal lifecycles are equivalent.

We now look at the cases where the normal lifecycles for an item type differ in some respect between the baselining product and a foreign product. We need to consider individually the different alternatives for the template rule for that item type, starting with the simplest.

- *ALL rule

  This rule raises no problems, as it selects all revisions of a relevant-item, regardless of their lifecycle states. This applies equally to foreign-items.

- *BUILT and *MADE OF rules

  These rules also raise no particular problems, as they are processed in an entirely different way, and are not directly concerned with normal lifecycles. The *BUILT and *MADE OF procedures are applied equally to relevant-items of both the baselining and foreign products.

- *LATEST rule

  This rule is handled in a special way, the effect of which is that the normal lifecycles for the baselining and foreign products do not need to correspond with each other at all. For a relevant-item of the baselining product, the latest revision is selected from all at any of the states in the baselining product's normal lifecycle; and for a foreign-item, the latest revision is selected from all at any of the states in that foreign product's own normal lifecycle.

- LFS rule with a specific state

  This is a more general version of *LATEST and is handled as follows. The specific state given is looked up in the normal lifecycle for the baselining product. Let us suppose the normal lifecycle has six states and that the specified state is number 3 (the first state is number 1, and the final normal state, in this case, number 6). For a relevant-item of the baselining product, the latest revision would be selected from all those at any of the states whose numbers are 3, 4, 5 and 6. For a foreign-item, we look up the foreign product's normal lifecycle for the names of the states which are number 3 (the same number as that of the specified state) or greater, and select the latest revision from those at any of *these* states.

  This would mean, if this normal lifecycle had nine states instead of six, that there would be seven states from which a selection could be made (numbers 3 to 9 inclusive) instead of four in the baselining product. In an extreme case, if the normal lifecycle for a foreign product had only two states (unlikely, but possible), then there would be no states from which to select, and so no revision could be selected for a foreign-item of this item type in that product.

  One example of this rule, which could be very useful, is to specify LFS along with the state which is *number 2* in the baselining product's normal lifecycle. The effect of this rule would be that, for every relevant-item of that item type in *any product*, the revision selected would be the latest at any normal state, *excluding* those which had initial status; i.e. it would guarantee that the selected revision had been actioned at least once after modification.

- All other codes and implicit states *FINAL and *HIGHEST

  A rule of this kind requires a significant degree of consistency between the normal lifecycles in the baselining product and a foreign product, in order to satisfactorily select a revision of a foreign-item. What happens is that only the baselining product's normal lifecycle is consulted, and the search (in the appropriate order of preference) is performed on all relevant-items of that type (i.e. associated with that normal lifecycle), including foreign-items, using the state names in the baselining product's normal lifecycle.

  So, to take one of the simpler cases, if the rule specified *FINAL, and the final normal state in the baselining product's lifecycle was DONE, then DONE would at least have to appear somewhere in the corresponding lifecycle for the foreign product, and the

foreign-item's revision selected would be the latest (if any) which had DONE status, regardless of what DONE actually meant in the foreign product's lifecycle.

It is therefore necessary, if any of these codes are going to be used, to have sufficient agreement among the normal lifecycles for all the products concerned, as to the meaning of most, if not all, of the state names used, in order to ensure that the selected revision of a foreign-item is the one actually desired.

Suspended Item Revisions

If the selection is by a *ALL (which is for archiving only), any suspended status is ignored, and all revisions, both non-suspended and suspended, are selected for inclusion.

Apart from *ALL, a suspended revision; is never selected for inclusion in a new release-baseline, created using a baseline-template. The selection processes already detailed (apart from those for *ALL) are applied exactly as if the suspended revisions did not exist.

(Suspended revisions can, of course, be found in already existing template release-baselines, due to having been selected for inclusion because they had not yet been suspended at the times when those baselines were created. It is also possible for suspended revisions to be included in new non-template baselines: i.e. revised and merged baselines.)

Checked Out Item Revisions

A checked out revision is never included in a release-baseline (and once a revision has been included in a release-baseline, its file is no longer available to be checked out – or indeed, altered in any way).

If the selection is by a *ALL rule, and a checked out revision would have been among those selected (if it had not been checked out), then this is flagged, and the requested baseline is not created.

Apart from *ALL rules, the selection processes already detailed are applied exactly as if the checked out revisions did not exist.

Suspended Design Parts

The specifications already given apply to all relevant items which are either OWNED or USED by at least one non-suspended design part included in the baseline. But if any item is relevant only because all the included design parts, which either OWN it or USE it, are SUSPENDED, then its processing is as above for "Suspended Item Revisions". All the revisions of such an item are processed as if they were suspended, regardless of their actual status.

Examples of Baseline Template Rules

Suppose the normal lifecycle for item type XYZ is:

```
PRELIMINARY → DESIGNED → COMPLETED → TESTED → RELEASED
```

and suppose an item of type XYZ (OWNED by an included, non-suspended design part) exists in five revisions (none of which is currently checked out or suspended) as follows,

each most recently updated on the date shown, and then actioned to the lifecycle state shown:

Revision	Last Updated	Current State
1	1/1/2003	TESTED
2	1/2/2003	TESTED
3	1/4/2003	COMPLETED
4	1/3/2003	COMPLETED
5	1/5/2003	PRELIMINARY

Then consider in turn each of the following *alternative choices* of template rule for item type XYZ. If the template rule were as shown, then for the above item, the revision which would be selected, together with the reason for this selection, is as shown alongside.

Lifecycle State	Code	Revision Selected	Reason
DESIGNED	Latest from state (LFS)	3	1, 2, 3 & 4 qualify; 3 is latest
DESIGNED	Most progressed state above specified state or specified state (MPS)	2	Latest at nearest-to-final state
DESIGNED	Specified state or most progressed state (SMP)	2	Latest at nearest-to-final state
DESIGNED	Specified state or next existing state upward (SUP)	3	Latest at next state up
DESIGNED	Specified state only (EQS)	None	None at DESIGNED state
PRELIMINARY	Most progressed state above specified state or specified state (MPS)	2	Nearest-to-final state preferred
PRELIMINARY	Specified state or most progressed state (SMP)	5	Specified state preferred
*All revisions (*ALL)		1,2,3,4 & 5	Includes everything
*Latest edit revision (*LATEST)		5	Latest of all at normal states
*Latest edit revision at final state in lifecycle (*FINAL)		None	None at final state

Lifecycle State	Code	Revision Selected	Reason
*Latest edited revision at the most progressed state (*HIGHEST)		2	Work back from final state; take latest at 1st state found
*Revision built from selected inputs (*BUILT)		?	Cannot say; see following notes
*Revision that makes selected outputs (*MADE OF)		?	Cannot say; see following notes

Notes

1 "None" means that, if this were the template rule, then this item would not appear in the baseline.

2 The revision (if any) which would be selected by *BUILT cannot be determined from the information given. This is because if this is a built item, the revision selected would depend on the corresponding revision(s) of the input item(s) used to build it, and those revisions would be selected by template rules for other item-types.

3 The revision (if any) which would be selected by *MADE OF cannot be determined from the information given. This is because if this is an item selected because it is in a made-of list of some built item, the revision selected would depend on which revision of the built item was included, and that revision would be selected by a template rule for some other item-type.

4 Revisions 1 or 4 could not be selected by any rule (except *ALL, and possibly *BUILT or *MADE OF, because in each case another revision, more recently updated, exists at the same lifecycle state.

Request Baseline Templates

Request baseline templates enable you to specify rules for selecting those requests that you specify as input for creating a baseline. When you create a baseline and you specify a list of requests, the requests will be filtered according to the rules defined in the template. Note that you need to specify one or more requests when you create the baseline. The template rules do not directly define these requests.

They comprise one or more rules that are made up from the following:

- Request type.
- Request status.
- Baseline status code, which itself comprises one of the following keys:
  - EQS – specified state only.
  - SUP – specified state or next existing state upward.

NOTE Baseline collective codes such as *MADE_OF and *LATEST, which are used in item baseline templates, are not relevant to request baselines and are not provided.

A request baseline template consists only of request template rules – it will not allow the addition of rules using item types. Conversely, an item baseline template does not allow the addition of request baseline template rules.

Creating a Baseline Using Request Baseline Templates

When a baseline is created specifying a request baseline template and a set of starting parent requests, then all the requests that:

- are related to those parent requests, and
- match the template rules

will be collected together for processing.

The template rules will be processed in exactly the same way they are for item templates, that is, requests will be selected based on the type, status, and the baseline status code that was specified. For example, if a template had a rule that said that:

- all requests of type PR,
- at status ACCEPTED,
- with the baseline status code EQS

were to be considered, then all the requests of type PR, at the status ACCEPTED only, would be used for inclusion into the baseline.

Once this list of requests has been determined, then **only** those items that are related to those requests with either an In Response To or, optionally, an Info relationship will be included into the baseline. However, because the baseline that is being created is a release baseline, only **one** revision of each item will be included in the baseline (not all revisions, as would be the case for a design baseline). This means, that even though the requests being selected may contain multiple revisions of the same item, the final baseline will only contain one revision of all these possible items.

To ensure that only one revision of an item is included in the final baseline in circumstances where multiple item revisions are related to requests, only the latest item revision will be selected using that item's pedigree. If item revisions are in conflict, that is, no common successor to these items are found, then the creation of the baseline will fail with an appropriate error message.

When the baseline has been created, the requests that were used to create it will be related as *In Response To* that new baseline.

Constraints

To enforce a consistent model with its behavior with respect to items only, the following additional constraints apply to creating a baseline with respect to cited requests:

1. Requests that are either at a closed or off-norm lifecycle state will not be processed by the SUP baseline code.
2. Only requests in the primary catalog will be processed.
3. Only requests related through a dependency relationship, that is, DEPEND, will be included in traversal scans.

4 If a request related through a dependency relationship does not fulfill the criteria specified in the request template rules, then that request will be ignored—as will every other request that is a child of that request. This means, that for a situation where N levels of requests are related together in a chain through dependent relationships, for example, CR_1 → CR_5 → CR_7 → CR_9 → CR_12, if CR_7 fails to match a template rule, then it, and CR_9 and CR_12 will be ignored from any further processing.

5 Only requests that are owned by the product on which the baseline is being created will be processed. Any requests owned by other products will be ignored—this includes children that such requests might have.

6 Only parts or items that are owned, or have usage relationships to the parent part specified when the baseline is created will be included in the final baseline. This means, that if a request refers to affected parts and/or items that may be out of scope – that is, not owned by or related to the parent part or any of its children – then these parts and items will be ignored.

Release Templates

In Chapter 2, "Process Modeling Concepts", the concept of a release is introduced in the topic "About Object Classes" on page 30. A **Release Template:**

- Defines a set of user-defined rules or criteria by which baselined design parts are selected for inclusion in a release. This rule is then applied to that design part and also to any subordinate to it in the design-structure tree, unless and until another rule specifies any such subordinate design part, in which case that other rule overrides for that sub-tree, and so on. Alternatively, a rule may be applied to ALL design parts of the product.

  If asterisk (*) is specified as the design part, then the rule will be applied to all design parts for which no specific item type rule exists i.e. "select all". The definition of a default rule in a template ensures that all design parts within a baselined structure are considered for inclusion in the release.

- Defines a set of user-defined rules or criteria by which baselined items are selected for inclusion in a release.

  If asterisk (*) is specified as the item type, then the rule will be applied to all items for which no specific item type rule exists i.e. "select all". The definition of a default rule in a template ensures that all items within a baselined structure are considered for inclusion in the release.

  If hyphen (-) is specified as the item type, then the rule will be applied to no items (this is used to inhibit the selection of items for some sub-tree of the design part structure).

- Optionally specifies operating system subdirectories to be *added* to the release pathname specified by users (directory into which the above are to be copied). The subdirectory must be a relative format (such as **xxx/yyy** *UNIX* or ***xxx\yyy*** *Windows*). If a subdirectory **is** specified, the items of all types specified by the rule, for all design parts selected by the rule, are placed in the subdirectory with the *leaf node* portion of their project filename. On the other hand, if a subdirectory is **not** specified, the items' *project filenames* will be used, relative to the operating system release directory the user requests for the release.

Any operating system directories or subdirectories which do not exist, Dimensions CM creates as and when they are needed (subject to the user having the necessary operating system file access rights for such directory creation).

The objective of subdirectories is to simplify subsequent handling of the release data. For example, a release-template can be used to specify that executable code, source-code modules, user documentation and system-specification documents are each to be grouped in different subdirectories. It is possible to include one item a number of times in a release, provided each copy is assigned to a different subdirectory.

- In support of the above operations, to optionally assign a database-unique "group name" to a number of otherwise unrelated item types. The group name may then be specified in a template rule, thereby bringing all the associated item types within the scope of that rule.

Release templates are defined using the Administration Console *Baseline and Release Templates* function.

NOTE All release-template rules apply only to the items selected by the baseline specified for a release. If a release-template is not used, then all the items in the baseline are placed in a single release directory, their *project filenames* being used relative to the operating system release directory the user requests for the release.

Different release-templates may be used with the same baseline to create a number of different release-configurations (e.g. test configurations of subsystems) or to make available different groups of items (e.g. user documentation).

Where appropriate, the same release-template may be used:

- in creating different releases of the same configuration
- in conjunction with different baselines and for different products.

Appendix C

Scripting Interface

In this Chapter

Introduction

Dimensions is a rich environment available on a wide range of platforms. Until recently Dimensions functionality could only be accessed via the command-line or from the DTK/trigger API. Both of these interfaces exposed only a subset of the existing functionality. The scripting interface overcomes this limitation by providing a set of Java classes that expose Dimensions components via a simple and consistent object model. This object model is primarily intended for use by scripting languages hosted in the JVM (Java Virtual Machine), such as JavaScript™, Python, Tcl and others. The current scripting interface includes Rhino, an open-source Java-based JavaScript interpreter from www.mozilla.org, and sample scripts that demonstrate the functionality that is available.

This chapter is intended for installers, administrators and users of Dimensions on UNIX and Windows platforms.

Supported OS Platforms

All platforms certified for use with a Dimensions 14.x server are supported, provided that *Java™ 2 Runtime Environment Standard Edition,* version 1.3.x or higher, is available on that platform.

Dimensions Components

The scripting interface includes support for the following Dimensions components.

Supported Dimensions Component	Supported Operations
Attribute Blocks	■ List existing attribute block definitions. ■ Create a new attribute block definition for an object type. ■ Update an attribute block definition. ■ Delete an attribute block definition. ■ List attributes assigned to an attribute block's columns. ■ Assign attributes to, or deassign attributes from, an attribute block's column.
Attribute Definition	■ List existing attribute definitions. ■ Create a new attribute definition for an object type. ■ Update an attribute definition. ■ Deassign an attribute definition from an object type; and delete it (optional). ■ List types assigned to an attribute definition.

(Sheet 1 of 6)

Supported Dimensions Component	Supported Operations
Base Database	■ Log in to Dimensions. ■ Log out from Dimensions. ■ Authenticate a user in order to perform tertiary node access to items located on a remote node.
Baseline Template	■ List existing baseline templates. ■ Create a baseline template. ■ Delete a baseline template. ■ List baselines using a baseline template. ■ Copy baseline template rule definitions between baseline templates.
Baseline Template Rule	■ List existing baseline template rules. ■ Create a baseline template rule. ■ Delete a baseline template rule. ■ Update a baseline template rule.
Build Areas	■ List existing build areas. ■ Create new build area definition. ■ Update a build area definition. ■ Delete a build area definition.
Build Projects	■ List existing build projects. ■ Create new build project definition. ■ Update a build project definition. ■ Delete a build project definition. ■ List projects assigned to a build project.
Build Stages	■ List existing build stages. ■ List lifecycle states using a build stage. ■ List build areas defined for a build stage.
Request Relationship Type	■ List defined request relationship types. ■ Create a relationship type. ■ Delete a relationship type. ■ Update a relationship type.
Request Type CM Rules	■ Define and/or update change management (CM) rules.
File Format	■ List existing formats. ■ Create a new format. ■ Update format attributes. ■ Delete a format.

(Sheet 2 of 6)

Supported Dimensions Component	Supported Operations
Item, Request, and Baseline Replication Configurations	■ List existing item and baseline replication configurations. ■ Create new replication configuration. ■ Update a replication configuration definition. ■ Delete a replication configuration definition.
Item To Item Relationship Type	■ List item to item relationship types. ■ Create an item to item relationship type. ■ Delete an item to item relationship type. ■ Update an item to item relationship type.
Item Type CM Rules	■ Define and/or update change management rules.
Item Type Group	■ List defined item type groups. ■ Assign an item type to an item type group. ■ Deassign an item type from an item group. ■ List types assigned to an item type group. ■ List release templates assigned to an item group
Item/Request Browse Templates	■ List existing browse template revisions. ■ Create a template revision. ■ Delete a template revision. ■ Import a file into a template revision. ■ Export a file from a template revision.
Lifecycle	■ List existing lifecycles. ■ Create a new lifecycle. ■ Update lifecycle attributes. ■ Delete a lifecycle. ■ List the types using a lifecycle. ■ List assigned image revisions. ■ Assign, or deassign, default image revisions to a lifecycle.
Lifecycle and Browse Template Assignments at Design Part Level	■ List existing assignments for a request type. ■ Create a new assignment. ■ Delete an assignment.
Lifecycle Image Revision	■ List lifecycle image revisions assigned to a lifecycle. ■ Create an image revision. ■ Delete an image revision. ■ Import a file into an image revision. ■ Export a file from an image revision

(Sheet 3 of 6)

Supported Dimensions Component	Supported Operations
Lifecycle State Transition	■ List a lifecycle's state transitions. ■ Create a new transition. ■ Delete a transition. ■ Add roles to, and remove roles from, the list of roles authorized to perform a transition. ■ Rename a state. ■ Delete a normal state.
Object Type	■ For all object types: ■ List existing object types. ■ Create an object type. ■ Update an object type definition. ■ Delete an object type definition. ■ Assign, or deassign, a lifecycle. ■ Define attributes and attribute blocks. ■ Copy attributes between object types. ■ For item types: ■ List assigned file formats. ■ Assign, or deassign, a file format. ■ Assign, or deassign, a default template revision to an item/request object type. ■ Define, update or delete item libraries and/or the default item library. ■ Define change management rules. ■ For request types: ■ Assign, or deassign, a default template revision to an item or request object type. ■ Define change management rules.
Priming Relationship	■ List priming relationships defined for a request type. ■ Create a priming relationship. ■ Delete a priming relationship. ■ List attribute mappings. ■ Add and/or delete attribute mappings.
Product	■ List existing products. ■ Create a new product. ■ Update a product definition. ■ Delete a product.

(Sheet 4 of 6)

Supported Dimensions Component	Supported Operations
Projects	■ List existing projects. ■ Create new project definition. ■ Update a project definition. ■ Delete a project definition. ■ List build areas defined for a project.
Relationship Names	■ List relationship names defined in a base database. ■ Create a relationship name. ■ Delete a relationship name. ■ Update a relationship name. ■ List item to item relationship types using a relationship name.
Release Template	■ List existing release templates. ■ Create a release template. ■ Delete a release template. ■ List releases using a release template. ■ Copy release template rule definitions between release templates.
Release Template Rule	■ List existing release template rules. ■ Create a release template rule. ■ Delete a release template rule. ■ Update a release template rule.
Role	■ List existing roles. ■ Create a new role. ■ Update role attributes. ■ Delete a role.
Role Assignment	■ List existing role assignments. ■ Add a role assignment. ■ Delete a role assignment.
User Report Definitions	■ List existing user report definitions. ■ Create a new report definition. ■ Update a report definition. ■ Delete a report definition. ■ Assign report files to, or deassign report files from, a report definition.s

(Sheet 5 of 6)

Supported Dimensions Component	Supported Operations
User Report Files	■ List existing report files. ■ Create a new report file. ■ Delete a report file. ■ Import a file into a report file. ■ Export the contents of a report file.
Valid CM Relationship Type	■ List valid change management relationships defined for a request type. ■ Create a valid relationship type. ■ Delete a valid relationship type. ■ Update a valid relationship type.
Valid Sets	■ Add a valid set object. ■ Delete a valid set object.
Version Branch	■ List existing branches. ■ Create a new branch. ■ Update branch attributes. ■ Delete a branch.

(Sheet 6 of 6)

Using the Scripting Interface Shell

Running the Shell

To run the scripting interface shell, type `dmpmcli` at your command prompt.

Syntax

The shell has the following syntax:

```
dmpmcli
[-user <username>
-pass <password>
-dbname <database>
-host <server>
-dsn <dsn-name>
]
[-param <parameter file>
[-file <script file> [<script arguments>]]
[-help]
[-version]
```

where:

Parameter	Description
-user <username>	Specifies a user ID that is registered on the Dimensions base database.
-pass <password>	Specifies the password for the user ID that you entered above.
-dbname <database>	Specifies the name of the base database.
-host <server>	Specifies the name of the Dimensions server. The `<server>` string may explicitly specify the port number of the Dimensions listener, for example: -host servername:5555
-dsn <dsn-name>	Specifies the connection string that enables you to connect to the database server for your Dimensions base database.
-param <parameter file>	Specifies a file containing connection parameters (see the example immediately below this table).
-file <script file> <script arguments>]	Specifies a file containing a script to be executed, and the script's (optional) arguments.
-help	Displays Help for the `dmpmcli` syntax.
-version	Displays the version of `dmpmcli` that you are running.

Prior to executing the specified javascript spource:

- dmpmcli overrides and specifies its own custom error reporting.
- Dimensions CM defines the additional non-ECMA global javascript functions:
  - system
  - source
  - run
  - print
  - quit
  - version
  - help
  - exit
- The variable "arguments" are populated with the specified command-line arguments.

Specifying Connection Parameters in a File

You can specify the connection parameters in a file rather than in the command line, for example:

```
-user dmsys
-pass <password>
-dbname intermediate
-host server1
-dsn PC50
```

Executing Scripts

You can use the -file option to specify a file containing a JavaScript™ script that is to be executed. If you do not specify a script, the interpreter enters interactive mode and the script source is read and executed from standard input until an EOF (End of File) character is detected (CTRL+Z on Win32 platforms and CTRL+D on UNIX platforms), or you invoke the exit() or quit() functions.

Examples

Entering interactive mode:

```
dmpmcli -user dmsys -pass <password> -dbname intermediate
-dsn PC50 -host dimhost
```

Connects to the Dimensions server node dimhost and enters interactive mode. To quit dmpmcli use the quit() or exit() functions.

Executing a script:

```
dmpmcli -user dmsys -pass <password> -dbname intermediate
-dsn PC50 -host dimhost:5555 -file
/home/dmsys/script/setupAttributes.js
```

Connects to the Dimensions server node dimhost (assuming that the Dimensions Listener is listening on port 5555) and executes the following script:

```
/home/dmsys/script/setupAttributes.js
```

Reading connection information from a file:

```
dmpmcli -param /home/dmsys/connection.data -file
/home/dmsys/script/setupAttributes.js
```

```
Connects to the Dimensions server node identified by the connection
information in the file
```

```
/home/dmsys/connection.data,
```

and executes the following script:

```
/home/dmsys/script/setupAttributes.js
```

Predefined Properties

Scripts executing in the dmpmcli shell have access to the following pre-defined top-level object properties:

Command	Description
arguments	An array containing the strings of all the arguments given at the command line when the shell was invoked.
help()	Displays usage and Help messages.
defineClass(className)	Defines an extension using the Java class named with the string argument. Uses ScriptableObject.defineClass()

Command	Description
run(["foo.js", "arg0", ...])	Runs the JavaScript source file specified by the first string in the argument, and passes the remaining arguments as that script's arguments. The parent environment is visible but *cannot* be modified.
source(["foo.js", "arg0", ...])	Runs the JavaScript source file named by the first string in the argument, and passes the remaining arguments as that script's arguments. The parent environment is visible and *can* be modified.
system(["cmd", ...])	Runs one or several native OS commands specified by the string argument.
loadClass(className)	Loads a class named by the string argument. The class must be a script compiled to a class file.
print([expr ...])	Evaluates and prints expressions.
quit() exit()	Quits the shell. The shell will also quit in interactive mode if you enter an EOF character at the prompt.
version([number])	Gets or sets the JavaScript version number.

Using the Object Model

Access to the Java objects from JavaScript™ is based on the LiveConnect technology. For a detailed description of JavaScript™ and LiveConnect, refer to the Core JavaScript Guide and Core JavaScript Reference documentation at the following URL:
`http://developer.netscape.com/docs/manuals/`

For an introduction to Java scripting with Rhino, refer to Rhino documentation at the following URL:
`http://www.mozilla.org/rhino/scriptjava.html`

NOTE For performance reasons, running Dimensions commands from dmpmcli via `BaseDatadabase.instance.runCommand()` does not refresh the object model collections.

Interface Summary

Interface	Description
AttributeBlock	Represents an attribute block (a multi-value, multi-field attribute definition).
AttributeDefinition	Represents one of the following attribute definitions: ■ Single-value, single-field. ■ Single-value, multi-field. Contains the collection of assigned object types.
Baseline	Represents a baseline.
BaselineReplicationConfiguration	Represents a baseline replication configuration.
BaselineTemplate	Represents a baseline template and contains a collection of assigned baseline template rules.
BaselineTemplateRule	Represents a baseline template rule.
BuildArea	Represents a build area.
BuildProject	Represents a build project.
BuildStage	Represents a build stage.
ChangeDocumentType	Represents a request type, and contains the collection of assigned browse templates.
ChangeDocumentTypeCMRules	Represents a request type's change management rules.
ChdocRelationshipType	Represents a request relationship type.
ChdocReplicationConfiguration	Represents a request replication configuration.
DimensionsObject	A marker interface representing a generic Dimensions object.
DimensionsObjectDetails	A marker interface representing generic details of a Dimensions object.
FileFormat	Represents a file format.
ItemLibrary	Represents an item type's item library or the default item library.
ItemRemoteSubordinate	Represents a remote subordinate definition used by item replication configurations.
ItemReplicationConfiguration	Represents an item replication configuration.
ItemToItemRelationshipType	Represents an item to item relationship type.
ItemType	Represents an item type and contains the collection of assigned file formats.
ItemTypeCMRules	Represents an item type's change management rules.
ItemTypeGroup	Represents an item type group.
Lifecycle	Represents a lifecycle and contains collections of lifecycle state transitions, assigned object types, and lifecycle image revisions.
LifecycleAndTemplateAssignment	Represents a lifecycle and browse template assignment for a request type at the design part level.

(Sheet 1 of 2)

Interface	Description
LifecycleImageRevision	Represents a lifecycle image revision.
LifecycleTransition	Represents a lifecycle state transition and contains the collection of roles authorized to perform the transition.
LocalSubordinate	Represents a local subordinate definition used by item replication configurations.
PrimingRelationship	Represents a priming relationship and associated primed attribute mappings.
Product	Represents a product and contains collections of role assignment details, object types, baseline templates, release templates, browse templates, request relationship types, and valid sets.
Project	Represents a Dimensions project.
ProjectStage	Represents a Dimensions project stage.
RelationshipName	Represents an item to item relationship name.
Release	Represents a release.
ReleaseTemplate	Represents a release template and contains a collection of assigned release template rules.
ReleaseTemplateRule	Represents a release template rule.
RemoteSubordinate	Represents a remote subordinate definition used by baseline replication configurations.
Replication	Represents the replication administration tool.
ReplicationConfiguration	Represents an abstract replication configuration.
ReplicationSubordinate	Represents an abstract subordinate definition.
Report	Represents a user report definition.
ReportFile	Represents a report file.
Role	Represents a project role.
TemplateRevision	Represents an item/request type browse template revision.
Type	Represents an object type, and contains collections of attribute blocks, attribute definitions and attribute rules.
ValidRelationshipType	Represents a valid relationship type.
ValidSet	Represents a valid set.
VersionBranch	Represents a version branch.

(Sheet 2 of 2)

For details about supported collection operations, see page 334.

Class Summary

Class	Description
AttributeBlockDetails	Represents details of an attribute block.
AttributeDataType	Represents predefined attribute data types.
AttributeDetails	Represents details of an attribute definition.
AttributeMappingDetails	Represents an attribute mapping used in priming.
AttributeRuleDetails	Represents the details of a Dimensions attribute rule.
AttributeType	Represents predefined attribute types.
defaultDatabase	Represents the root of the object hierarchy. Authenticates users and contains collections of version branches, file formats, item type groups, lifecycles, products, relationship names, and roles.
BaselineReplicationConfigurationDetails	Represents details of a baseline replication configuration.
BaselineTemplateDetails	Represents details of a baseline template.
BaselineTemplateRuleDetails	Represents details of a baseline template rule.
BuildAreaDetails	Represents details of a build area.
BuildProjectDetails	Represents details of a build project.
ChdocRelationshipTypeDetails	Represents details of a request relationship type.
ChdocReplicationConfigurationDetails	Represents details of a request replication configuration.§
ChdocSuperType	Represents predefined request super types.
CMPhasePermissions	Encapsulates permissions on operations that may be applied to requests, when they are at certain predefined lifecycle phases.
FileFormatDetails	Represents details of a file format.
FileFormatType	Represents predefined file format types.
ImplicitStateCode	Encapsulates predefined implicit state shortcut codes used by Dimensions baseline template rules that do not require a normal lifecycle state to be specified.
ItemLibraryDetails	Represents details of an item library.
ItemRemoteSubordinateDetails	Represents details of a remote subordinate definition used by item replication configurations.
ItemReplicationConfigurationDetails	Represents details of an item replication configuration.
ItemToItemRelationshipTypeDetails	Represents details of an item to item relationship type.
ItemTypeGroupDetails	Represents details of an item type group.
LifecycleAndTemplateAssignmentDetails	Represents details of a lifecycle and browse template assignment for a request type at the design part level.
LifecycleDetails	Represents details of a lifecycle.

(Sheet 1 of 2)

Class	Description
LifecycleImageRevisionDetails	Represents details of a lifecycle image revision.
LifecycleTransitionDetails	Represents details of a lifecycle state transition.
LocalSubordinateDetails	Represents details of a local subordinate definition used by item replication configurations.
OsType	Represents predefined OS types used by user-defined report definitions.
PrimingRelationshipDetails	Represents details of a priming relationship.
ProductDetails	Represents details of a product.
ProjectDetails	Represents details of a Dimensions project.
ProjectStageDetails	Represents the details of a Dimensions project stage.
PseudoBranches	Represents predefined pseudo branches used by Dimensions replication configurations.
RelationshipClass	Encapsulates the system-defined request relationship types *Info* and *Dependent*.
RelationshipNameDetails	Represents the details of an item type relationship name.
ReleaseTemplateDetails	Represents details of a release template.
ReleaseTemplateRuleDetails	Represents details of a release template rule.
RemoteSubordinateDetails	Represents details of a remote subordinate definition used by baseline replication configurations.
ReplicationConfigurationDetails	Represents details of an abstract replication configuration.
ReportDetails	Represents details of a user report definition.
ReportFileDetails	Represents details of a report file.
ReportScope	Encapsulates predefined Dimensions report scopes.
RoleAssignmentDetails	Represents details of a role assignment.
RoleCapability	Represents predefined role capabilities.
RoleDetails	Represents details of a role.
RuleStateSelector	Encapsulates predefined baseline template rule state selectors.
TemplateRevisionDetails	Represents details of a template revision.
TransitionRoleDetails	Represents details of a Dimensions role authorized to perform a lifecycle state transition.
TypeDetails	Represents details of an object type.
TypeOptions	Represents predefined object type options that might be enabled for a specific class of an object type.
TypeScope	Represents predefined object type scopes.
ValidRelationshipTypeDetails	Represents details of a valid relationship type.
ValidSetDetails	Represents details of a valid set.
ValidSetRowDetails	Represents an eight column row of a valid set.
VersionBranchDetails	Represents details of a version branch.

(Sheet 2 of 2)

Collection Interface Summary

Collection Interface	Description
AttributeBlocks	Contains all attribute blocks in a product.
AttributeDefinitions	Contains all attribute definitions in an object type.
BaselineTemplates	Contains all baseline templates in a base database.
BaselineTemplateRules	Contains all baseline template rules in a baseline template.
BuildAreas	Contains all build areas defined for a project or build stage.
BuildProjects	Contains all build projects in a base database.
BuildStages	Contains all build stages in a base database.
ChdocRelationshipTypes	Contains all request relationship types in a product.
DimensionsObjectCollection	Contains the common interface of the Dimensions collection hierarchy.
FileFormats	Contains all file formats in a base database.
ItemTypeGroups	Contains all item type groups in a base database.
LifecycleImageRevisions	Contains all lifecycle image revisions for a lifecycle.
Lifecycles	Contains all lifecycles in a base database.
LifecycleStates	Contains lifecycle state objects (cannot be modified).
LifecycleTransitions	Contains all lifecycle state transitions in a lifecycle.
LocalSubordinates	Contains all local subordinates for an item replication configuration.
Products	Contains all products in a base database.
Projects	Contains all projects in a base database.
ProjectStages	Contains all project stage objects that are defined for a project.
RelationshipNames	Contains all relationship names in a base database.
ReleaseTemplateRules	Contains all release template rules in a release template.
ReleaseTemplates	Contains all release templates in a base database.
RemoteSubordinates	Contains all remote subordinates for an item or a baseline replication configuration.
ReplicationConfigurations	Contains all replication configuration definitions scoped by replicated object type.
ReportFiles	Contains all report files in a base database.
Reports	Contains all user report definitions in a base database.
Roles	Contains all roles in a base database.
TemplateRevisions	Contains all template revisions defined for an item or request type.
Types	Contains all object types of the same scope in a product.
ValidSetRows	Contains all valid set rows in a valid set.
ValidSets	Contains all valid sets in a product.
VersionBranches	Contains all version branches in a base database.

Supported Collection Operations

The collection operations described in the table below apply to the collection interfaces in the section "Collection Interface Summary" on page 333

Operation Name	Description
Object add(ObjectDetails details)	Adds a new Dimensions object to the collection. Typically, creation of a Dimensions object requires many arguments. In order to deal with this relative complexity, the object model provides a number of *ObjectDetails* objects classes representing information required to create a Dimensions component. Normally, you create an *ObjectDetails* object and pass it to the collection's add method.
Object get(String name) Object get(String namePart1, String namePart2)	Returns an existing Dimensions object from the collection. If `null` is returned, the object was not found. Some collection objects are identified by a pair of 'names' (for example, a Lifecycle Transition is uniquely identified by its 'From State' and 'To State'). In such cases, the corresponding collection provides a convenient method, by accepting all components of a 'name' as parameters.
ObjectDetails remove(Object obj)	Removes a Dimensions object from the collection, and returns an ObjectDetails object that represents the removed object.
Iterator iterator()	Returns a java.util.Iterator object over the names of the collection objects.
void refresh()	Reloads the objects in the collection in order to reflect the current contents of the base database. You should reacquire references to the collection's objects after calling this method. **Note**: Changes to the data held in the base database made by other processes (for example, the Administration Console or other `dmpmcli` sessions) after the collection was accessed for the first time, might not be automatically reflected in the collection unless the refresh() method is explicitly invoked.
int size()	Returns the number of objects in the collection.

NOTE `Both add and remove methods automatically commit changes to the database.`

Web-Based API Documentation

For a full description of all exposed components and their respective properties and methods, refer to the Web-based (HTML) API documentation available in the following directories:

- UNIX platforms:
  `${DM_ROOT}/java_api/docs/api/index.html`
- Windows platforms:
  `%DM_ROOT%\java_api\docs\api\index.html`

Example Java Scripts

Example Java scripts are located in the following :

- UNIX platforms:
  `${DM_ROOT}/AdminConsole/examples/`
- Windows platforms:
  `%DM_ROOT%\AdminConsole\examples\`

The following Java scripts illustrate common Dimensions tasks:

Script Name	Description
assignedFormatsDemo.js	■ Assigns a file format to an item type. ■ Deassigns a file format from an item type. ■ Lists all file formats assigned to an item type.
assignedTypesDemo.js	■ Lists types assigned to a lifecycle. ■ Assigns a lifecycle to a type. ■ Deassigns a lifecycle from a type.
attributeBlockDemo.js	■ Creates attribute blocks. ■ Updates attribute block details. ■ Deletes attribute blocks. ■ Assigns, or deassigns, attributes to or from attribute blocks. ■ Lists attribute blocks defined for a type.
attributeDemo.js	■ Creates attributes and assign them to types. ■ Updates attribute details. ■ Deassigns attributes from types. ■ Lists attributes assigned to an object type.
attributeRuleDemo.js	■ Creates attribute update rules. ■ Deletes attribute update rules. ■ Lists attribute update rules.
baselineReplConfigsDemo.js	■ Creates a baseline replication configuration. ■ Creates a remote subordinate. ■ Updates replication configuration details. ■ Updates details of a remote subordinate. ■ Deletes remote subordinates. ■ Deletes replication configurations. ■ Lists baseline replication configurations. ■ Lists remote subordinates defined for a replication configuration.

(Sheet 1 of 6)

Script Name	Description
baselineTemplateDemo.js	■ Creates a baseline template. ■ Modifies baseline template rules. ■ Copies baseline template rules between baseline templates. ■ Deletes a baseline template. ■ Lists all baseline templates defined in the base database. ■ Lists all baselines using a baseline template.
branchDemo.js	■ Creates version branches. ■ Updates branch details. ■ Deletes branches. ■ Lists all defined branches.
browseTemplatesDemo.js	■ Creates a browse template revision. ■ Lists all browse template revisions defined for an object type. ■ Exports or imports browse template contents. ■ Deletes a browse template revision.
buildAreasDemo.js	■ Creates build areas. ■ Updates build area details. ■ Deletes build areas. ■ Lists build areas.
buildStagesDemo.js	■ Assigns, or deassigns, a lifecycle state to a build stage. ■ Lists build stages. ■ Lists lifecycle states assigned to a build stage.
chdocRelTypeDemo.js	■ Creates request relationship types. ■ Updates request relationship type details. ■ Deletes request relationship types. ■ Lists request relationship types defined in a product.
CMRulesDemo.js	■ Defines and updates CM rules for item and request types. ■ Updates CM rules details for item and request types. ■ Enables and/or disables CM rules for item and request types. ■ Lists CM rules defined for item and request types.
copyAttrsDemo.js	■ Copies attribute definitions between object types. ■ Lists attributes assigned to an object type.

(Sheet 2 of 6)

Script Name	Description
copyRulesDemo.js	■ Copies attribute update rules between object types. ■ Lists attribute update rules assigned to an object type.
formatsDemo.js	■ Creates file formats. ■ Updates file format details. ■ Deletes file formats. ■ Lists file formats defined in a base database.
itemLibraryDemo.js	■ Defines a default item library for a product. ■ Defines an item library for an item type. ■ Updates default item library details. ■ Updates item type item library details. ■ Deletes an item library.
itemReplConfigsDemo.js	■ Creates a local or remote subordinate. ■ Creates an item replication configuration. ■ Updates replication configuration details. ■ Updates details of a local or remote subordinate. ■ Deletes replication configurations. ■ Deletes local or remote subordinates. ■ Lists item replication configurations. ■ Lists local and remote subordinates defined for a replication configuration.
itemToItemRelTypeDemo.js	■ Creates an item to item relationship type. ■ Updates item to item relationship types details. ■ Deletes an item to item relationship type. ■ Lists item to item relationship types defined for an item type.
itemTypeGroupDemo.js	■ Creates item type groups. ■ Assigns and deassigns item types to, or from, item type groups. ■ Deletes item type groups. ■ Lists item type groups defined in a base database. ■ Lists release templates using an item type group.
lifecycleDemo.js	■ Creates a lifecycle. ■ Updates lifecycle details. ■ Deletes a lifecycle. ■ Lists all lifecycles defined in a base database. ■ Lists lifecycle state transitions. ■ Lists roles authorized to perform a lifecycle transition.

(Sheet 3 of 6)

Script Name	Description
lifecycleImagesDemo.js	■ Creates new image revisions. ■ Assigns, or deassigns, the default image revision to a lifecycle. ■ Exports images from, or imports images to, a lifecycle image gallery. ■ Deletes image revisions. ■ Lists image revisions assigned to a lifecycle.
lifecycleTemplateAssignmentsDemo.js	■ Creates a lifecycle and browse template assignment at the design part level. ■ Deletes a lifecycle and browse template assignment at the design part level. ■ Lists lifecycle and browse template assignments at the design part level.
lifecycleTransitionsDemo.js	■ Modifies a lifecycle by adding or deleting a normal/off normal transition. ■ Modifies a lifecycle by adding, removing or updating the list of user roles authorized to perform a lifecycle state transition. ■ Modifies a lifecycle by removing a normal state. ■ Modifies a lifecycle by renaming a state. ■ Lists all lifecycles defined in a base database. ■ Lists lifecycle state transitions. ■ Lists roles authorized to perform a lifecycle transition.
primingRelDemo.js	■ Defines a priming relationship. ■ Deletes a priming relationship. ■ Modifies attribute mappings for a priming relationship. ■ Lists request types that may be primed from a request type.
productDemo.js	■ Creates a product. ■ Updates product details. ■ Deletes a product. ■ Lists products defined in a base database.
relationshipNameDemo.js	■ Defines an item type relationship name. ■ Updates item type relationship name details. ■ Deletes item type relationship names. ■ Lists item type relationship names defined in a base database.

(Sheet 4 of 6)

Script Name	Description
releaseTemplateDemo.js	■ Creates a release template. ■ Modifies release template rules. ■ Deletes a release template. ■ Copies release template rules between release templates. ■ Lists all release templates defined in the base database. ■ Lists all releases using a release template.
renameAttributeDemo.js	■ Renames an attribute's ID within the base database.
reportsDemo.js	■ Creates user report definitions. ■ Updates user report definitions. ■ Exports or imports report file contents. ■ Assigns and deassigns a report file to, or from, a user report definition. ■ Deletes user report definitions. ■ Lists user report definitions.
roleAssignmentsDemo.js	■ Adds or removes role assignments. ■ Lists all role assignments for a product.
roleDemo.js	■ Creates a user role. ■ Updates role details. ■ Deletes a role. ■ Lists all roles defined in a base database.
typesDemo.js	■ Creates object types. ■ Updates object type details. ■ Assigns, or deassigns, a lifecycle to or from a type. ■ Deletes object types. ■ Lists object types within a certain scope in a product.
typesUsingAttrsDemo.js	■ Lists all object types using an attribute definition.
validRelTypesDemo.js	■ Creates a valid relationship type. ■ Updates valid relationship type details. ■ Deletes a valid relationship type. ■ Lists valid relationship types defined for a request type.

(Sheet 5 of 6)

Script Name	Description
validsetsDemo.js	■ Creates valid sets. ■ Updates valid sets. ■ Deletes valid sets. ■ Lists valid sets.
validsetValuesDemo.js	■ Appends valid set values. ■ Replaces valid set values. ■ Removes valid set values. ■ Lists valid set values.

(Sheet 6 of 6)

Appendix D

Managing Notification Emails

In this Appendix

About Notification Emails

Overview

The email notification system in Dimensions CM is designed to provide more control over email notification and content, and also to provide a mechanism to allow administrators to configure which particular users/groups receive email notifications. This new notification system is designed to work in tandem with the existing email notification mechanism and if switched on via subscriptions, will replace the standard default Dimensions CM email notifications.

There are a number of standard ("system") notifications defined to which a user or group can be subscribed. (See "Notification Types" on page 344.) This subscription process is performed by an administrator with the appropriate privilege (as described in "Using the Email Notification System" on page 343). Once one or more users have been subscribed to a particular notification then the default emails, which are usually sent to everyone with the appropriate role, will be "switched off" in favor of the new subscriptions. In this way, an administrator can configure users to only receive those notifications that are relevant to them and disable all other "spam" email.

When using the default email notification system, email messages are sent immediately whenever a command is run. This can mean that systems with a slow connection to the email server can suffer from an unnecessary time lag before the command completes. The new email subscription system however, writes every email event to the database to allow it to be processed "on-batch" via a new utility called `dmemail` at a later time. This means that commands are no longer tied to the response time of the email server, and also allows a system administrator much more flexibility as to scheduling when emails are processed.

The `dmemail` utility allows much more fine-grained control over the processing of email than is provided by the default mail system and can be configured to run either as a scheduled task or "on demand". This utility scans all the email events that are pending to be sent, applies the various email templates, and sends the resulting email to the subscribed users. For more information on the dmemail utility see "Using the dmemail Utility" on page 346.

This email subscription model supports the use of email templates to allow the administrator to customize the email content which is sent to the end user. The templating syntax, which is used within the template files, is the same as that used for writing remote jobs and build templates, which means it can support complex IF/ELSE constructs and other control conditions For more information on this syntax see the *Developer's Reference*. The location of the templates is controlled by the variable DM_EMAIL_TEMPLATE_DIR in the Dimensions CM file `dm.cfg`, but by default, is under the `$<DM_ROOT>` directory in a subdirectory called `email_templates`. One template file exists for each subscribable notification, and can be found using the Administration Console. By default, these templates are written to support HTML syntax.

Using the Email Notification System

To take advantage of the new email notification system do the following:

- Subscribe users or groups to the mail events for which you want notifications to be generated via either the command line or the Administration Console.
- Create/edit the required email templates to customize the notification content as desired.
- Edit the dmemail configuration file (`email_config.dat`) and lastly, run the dmemail utility as required.

The process of subscribing users to email notifications using the Administration Console is described in "About Mail Notifications" on page 100, or you can use the following commands below. These are more fully described in the *Command-Line Reference Guide*.

- SUB: Subscribe user/group to notification.
- USUB: Unsubscribe user/group to notification.
- LMNR: List mail notification rules.

The templates supplied, and the variables available, are described on page 354. For more details about the email templating language see the *Developer's Reference.*

The `dmemail` command-line utility is described on page 346. The options for the configuration file are described in page 349.

NOTE Someone who is actually subscribed to this notification will get an email for every request that is closed within the system, regardless what role (s)he holds for that request, or who is the request originator. When request is closed by subscribed user (s)he will not receive notification

Notification Types

These are the predefined email notification types to which users and groups can subscribe.

Email Notification Name	Type	Object Class	Description
CREATE_USER_NOTIFICATION	Create	User	(Administration notification) A new user has been created in the base database.
DELEGATED_ITEM_NOTIFICATION	Delegate	Item	An item has been delegated.
DELEGATED_REQUEST_NOTIFICATION	Delegate	Request	An request has been delegated.
DELETE_USER_NOTIFICATION	Delete	User	(Administration notification) An existing user has been deleted from the base database.
DELIVERY_NOTIFICATION	Deliver	Project	A delivery has been made to a stream.
DEMOTE_BASELINE_NOTIFICATION	Demote	Baseline	A baseline has been demoted.
DEMOTE_ITEM_NOTIFICATION	Demote	Item	An item has been demoted.
DEMOTE_REQUEST_NOTIFICATION	Demote	Request	A request has been demoted.
DEPLOY_BASELINE_NOTIFICATION	Deploy	Baseline	A baseline has been deployed.
DEPLOY_ITEM_NOTIFICATION	Deploy	Item	An item has been deployed.
DEPLOY_NOTIFICATION	Deploy		An item, request, or baseline has been deployed.
DEPLOY_REQUEST_NOTIFICATION	Deploy	Request	A request has been deployed.
ITEM_LOCK_NOTIFICATION	Lock	Item	An item has been locked in a stream.
ITEM_UNLOCK_NOTIFICATION	Unlock	Item	An item deployment has been unlocked in a stream.
MERGE_ITEM_CONFLICT_NOTIFICATION	Merge	Item	An item conflict needs to be merged.
PENDING_BASELINE_NOTIFICATION	Action	Baseline	A baseline has been created or actioned.
PENDING_ITEM_NOTIFICATION	Action	Item	An item has been created or actioned.
PENDING_PROJECT_NOTIFICATION	Action	Project	A project has been created or actioned.
PENDING_REQUEST_NOTIFICATION	Action	Request	A request has been created or actioned.
PROJECT_CONTENT_CHANGED_NOTIFICATION	Action	Project	An operation has changed the content of a project or stream, for example: ■ Create, modify, move, delete, export or remove an item. ■ Create, move or delete a folder.

Email Notification Name	Type	Object Class	Description
PROJECT_LOCK_NOTIFICATION	Lock	Project	A project or stream has been locked.
PROJECT_UNLOCK_NOTIFICATION	Unlock	Project	A project or stream has been unlocked.
PROMOTE_BASELINE_NOTIFICATION	Promote	Baseline	A baseline has been promoted.
PROMOTE_ITEM_NOTIFICATION	Promote	Item	An item has been promoted.
PROMOTE_REQUEST_NOTIFICATION	Promote	Request	A request has been promoted.
REQUEST_CLOSURE_NOTIFICATION	Update	Request	A request has reached its end of lifecycle state. ■ If you use this notification, the new email system is always used (even if you are not subscribed). ■ For request types that have the "Notify Originator on Closure" option set, an email is sent to the originator on closure, even if they are not subscribed. ■ Subscribers receive an email for every request that is closed, regardless of what role they hold for that request or who is the request originator.
ROLLBACK_BASELINE_NOTIFICATION	Rollback	Baseline	A baseline has been rolled back from an area.
ROLLBACK_ITEM_NOTIFICATION	Rollback	Item	An item has been rolled back from an area.
ROLLBACK_NOTIFICATION	Rollback		An item, request, or baseline has been rolled back from an area.
ROLLBACK_REQUEST_NOTIFICATION	Rollback	Request	A request has been rolled back from an area.
UPDATED_ACTION_DESCRIPTION_NOTIFICATION	Update	Request	The action description of a request in a user's inbox has been added or updated.
UPDATED_BASELINE_ATTRIBUTE_NOTIFICATION	Update	Baseline	The attribute of a baseline in a user's inbox has been updated.
UPDATED_DETAILED_DESCRIPTION_NOTIFICATION	Update	Request	The detailed description of a request in a user's inbox has been updated.
UPDATED_FILENAME_NOTIFICATION	Update	Item	The project filename of an item in a user's inbox has been updated.
UPDATED_ITEM_ATTRIBUTE_NOTIFICATION	Update	Item	An attribute of an item in a user's inbox has been updated.
UPDATED_PROJECT_ATTRIBUTE_NOTIFICATION	Update	Project	An attribute of a project or stream in a user's inbox has been updated.

Email Notification Name	Type	Object Class	Description
UPDATED_REQUEST_ATTRIBUTE_NOTIFICATION	Update	Request	An attribute of a request in a user's inbox has been updated.
UPDATED_REQUEST_RELATIONSHIPS_NOTIFICATION	Update	Request	An object has been related, or unrelated, to a request in a user's inbox.
UPDATED_REQUIREMENT_NOTIFICATION	Update	Requirement	A requirement that is related to a request in a user's inbox has been updated.

Email Digesting

When a user or a group is subscribed to a notification, there is the option to make the subscription a digest type, which means that all emails for that user, or group of users, will be "batched up" and sent as a single digest email. This will allow users or groups to receive a summary of all the activity that has been undertaken which matches their subscriptions.

Using the dmemail Utility

What is dmemail?

The `dmemail` utility is a command that processes queued notification events held in the Dimensions CM database and turns them into actual emails. It is responsible for expanding the email templates with the mail content and sends the resulting email to end users.

Modes of Operation

`dmemail` is a command-line utility that can be run in three ways:

- Direct from a command prompt.
- Automated via an existing customer scheduling system, such as `cron`.
- Automated via the Dimensions listener itself.

Security

`dmemail` needs to run from a user account that is part of the Dimensions "ADMIN" group. When `dmemail` is scheduled using the Dimensions listener, then it will run as the same user as the `dmpool` process, which is specified in the file `%DM_ROOT%\dfs\listener.dat` (Windows) or `$DM_ROOT/dfs/listener.dat (UNIX)`.

Command-Line Syntax

The command-line syntax for this utility is:

```
dmemail [config-file]
```

if `config-file` is omitted however, it defaults to:

```
%DM_ROOT%\dfs\email_config.dat (Windows)
or
$DM_ROOT/dfs/email_config.dat (UNIX)
```

Other parameters can either be specified as part of the command line or entered in the configuration file. See "The Configuration Parameters" on page 349 for details of the parameters.

Subsequent parameters when entered on the command line are used as key-value pairs. If you want to use the default configuration filename and have parameters, use "-" as parameter 1.

Parameters entered on the command line take precedence over those options specified in the configuration file.

If you need spaces in the parameters, use double quotes. The parameter names are not case sensitive, and values of Y/y and N/n are equivalent.

Examples:

```
dmemail

dmemail -?

dmemail - debug=y logging=n "hello = greetings tim"
```

dm.cfg Parameters

The following entries are required in the `dm.cfg` file. For details the *System Administration Guide*:

- DM_TMP

  Dimensions temporary file folder. This is used as default for working files, if other (email specific) options are not specified.

- DM_MAILS

  Symbol used by existing Dimensions email system to interface with an Operating System mail command. This is documented in the *General Administration* section of the *System Administration Guide*.

- DM_MAILS_HTTP

  Custom version of the above symbol for use with HTML email. Some mail commands have special options to enable HTML mail. For example, the Dimensions supplied program "pcmsmail" needs to be given the "-i" option. This would be specified in this variable. Example of the last two symbols:

```
DM_MAILS        %DM_PROG%pcmsmail.exe -h "mail-server.com" -f "PCMS"
    -file "%%s" -s "%%s" -t "%%s"
DM_MAILS_HTTP    %DM_MAILS% -i
```

- DM_EMAIL_AUTOSTART_TIMES

  This variable is used to control the automatic startup and running of the dmemail application. It specifies the scheduled times that dmemail is required to start processing content.

```
DM_EMAIL_AUTOSTART_TIMES <spec> <spec> <spec> ........
<spec> : XX:YY | +XX:YY
XX:YY  : process e-mails at exactly XX:YY on the local 24 hour clock.
    (eg 03:00 will run a 3am local time)
+XX:YY : process e-mails every XX:YY hours/minutes, repeatedly. (eg.
    01:00 will run every hour)
```

- DM_EMAIL_AUTOSTART_DIR

  Working files and a log of activity is recorded in this directory when dmpool is managing the dmemail process. The default is DM_TMP.

- DM_EMAIL_MSG_LEVEL

  A severity level indicating the level of event that will generate a record in the log. 0 generates the most information. Higher values restrict the messages to more severe events. To only see information about serious errors, set this to 4.

- DM_EMAIL_TEMPLATE_DIR

  This overrides the default location of the email templates, if required. If not specified, the default is `%DM_ROOT%\email_templates` (Windows) or `$DM_ROOT/email_templates` (UNIX).

The Configuration Parameters

These configuration parameters can be specified in the file `email_config.dat` or in the command line. They are case-insensitive, and are specified as `keyword=value`. A "#" indicates a comment in this file, and blank lines are allowed.

These keywords and their parameters are described in the sections that follow, grouped under functional headings.

Database Processing

- DATABASE

  Multiple databases can be processed in sequence, by specifying a keyword "DATABASE_i" where i is an integer (such as 1,2 3) for each subsequent database to be processed. In each case, the DATABASE key specifies the usual Dimensions CM "connection string" for a database, i.e. as you would for DMDB. If the database requires a password to allow the user ID to connect, then this can be set using the dmpasswd utility documented in the *Command-Line Reference Guide*.

  Example:

  ```
  DATABASE_1 =intermediate@dim10
  DATABASE_2=intermediate@dim10-test
  DATABASE_3=db2user/db2passwd:intermediate@dim10-db2-test
  ```

  ```
  HOST
  DMUSER
  ```

- HOST and DMUSER

  These are specs for a Dimensions server and user ID. This is only required for Dimensions RM functionality. In general operation, `dmemail` does not require a Dimensions server to be running, only the database to be available.

  ```
  HOST_<i>
  DMUSER_<i>
  ```

  These operate similarly to DATABASE_<i> in the case where multiple databases are to be processed.

Processing Options

- PROCESS_FILES

  Y or N

  Indicates if `dmemail` should process the stored "email files" and physically "post" them to the email system. Setting this to N is very useful in testing to avoid a flood of unwanted emails.

  Temporary files are created, used, and then deleted for each email. During a long run of dmemail, only a handlful of temporary files should exist.

Debugging Options

These options should only be changed if requested by Support.

These debug options are probably best specified in the command line. if you have set debug=y, then the options in effect are printed in the log.

- PROCESS_ROWS

  Y or N

  Indicates if `dmemail` should generate new "email files" on disk from the event record rows in the database. Setting this to N means that new events are not processed, which is helpful if you are working on fixing a problem with existing emails.

- DELETE_ROWS

  Y or N

  Indicates if dmemail should delete rows from the database once they have been dealt with. This is essential in a live operation, or the same email will be regenerated repeatedly. However, in testing it may be useful to keep the data for a second test.

- DELETE_FILES

  Y or N

  Indicates if dmemail should delete the physical "email files" after posting them. This is essential in a live operation, or the same email will be generated repeatedly. However, in testing it may be useful to avoid deleting the body files, so you can inspect them for testing purposes

- LOGGING

  Y or N

- LOG_FILE

  <filename>

- DEBUG

  Y or N

  Add very detailed information to output, including the "symbol table" contents. The symbol table dump will contain details of all symbols available for expansion in the template.

Locating Email Templates

- TPL_SEARCH_<i>

  Where <i> is a small number starting at 0, and is used to specify a list of locations where email templates can be found. The reason for allowing a list of locations is so that custom versions of supplied templates can be stored in a local directory that is searched ahead of the system directory.

- LOCALE_MAP_<i>
- LOCALE_DIR_<i>

  Where <i> is a small number, starting at zero. For each value of i, these two symbols specify a mapping from a "locale name" to a directory name.

The locale name is a string defining the preferred "language" for a particular email recipient. The resulting directory name is then added to the front of the search list defined above. This allows certain email templates to be redesigned in a language-sensitive way, while at the same time allowing the default templates to be used if no such customizing has been provided.

example:

```
TPL_SEARCH_0=c:\custom\email_templates
TPL_SEARCH_1=c:\Serena\Dimensions\14.4\email_templates
LOCAL_MAP_0 =GERMAN
LOCAL_DIR_0 =c:\custom\email_templates\german
```

In this example, a template will be searched for in the following places:

```
default:c:\custom\email_templates
c:\Serena\Dimensions\14.4\email_templates
German:c:\custom\email_templates\german
c:\custom\email_templates
c:\Serena\Dimensions\14.4\email_templates
```

Miscellaneous Options

- MAX_EMAIL_LIST_SIZE

  Integer

  Emails are sent using a shell command line, which may have a limited length. If this (MAX_EMAIL_LIST_SIZE) is set to a fairly low value (in characters), then when large distributions are generated, they will be split into multiple emails, each with a smaller list of recipients.

- DEFAULT_SUBJECT

  Specifies a default subject line that will be used if the template being processed does not specify one. Most templates will set the subject line, as this allows variables (such as item names) to be show in the subject itself.

- DEFAULT_SUBJECT_MULTIPLE

  When multiple emails are merged together in a digest structure, this variable specifies what the default subject will be.

- MULTIPLE_TEMPLATE

  This names a special template that is a "container" that holds the individual pieces of a merged (digest) email. It can be customized to achieve certain formatting effects. The supplied version, called "multiple_notifications.tplt", uses a simple HTML splitter bar between individual email bodies. Consult this example for further information.

- ENVELOPE

  This names a special template which is used like an envelope, wrapping up the email body just prior to posting. The supplied example, "html-envelope-outer.tplt" adds the necessary HTML syntax to the top and bottom of the email, to make a valid HTML message. It could be customized to add customer-specific branding to all Dimensions CM emails.

- EMAIL_SLEEP

  In some configurations, SMTP servers cannot respond quickly enough to a large number of emails, such as may be generated by dmemail. This could, for example, be the result of an anti-spam policy implemented by the SMTP server. This parameter specifies a value in seconds which is used as a delay between sending messages. By default, this figure is set to 30. If you do not have any such spam policy in place, then it is highly recommended that you lower this to a figure of 1 or 2.

- EMAIL_DIRECTORY

  Emails are first created as files, in this directory. If this is not specified, the default at DM_TMP will be sent.

- DIGEST

  This controls the way emails are digested.

  0 - No digesting for super-fast processing of an email backlog. This directly sends each row.

  1 - (default) The current behavior based on digest flag against user subscription

  2 - Forced mode - all messages will be digested, to produce only one email per user no matter what. This could be used to ease the burden on the mail server in a backlog situation.

- DIGEST_THRESHOLD

  This sets the limit for the number of messages to be digested.

  Numeric value, default 10000.

  If the number of rows to be processed is larger than this, then dmemail refuses to work unless DIGEST=0 is in effect. This limit can be overridden when large runs are required.

- FILTER_SQL

  This specifies a "sql where clause component" that will be used to select rows from the MAILS2PROCESS table. An administrator can use this to great effect to test out situations on a live system without sending thousands of mails.

  Here are some examples:

  dmemail - "filter_sql=obj_uid=12345"

  dmemail - "filter_sql=create_date between xxxx and yyyyy"

  dmemail - "filter_sql=rel_class = 7"

  The full SQL generated is shown in the log when debug=y.

  here is a typical test:

  dmemail - process_files=n delete_files=n "filter_sql=obj_uid>12345"

- PURGE_ROWS

  Y/N

  If Y, then rows that are ignored by the filter used above, are purged from the table. If the FILTER_SQL specified a condition that requested only recent rows, then this would delete all the older rows.

  This works by inverting the condition specified with FILTER_SQL

- NO_STOPFILE

  Y/N

  Specifying this in the config file will allow the command line program to use scheduling mode, just as if it had been started from the Dimensions pool process. The syntax is:

  ```
  dmemail -dmpool
  ```

  For this to work, the `dm.cfg` variables that control scheduling need to be setup.

- ALLOW_MISSING_SYMBOLS

  If this is set to Y, then the email template will expand, even if some variables are not defined. This is very useful for processing a batch of emails which have a known minor problem (such as a missing description). In place of the symbol, text will be generated as in the following example:

  Template before expansion:

  ```
  Request: %DMTEST.
  ```

  Email after expansion:

  ```
  Request: [symbol DMTEST is not defined]
  ```

Notification Templates

The default email templates are located in: <DM_ROOT>/email_templates/

Email Notification Name	Template Name
CREATE_USER_NOTIFICATION	CREATE_USER_NOTIFICATION.tplt
DELEGATED_ITEM_NOTIFICATION	DELEGATED_ITEM_NOTIFICATION.tplt
DELEGATED_REQUEST_NOTIFICATION	DELEGATED_REQUEST_NOTIFICATION.tplt
DELETE_USER_NOTIFICATION	DELETE_USER_NOTIFICATION.tplt
DELIVERY_NOTIFICATION	DELIVERY_NOTIFICATION.tplt
DEMOTE_BASELINE_NOTIFICATION	PROMOTE_BASELINE_NOTIFICATION.tplt
DEMOTE_ITEM_NOTIFICATION	PROMOTE_ITEM_NOTIFICATION.tplt
DEMOTE_REQUEST_NOTIFICATION	PROMOTE_REQUEST_NOTIFICATION.tplt
DEPLOY_BASELINE_NOTIFICATION	DEPLOY_BASELINE_NOTIFICATION.tplt
DEPLOY_ITEM_NOTIFICATION	DEPLOY_ITEM_NOTIFICATION.tplt
DEPLOY_NOTIFICATION	DEPLOY_NOTIFICATION.tplt
DEPLOY_REQUEST_NOTIFICATION	DEPLOY_REQUEST_NOTIFICATION.tplt
DEPLOY_REQUEST_NOTIFICATION	DEPLOY_REQUEST_NOTIFICATION.tplt
ITEM_LOCK_NOTIFICATION	ITEM_LOCK_NOTIFICATION.tplt
ITEM_UNLOCK_NOTIFICATION	ITEM_UNLOCK_NOTIFICATION.tplt
MERGE_ITEM_CONFLICT_NOTIFICATION	MERGE_ITEM_CONFLICT_NOTIFICATION.tplt
PENDING_BASELINE_NOTIFICATION	PENDING_BASELINE_NOTIFICATION.tplt
PENDING_ITEM_NOTIFICATION	PENDING_ITEM_NOTIFICATION.tplt
PENDING_PROJECT_NOTIFICATION	PENDING_PROJECT_NOTIFICATION.tplt
PENDING_REQUEST_NOTIFICATION	PENDING_REQUEST_NOTIFICATION.tplt
PROJECT_CONTENT_CHANGED_NOTIFICATION	PROJECT_CONTENT_CHANGED_NOTIFICATION.tplt
PROJECT_LOCK_NOTIFICATION	PROJECT_LOCK_NOTIFICATION.tplt
PROJECT_UNLOCK_NOTIFICATION	PROJECT_UNLOCK_NOTIFICATION.tplt
PROMOTE_BASELINE_NOTIFICATION	PROMOTE_BASELINE_NOTIFICATION.tplt
PROMOTE_ITEM_NOTIFICATION	PROMOTE_ITEM_NOTIFICATION.tplt
PROMOTE_REQUEST_NOTIFICATION	PROMOTE_REQUEST_NOTIFICATION.tplt
REQUEST_CLOSURE_NOTIFICATION	REQUEST_CLOSURE_NOTIFICATION.tplt
ROLLBACK_BASELINE _NOTIFICATION	DEPLOY_BASELINE _NOTIFICATION.tplt
ROLLBACK_ITEM_NOTIFICATION	DEPLOY_ITEM_NOTIFICATION.tplt
ROLLBACK_NOTIFICATION	DEPLOY_NOTIFICATION.tplt
ROLLBACK_REQUEST_NOTIFICATION	DEPLOY_REQUEST_NOTIFICATION.tplt
UPDATED_ACTION_DESCRIPTION_NOTIFICATION	UPDATED_ACTION_DESCRIPTION_NOTIFICATION.tplt
UPDATED_BASELINE_ATTRIBUTE_NOTIFICATION	UPDATED_BASELINE_ATTRIBUTE_NOTIFICATION.tplt
UPDATED_DETAILED_DESCRIPTION_NOTIFICATION	UPDATED_DETAILED_DESCRIPTION_NOTIFICATION.tplt
UPDATED_FILENAME_NOTIFICATION	UPDATED_FILENAME_NOTIFICATION.tplt

Email Notification Name	Template Name
UPDATED_ITEM_ATTRIBUTE_NOTIFICATION	UPDATED_ITEM_ATTRIBUTE_NOTIFICATION.tplt
UPDATED_PROJECT_ATTRIBUTE_NOTIFICATION	UPDATED_PROJECT_ATTRIBUTE_NOTIFICATION.tplt
UPDATED_REQUEST_ATTRIBUTE_NOTIFICATION	UPDATED_REQUEST_ATTRIBUTE_NOTIFICATION.tplt
UPDATED_REQUEST_RELATIONSHIPS_NOTIFICATION	UPDATED_REQUEST_RELATIONSHIPS_NOTIFICATION.tplt
UPDATED_REQUIREMENT_NOTIFICATION	UPDATED_REQUIREMENT_NOTIFICATION.tplt

The table below lists the variables that are valid for each email notification:

NOTE System-defined and user-defined attributes are associated with a particular object type. You can only use attributes in a notification template that belong to the object type for which that notification is generated.

Email Notification Name	Variable Name	Variable Description
CREATE_USER_NOTIFICATION	MAIL_REASON	The reason for the email.
DELEGATED_ITEM_NOTIFICATION	CLASS_TYPE	The class of the object, e.g. item.
DELEGATED_REQUEST_NOTIFICATION	NOTIFICATION_ID	The name of the notification.
DELETE_USER_NOTIFICATION	OBJECT_SPEC	The specification of the object.
DELIVERY_NOTIFICATION	FILE_OPERATIONS	List of file operations.
DEPLOY_BASELINE_NOTIFICATION DEPLOY_ITEM_NOTIFICATION DEPLOY_NOTIFICATION DEPLOY_REQUEST_NOTIFICATION ROLLBACK_BASELINE_NOTIFICATION ROLLBACK_ITEM_NOTIFICATION ROLLBACK_NOTIFICATION ROLLBACK_REQUEST_NOTIFICATION	NOTIFICATION_ID	The name of the notification.
	MAIL_REASON	The reason for the email.
	CLASS_TYPE	The class of the object, e.g. item.
	OBJECT_SPEC	The specification of the object.
	OBJECT_DESCRIPTION	The description of the object.
	USER_COMMENT	The user comment supplied.
	PROJECT_SPEC	The specification of the project or stream.
	AREA_ID	The name of the area deployed to.
	AREA_JOB_STATUS	The status of the job.
	AREA_JOB_MESSAGE	The message from the job.
ITEM_LOCK_NOTIFICATION ITEM_UNLOCK_NOTIFICATION	ATTR_*	User defined attributes for the target item.
	ATTRS_*	System defined attributes for the target item.
MERGE_ITEM_CONFLICT_NOTIFICATION	OBJECT_DESCRIPTION	The description of the object.

Email Notification Name	Variable Name	Variable Description
PENDING_BASELINE_NOTIFICATION	ATTR_*	User-defined attributes. **Note:** Attributes like this have user defined names, descriptions and values. This information is delivered to the template in an array. The first element of the array is the description, for example, `ATTR_SOMETHING(0)`. The second element of the array is the actual value of that attribute for this instance, which is coded as `ATTR_SOMETHING(1)`. There will not be `ATTR_SOMETHING(2)`.
PENDING_ITEM_NOTIFICATION	ATTRS_*	System-defined attributes. See information about the templating language in the *Developer's Reference*.
PENDING_PROJECT_NOTIFICATION	MAILING_USERS	The users who are being emailed.
	IMPACTED_PARTS	Contains the design parts affected by the request (optionally scoped by a product if `MAIL_OPTION` is 4).
	IMPACTED_ITEMS	Contains the items related to the request (optionally scoped by a product if `MAIL_OPTION` is 4)
PENDING_REQUEST_NOTIFICATION	MAIL_SUBJECT	The email subject.
PROJECT_CONTENT_CHANGED_NOTIFICATION	FILE_OPERATIONS	List of file operations.
PROJECT_LOCK_NOTIFICATION PROJECT_UNLOCK_NOTIFICATION	ATTR_*	User defined attributes for the target project.
	ATTRS_*	System defined attributes for the target project.
REQUEST_CLOSURE_NOTIFICATION	ATTR_*	User defined attributes for the target request.
	ATTRS_*	System defined attributes for the target request.
ROLLBACK_ITEM_NOTIFICATION	IMPACTED_PARTS	Contains the design parts affected by the request (optionally scoped by a product if `MAIL_OPTION` is 4).
	IMPACTED_ITEMS	Contains the items related to the request (optionally scoped by a product if `MAIL_OPTION` is 4)

Email Notification Name	Variable Name	Variable Description
PROMOTE_BASELINE_NOTIFICATION PROMOTE_ITEM_NOTIFICATION PROMOTE_REQUEST_NOTIFICATION DEMOTE_BASELINE_NOTIFICATION DEMOTE_ITEM_NOTIFICATION DEMOTE_REQUEST_NOTIFICATION	NOTIFICATION_ID	The name of the notification.
	MAIL_REASON	The reason for the email.
	CLASS_TYPE	The class of the object, e.g. item.
	OBJECT_SPEC	The specification of the object.
	OBJECT_DESCRIPTION	The description of the object.
	CURRENT_USER	The current user doing the operation.
	PREVIOUS_STAGE	The stage the object was promoted from.
	NEW_STAGE	The stage the object was promoted to.
	USER_COMMENT	The user comment supplied.
	NEXT_STAGE	The next possible stage(s) the object can be promoted to.
	PENDING_USERS	The user(s) who have the object in their inbox.
UPDATED_ACTION_DESCRIPTION_NOTIFICATION	MAIL_REASON	The reason for the email.
UPDATED_BASELINE_ATTRIBUTE_NOTIFICATION	PRODUCT	The product ID.
UPDATED_DETAILED_DESCRIPTION_NOTIFICATION	CLASS_TYPE	The class of the object, e.g. item.
UPDATED_FILENAME_NOTIFICATION	OBJECT_ID	The object ID.
UPDATED_ITEM_ATTRIBUTE_NOTIFICATION	USER_COMMENT	A user comment, if specified.
UPDATED_PROJECT_ATTRIBUTE_NOTIFICATION	CURRENT_OWNER	The current owner.
	CURRENT_USER	The current user doing the operation.
	VARIANT	The object variant.
	REVISION	The object's revision.
	PREVIOUS_STATUS	The object's previous status.
	CURRENT_STATUS	The object's current status.
	ROLE	The delegated role.
	NEW_STATUS	The object's new status.
	NEXT_STATUS	The object's next status.
	ORIGINATOR	The object's originator.
	ACTION_DATE	The object's action date.
	CREATE_DATE	The object's creation date.
	UPDATE_DATE	The object's update date.
	OBJECT_TYPE	The object's type name.
	ACTION_USER	The user actioning the object.
	PENDING_USERS	The user(s) who have the object in their inbox.
	STAGE_ID	The object's stage ID.
	OWNER_SITE	The object's owning site.

Email Notification Name	Variable Name	Variable Description
con..	WEB_URL	A HTTP URL for accessing the object.
	MAIL_SUBJECT	The mail subject.
UPDATED_REQUEST_ATTRIBUTE_NOTIFICATION	OBJECT_SPEC	The specification of the object.
UPDATED_REQUEST_RELATIONSHIPS_NOTIFICATION	OBJECT_DESCRIPTION	The description of the object.
UPDATED_REQUIREMENT_NOTIFICATION	ATTR_*	User defined attributes.
	ATTRS_*	System defined attributes. See information about the templating language in the *Developer's Reference.*
	MAILING_USERS	The users who are being emailed.
	MAILING_DIGEST	If these are digested notifications.
	PRODUCT	The product ID.
	OBJECT_ID	The object ID.
	PREVIOUS_STATUS	The object's previous status, if applicable
	CURRENT_STATUS	The object's current status.
	CURRENT_OWNER	The current owner.
	CURRENT_USER	The current user doing the operation.
	ROLE	The delegated role.
	NEW_STATUS	The object's new status.
	NEXT_STATUS	The object's next status.
	ORIGINATOR	The object's originator.
	ACTION_DATE	The object's action date.
	CREATE_DATE	The object's creation date.
	UPDATE_DATE	The object's update date.
	OBJECT_TYPE	The object's type name.
	ACTION_USER	The user actioning the object.
	PENDING_USERS	The user(s) who have the object in their inbox.
	OWNER_SITE	The site owning the object.
	DELEGATED_SITE	The site delegated the object.
	WEB_URL	A HTTP URL for accessing the object.
	MAIL_OPTION	Mail option being performed.
	RELATED_OBJECTSPEC	The related object specification.
	ITEM_ID	The item ID if applicable.
	RELATED_CLASS	The related object class.
	RELATE_OPERATION	The relate operation being performed.
	VARIANT	The variant of the object.
	REVISION	The revision of the object.

Email Notification Name	Variable Name	Variable Description
n/a	RTM_DATABASE	Only applicable to users of RM and cover RM object details.
	RTM_PROJECT	
	RTM_CLASS	
	RTM_OBJECT_ID	
	RTM_MAIL_REASON	
	RTM_MAIL_MESSAGE	

Please note, not all of the variables specified above will be populated under all circumstances. Trying to use variables that are not populated under all circumstances will result in either the variable being given a blank value, or a template error when trying to process that template. Please see below for more details on how to resolve such issues.

Multi-Value Attributes

The following variables can be used for multi-value attributes:

- ATTR_<name>_VALS for user-defined attribute values
- ATTRS_<name>_VALS for system defined attribute values
- where <name> is the attribute name.

These can be used for single-value or multi-value attributes.

For example:

```
)IFDEF ATTR_<name>
   <P>List of ~(ATTR_<name>(0)). values:<BR>
   <UL>
      )REP ATTR_<name>_VALS
      <LI>~ATTR_<name>_VALS.</LI>
      )ENDR
   </UL></P>
)ENDIF
```

If you wish to display a multi-field multi-valued attribute, you would need to display the grouped multi-value attributes that it consists of.

Administration and Error Handling

If you customize your email templates, you must ensure that the changes you have made contain valid variables and comply to the necessary templating syntax. You can check this by running dmemail with the debugging symbols set (see "Debugging Options" on page 350) and running the utility in command-line mode after events which fire the modified email template have been performed. Running in this mode, dmemail will send output to the console window detailing exactly what it is doing. This will also include any errors encountered while expanding the templates. If any error is encountered while trying to process a mail event, then dmemail will skip that event and move on to process the others This means that if your users are not receiving the emails that you would expect,

then you might have introduced errors into the templates that need to be corrected as described above.

NOTE Please be aware that some mail servers implement anti-spam policies by filtering out invalid email addresses. When using mail notifications, please ensure that all the email addresses specified for your users are correct (see User Registration in the Administration Console). Failure to do so may result in email notifications being blocked by the mail server.

Appendix E

Dimensions CM Privileges

In this Appendix

Privilege Overview

In Dimensions® CM 14.x all implicit privilege definitions that used to be encapsulated in role definitions and inboxes have now been made explicit and grantable to users, roles, or groups based on certain rules. This is to allow the maximum level of flexibility when allowing certain users to do things. This appendix lists what those privileges are and what rules can be used to say when those privileges apply. It is important to note that certain system roles like PRODUCT-MANAGER, CHANGE-MANAGER, PARTS-CONTROLLER, and WORKSET-MANAGER amongst others, still retain their implicit abilities, although in some circumstances these abilities can be changed by altering the default privilege set up.

Privilege Rules

This appendix lists all the privileges that are available in Dimensions CM together with the available rules for each privilege. Each rule specifies the circumstances under which a privilege is to be activated for a user. Not all of the rules are available for all of the privileges.

The procedure for assigning privilege rules in the Administration Console is described in "Managing Privileges" on page 96.

The complete set of privilege rules is listed below together with their descriptions.

Name	Rule ID	Description
Explicitly deny to users/ groups	USER_DISABLE	The privilege will be explicitly denied for the users/ groups listed.
Explicitly grant to users/ groups	USER_ENABLE	The privilege will be explicitly granted for those users/ groups listed.
Grant to all users	ANYUSER	The privilege will be granted to everyone.
Object is in the user's inbox or the user has current role	OBJ_PEND	The privilege will be granted if the object (a container or category) for which the general grant rule is specified is in the user's inbox or the user has a role on a state transition from the object's current state. For example, a project is in the user's inbox or the user has a role on a state transition from the project's current state. **Note:** When the privilege is *action to next state* (for baselines, requests, items, or project/streams) the "or" in the above rule becomes "and". Regarding items and baselines and the behavior of privileges at the initial lifecycle state, the object when first created will only be in the inbox of the originator and therefore only that user, or by default, those in the ADMIN group, can perform the operations required, such as update item, until it is actioned on.
User has any role on any product	ANYROLE_DB	The privilege will be granted if the user has any role on any product in the base database.
User has any role on any transition of the whole stage lifecycle	ROLE_STAGE_LIFECYCLE	The privilege will be granted if the user holds any role on any transition in the whole Global Stage Lifecycle.

Name	Rule ID	Description
User has any role on the design part owning the object	ANYROLE_PART	The privilege will be granted if the user has any role on the design part that owns the object for which the general grant rule is specified.
User has any role on the initial lifecycle state transition	ROLE_INITIAL_LIFECYCLE	The privilege will be granted if the user holds any role on the initial state transition in the lifecycle of the object for which the general grant rule is specified.
User has any role on the object lifecycle	ROLE_LIFECYCLE	The privilege will be granted if the user holds any role on any transition in the whole lifecycle of the object for which the general grant rule is specified.
User has any role on the product owning the object	ANYROLE_PRODUCT	The privilege will be granted if the user has any role on the product owning the object for which the general grant rule is specified.
User has one or more of the following roles on any product	ROLES_DB	The privilege will be granted if the user holds one of the listed roles on any product in the base database.
User has any role on the transition from the current stage to the selected stage	ROLE_STAGE	The privilege will be granted if the user has any role on the transition from the current stage to the selected stage
User has any role on the transition from the current stage to the selected stage	ROLES_STAGE	The privilege will be granted if the user holds one of the listed roles on any transition to the current stage in the Global Stage lifecycle.
User has one or more of the following roles on the design part	ROLES_PART	The privilege will be granted if the user holds one of the listed roles on the design part owning the object for which the general grant rule is specified.
User has one or more of the following roles on the product	ROLES_PRODUCT	The privilege will be granted if the user holds one of the listed roles on the product owning the object for which the general grant rule is specified.
User has one or more of the following roles on the transition from the current stage to the next stage in the stage lifecycle	ROLES_NEXT_STAGE	The privilege will be granted if the user holds one of the listed roles on the transition from the current stage to the next stage in the Global Stage lifecycle.
User is the originator of the object	ORIGINATOR_OBJ	The privilege will be granted if the current user is the originator/creator of the object for which the general grant rule is specified.

CAUTION! Default Grant Rules for *Update and Deliver* Privileges

When you create a new base database or upgrade an existing Dimensions 9 installation, the default grant rules for the **Update Files from Project/Stream** and **Deliver Files into Project/Stream** privileges include the **User holds any role on the product owning the object** rule. As a result, there is a security issue where certain users are able to download and upload files from any project in the product including those to which they should not have access. To correct this, you must remove the **User holds any role on the product owning the object rule** from the grant rules for the **Update Files from Project/Stream** and **Deliver Files into Project/Stream** privileges.

Privilege Reference

This section describes all of the different privileges, and lists the rules that can apply to each. For a detailed description of each rule, see "Privilege Rules above".

There are several different types of privileges described:

Administration privileges apply to all products in the base database. The other, product level privileges only apply to your currently selected product.

Note About Roll Back Privileges

Access to all roll back operations is controlled through the use of the item roll back privilege. Although they are displayed through the user interface, the roll back request and roll back baseline privileges are not used, and will be ignored. Future releases of Dimensions will provide a single roll back privilege to control access to these operations.

Note About Default Grant Rules for Download Files from Project Privilege

When you create a new base database, the default grant rules for the Update Files from Project/Stream and Deliver Files into Project/Stream privileges include the "User holds any role on the product owning the object" rule. As a result, there is a security issue where certain users are able to download and upload files from any project in the product including those to which they should not have access. To correct this, you must remove the "User holds any role on the product owning the object" rule from the grant rules for "Update Files from Project/Stream" and "Deliver Files into Project/Stream" privileges.

Administration Privileges

Administration privileges apply to all products in the base database.

Name	ID	Description
Create Deployment Areas	ADMIN_CREATE_DEPLOY_AREA	This privilege allows you to create a deployment area.
Create Library Cache Areas	ADMIN_CREATE_LIBCACHE_AREA	This privilege allows you to create a library cache area.
Create Products	ADMIN_CREATE_PRODUCT	This privilege allows you to create products.
Create Work Areas	ADMIN_CREATE_WORK_AREA	This privilege allows you to create a work area.

Name	ID	Description
Delete Deployment Areas	ADMIN_DELETE_DEPLOY_AREA	This privilege allows you to delete a deployment area.
Delete Library Cache Areas	ADMIN_DELETE_LIBCACHE_AREA	This privilege allows you to delete a library cache area.
Delete Work Areas	ADMIN_DELETE_WORK_AREA	This privilege allows you to delete a work area.
Manage and View Other Users' Lists	ADMIN_OTHER_PENDLIST	This privilege allows you to view other users' lists.
Manage Baseline and Release Templates	ADMIN_TEMPLATEMAN	This privilege allows you to manage baseline and release template definitions.
Manage Build Configurations	ADMIN_BUILDMAN	This privilege allows you to manage build configurations.
Manage Customer Definitions	ADMIN_CUSTDEFMAN	This privilege allows you to manage customer definitions.
Manage Email Notifications	ADMIN_EMAILNOTIFY_SUBSCRIBE	This privilege allows you to manage email notifications.
Manage File Format Definitions	ADMIN_FORMATMAN	This privilege allows you to manage file formats for items and requests.
Manage Item Relationship Names	ADMIN_ITEMRELSMAN	This privilege allows you to manage item relationships.
Manage Lifecycles	ADMIN_LIFECYCLEMAN	This privilege allows you to manage lifecycle definitions.
Manage Mover Deployment Definitions	ADMIN_MOVER_DEPLOY	This privilege allows you to manage deployments via Serena Mover.
Manage Network Definitions	ADMIN_NETWORK	This privilege allows you to manage the Dimensions network definitions.
Manage Privileges	ADMIN_PRIVILEGEMAN	This privilege allows you to grant admin and non-admin privileges.
Manage Public Queries	ADMIN_PUBLICQUERIES	This privilege allows you to manage public based queries.
Manage Replication Configurations	ADMIN_REPL	This privilege allows you to manage Replication configuration definitions, run replicator, and also run pdiff
Manage Role Definitions	ADMIN_ROLEMAN	This privilege allows you to manage role definitions.
Manage Upload Rules	ADMIN_UPLOADMAN	This privilege allows you to manage upload and item/request format definitions.
Manage User Interface Profiles	ADMIN_UI_PROFILES	This privilege allows you to manage user interface profiles.
Manage User Report Configurations	ADMIN_REPORTMAN	This privilege allows you to manage user report configurations and definitions.
Manage Users and Group Definitions	ADMIN_USERMAN	This privilege allows you to manage users and groups for a database.

Name	ID	Description
Manage Version Branch Definitions	ADMIN_BRANCHMAN	This privilege allows you to manage the version branches in the database.
Publishing Preferences	ADMIN_PUBLISHPREFS	This privilege allows you to publish a set of default preferences.
Run Admin Reports	ADMIN_RUN_REPORT	This privilege allows you to run admin reports and other reports.
Update Deployment Area Properties	ADMIN_UPDATE_DEPLOY_AREA	This privilege allows you to update deployment area properties such as hostname, area owner user and password.
Update Library Cache Area Properties	ADMIN_UPDATE_LIBCACHE_AREA	This privilege allows you to update library cache area properties such as hostname, area owner user and password.
Update Work Area Properties	ADMIN_UPDATE_WORK_AREA	This privilege allows you to update work area properties such as hostname, area owner user and password.

Baseline Level Privileges

Name	ID	Description
Action to Any State	BASELINE_ACTION_ANYSTATE	This privilege allows you to action a baseline to any state in the lifecycle.
Action to Next State	BASELINE_ACTION_NEXTSTATE	This privilege allows you to action a baseline to the next state in the lifecycle. **Note:** When the rule is *Object is in the user's inbox or the user has current role* (OBJ_PEND) the "or" in this rule becomes "and". Regarding items and baselines and the behavior of privileges at the initial lifecycle state, the object when first created will only be in the inbox of the originator and therefore only that user, or by default, those in the ADMIN group, can perform the operations required, such as update item, until it is actioned on.
Build from a Baseline	BASELINE_BUILD	This privilege allows you to build from a baseline.
Create	BASELINE_CREATE	This privilege allows you to create baselines.
Create Archive	BASELINE_CREATE_ARCHIVE	This privilege allows you to create an archive (ART) within the product.
Delete	BASELINE_DELETE	This privilege allows you to delete baselines.
Delete Archive	BASELINE_DELETE_ARCHIVE	This privilege allows you to delete an archive (ART) from your product.
Demote to Any Stage	BASELINE_DEMOTE_ANYSTAGE	This privilege allows you to demote baselines to the any demotion stage in the lifecycle
Demote to Next Stage	BASELINE_DEMOTE_NEXTSTAGE	This privilege allows you to demote baselines to the next demotion stage in the lifecycle
Deploy to Areas	BASELINE_DEPLOY	This privilege allows you to deploy baselines to areas
Promote to Any Stage	BASELINE_PROMOTE_ANYSTAGE	This privilege allows you to promote baselines to the any promotion stage in the lifecycle
Promote to Next Stage	BASELINE_PROMOTE_NEXTSTAGE	This privilege allows you to promote baselines to the next promotion stage in the lifecycle
Relate Baseline to Baseline	BASELINE_RELATE_BASELINE	This privilege allows you to relate/unrelate a baseline to/from another baseline.
Rename	BASELINE_RENAME	This privilege allows you to rename baselines.

Name	ID	Description
Rollback from Areas	BASELINE_ROLLBACK	**Note**: This privilege does not have any effect, and should not be used. See "Note About Roll Back Privileges" on page 364.
Transfer Baseline In	BASELINE_TRANSFER_IN	This privilege allows you to transfer baselines into a product (ART).
Transfer Baseline Out	BASELINE_TRANSFER_OUT	This privilege allows you to transfer baselines out from a product (ART).
Update	BASELINE_UPDATE	This privilege allows you to edit baseline attributes.
Update Files from Baseline	BASELINE_DOWNLOAD	This privilege allows you to update files in your work area from a baseline.

Item Level Privileges

Name	ID	Description
Action to Any State	ITEM_ACTION_ANYSTATE	This privilege allows you to action an item to any state in the lifecycle.
Action to Next State	ITEM_ACTION_NEXTSTATE	This privilege allows you to action items to the next state in the lifecycle. **Note:** When the rule is *Object is in the user's inbox or the user has current role* (OBJ_PEND) the "or" in this rule becomes "and". Regarding items and baselines and the behavior of privileges at the initial lifecycle state, the object when first created will only be in the inbox of the originator and therefore only that user, or by default, those in the ADMIN group, can perform the operations required, such as update item, until it is actioned on.
Archive	ITEM_ARCHIVE	This privilege allows you to archive items and retrieve previously archived items (ART).
Break Lock	ITEM_UNLOCK_OTHER	This privilege allows you to unlock an item previously locked by another user in that stream.
Browse	ITEM_BROWSE	This privilege allows you to browse an item or search its content.
Create	ITEM_CREATE	This privilege allows you to create items. **Note:** If a user has the necessary privilege to create an item, but they do not hold the necessary role on the initial lifecycle state, Dimensions will temporarily grant them that role, thus enabling the create operation to complete successfully. In such a case, a warning message will be displayed, and when the user actions the item to the next state the temporary role will be deleted.
Delegate	ITEM_DELEGATE	This privilege allows you to delegate items to users and groups.
Delete	ITEM_DELETE	This privilege allows you to delete items.
Demote to Any Stage	ITEM_DEMOTE_ANYSTAGE	This privilege allows you to demote items to the any demotion stage in the lifecycle
Demote to Next Stage	ITEM_DEMOTE_NEXTSTAGE	This privilege allows you to demote items to the next demotion stage in the lifecycle
Deploy to Areas	ITEM_DEPLOY	This privilege allows you to deploy items to areas
Lock/Unlock	ITEM_LOCK	This privilege allows you to lock or unlock items in a stream
Move Item to Another Design Part	ITEM_MOVE_PART	This privilege allows you to move an item to a different design part.

Name	ID	Description
Promote to Any Stage	ITEM_PROMOTE_ANYSTAGE	This privilege allows you to promote items to the any promotion stage in the lifecycle
Promote to Next Stage	ITEM_PROMOTE_NEXTSTAGE	This privilege allows you to promote items to the next promotion stage in the lifecycle
Relate Item to Design Part	ITEM_RELATE_PART	This privilege allows you to relate/unrelate items to/from a design part.
Relate Item to Item	ITEM_RELATE_ITEM	This privilege allows you to relate/unrelate items to/from items.
Rename	ITEM_RENAME	This privilege allows you to rename item identifiers.
Revise Item Content	ITEM_UPDATECONTENT	This privilege allows you to check out, check in and merge items.
Rollback from Areas	ITEM_ROLLBACK	This privilege allows you to rollback items from areas. See "Note About Roll Back Privileges" on page 364.
Suspend	ITEM_SUSPEND	This privilege allows you to suspend items.
Update	ITEM_UPDATE	This privilege allows you to edit the attributes of an item.

Design Part Level Privileges

Name	ID	Description
Create	PART_CREATE	This privilege allows you to create design parts.
Delete	PART_DELETE	This privilege allows you to delete design parts.
Relate Design Part to Design Part	PART_RELATE_PART	This privilege allows you to relate/unrelate a design part to/from another design part.
Rename	PART_RENAME	This privilege allows you to rename design parts.
Suspend	PART_SUSPEND	This privilege allows you to suspend design parts.
Update	PART_UPDATE	This privilege allows you to update the attributes of the design part and create a new design part revision (or pcs).

Product Level Privileges

Name	ID	Description
Assign Roles to Users And Groups	PRODUCT_ROLE_ASSIGN	This privilege allows you to assign roles to users and groups.
Delete	PRODUCT_DELETE	This privilege allows you to delete a product.
Manage Libraries	PRODUCT_LIBRARYMAN	This privilege allows you to define product libraries.
Manage Object Types	PRODUCT_OBJTYPEMAN	This privilege allows you to manage the object type definitions and their attributes in your product.
Manage Project/Stream Upload Inclusions/ Exclusions	PRODUCT_PROJECTUPLOADMAN	This privilege allows you to manage project/stream-specific upload inclusions/exclusions.
Manage Validsets	PRODUCT_VALIDSETMAN	This privilege allows you to manage valid-set definitions and values.
Override Process Checks	PRODUCT_OVERRIDE_PROCESS_ CHECK	This privilege allows you to override certain levels of process checks.
Perform Mover Deployments	PRODUCT_RUN_MOVER_DEPLOY	This privilege allows you to perform deployments via either Mover or Dimensions.
Perform Requirement Related Operation	PRODUCT_REQUIREMENTMAN	This privilege allows you to perform requirement related operations.
Refresh Inboxes for All Users	PRODUCT_RUN_REFRESH_INBOX ES	This privilege allows you to refresh the user in-box of objects belonging to the product via the PEND command.
Rename	PRODUCT_RENAME	This privilege allows you to rename a product.
Run Reports	PRODUCT_RUN_REPORT	This privilege allows you to run reports.
Update	PRODUCT_UPDATE	This privilege allows you to update the attributes of the product.
View Other Users' Privileges	PRODUCT_VIEW_USERS_PRIVS	This privilege allows you to view other users' privileges.

Project/Stream Level Privileges

Name	ID	Description
Action to Any State	PROJECT_ACTION_ANYSTATE	This privilege allows you to action a project/stream to any state in the lifecycle.
Action to Next State	PROJECT_ACTION_NEXTSTATE	This privilege allows you to action a project/stream to the next state in the lifecycle. **Note:** When the rule is *Object is in the user's inbox or the user has current role* (OBJ_PEND) the "or" in this rule becomes "and".
Add Item revisions to Project	PROJECT_ADDFILE	This privilege allows you to add item revisions to a project.
Assign Deployment Areas to Project/Stream	PROJECT_ASSIGN_DEPL_AREA	This privilege allows you to relate/unrelate deployment areas to/from a project or stream.
Attach Baseline as Sub Project	PROJECT_ATTACH_BASELINE	This privilege allows you to attach/detach a sub project baseline to/from a project.
Attach Project as Sub Project	PROJECT_ATTACH_SUBPROJECT	This privilege allows you to attach/detach a sub project to/from a project.
Audit a Project/Stream	PROJECT_AUDIT	This privilege allows you to audit a project or stream via the AUDIT command.
Build from a Project/Stream	PROJECT_BUILD	This privilege allows you to build from a project or stream.
Bypass Locked Project/Stream	PROJECT_BYPASSLOCKED	This privilege allows you to bypass locked project/stream constraints.
Change CM Rules for Project/Stream	PROJECT_EDIT_CMRULES	This privilege allows you to change whether or not CM Rules are active for a project or stream.
Create	PROJECT_CREATE	This privilege allows you to create projects.
Create Directories	PROJECT_CREATE_DIR	This privilege allows you to create project/stream directories.
Create Stream	PROJECT_STREAM_CREATE	This privilege allows you to create streams.
Delete	PROJECT_DELETE	This privilege allows you to delete a project.
Delete Directories	PROJECT_DELETE_DIR	This privilege allows you to delete project/stream directories.
Delete Stream	PROJECT_STREAM_DELETE	This privilege allows you to delete a stream.

Name	ID	Description
Update Files from Project/Stream	PROJECT_DOWNLOAD	This privilege allows you to update files in your work area from a project/ stream. **Note**: If this privilege is removed, the Download command enforces the privileges specified for Fetch Item.
Import Request into Project	PROJECT_IMPORT_REQUEST	This privilege allows you to import a request into a project.
Lock	PROJECT_LOCK	This privilege allows you to lock and unlock a project or stream.
Populate an Area from a Project/ Stream	PROJECT_POPULATE_AREAS	This privilege allows you to populate an area from a project or stream.
Relate Requests to Project/Stream	PROJECT_RELATE_REQUEST	This privilege allows you to relate/ unrelate requests to/from a project or stream.
Remove Item revisions from Project	PROJECT_REMOVEFILE	This privilege allows you to remove item revisions from a project.
Rename	PROJECT_RENAME	This privilege allows you to rename a project or stream.
Rename Directories	PROJECT_RENAME_DIR	This privilege allows you to rename or move directories.
Rename Item Filenames	PROJECT_RENAME_FILE	This privilege allows you to rename project filenames.
Update	PROJECT_UPDATE	This privilege allows you to update the attributes of a project or stream.
Deliver Files into Project/Stream	PROJECT_UPLOAD	This privilege allows you to deliver files from your work area into a project or stream. **Note**: If this privilege is removed, upload enforces the privileges for the commands used to effect the changes. These can include CI, UI, RI, AIWS, SWF, RIWS, DI, CWSD, MWSD and DWSD. Which commands get used depends on the upload options and the content of the area compared to the project at the time of upload.
Control Personal Stream		After delegating a personal stream that you originated you can continue contributing to it. ADMIN group has this privilege granted by default.

Release Level Privileges

Name	ID	Description
Create	RELEASE_CREATE	This privilege allows you to create a release.
Delete	RELEASE_DELETE	This privilege allows you to delete a release.
Forward to Customer	RELEASE_FORWARD_CUSTOMER	This privilege allows you to forward or withdraw releases from customers.

Request Level Privileges

Name	ID	Description
Action to Any State	REQUEST_ACTION_ANYSTATE	This privilege allows you to action a request to any state in the lifecycle.
Action to Next State	REQUEST_ACTION_NEXTSTATE	This privilege allows you to action requests to the next state in the lifecycle. **Note:** When the rule is *Object is in the user's inbox or the user has current role* (OBJ_PEND) the "or" in this rule becomes "and".
Add Action Description	REQUEST_ADD_ACTION_DESC	This privilege allows you to add the action description of a request.
Add/Edit Detailed Description	REQUEST_DETAILDESC	This privilege allows you to add/edit the detailed description of a request.
Browse	REQUEST_BROWSE	This privilege allows you to browse requests.
Create	REQUEST_CREATE	This privilege allows you to create requests. **Note:** If a user has the necessary privilege to create a request, but they do not hold the necessary role on the initial lifecycle state, Dimensions will temporarily grant them that role, thus enabling the create operation to complete successfully. In such a case, a warning message will be displayed, and when the user actions the request to the next state the temporary role will be deleted.
Delegate	REQUEST_DELEGATE	This privilege allows you to delegate requests to users and groups.
Delete	REQUEST_DELETE	This privilege allows you to delete requests. **Note**: This only applies to draft requests.
Demote to Any Stage	REQUEST_DEMOTE_ANYSTAGE	This privilege allows you to demote requests to the any demotion stage in the lifecycle
Demote to Next Stage	REQUEST_DEMOTE_NEXTSTAGE	This privilege allows you to demote requests to the next demotion stage in the lifecycle
Deploy to Areas	REQUEST_DEPLOY	This privilege allows you to deploy requests to areas
Edit Action Description	REQUEST_EDIT_ACTION_DESC	This privilege allows you to edit the action description of a request.
Move	REQUEST_MOVE	This privilege allows you to move requests between the primary and secondary catalog.

Name	ID	Description
Perform Replication Operations	REQUEST_RREPLIC_OPS	This privilege allows you to perform replication specific operations for requests.
Prime	REQUEST_PRIME	This privilege allows you to prime requests.
Promote to Any Stage	REQUEST_PROMOTE_ANYSTAGE	This privilege allows you to promote requests to the any promotion stage in the lifecycle
Promote to Next Stage	REQUEST_PROMOTE_NEXTSTAGE	This privilege allows you to promote requests to the next promotion stage in the lifecycle
Relate Request to Baseline	REQUEST_RELATE_BASELINE	This privilege allows you to relate/unrelate a request to/from baselines.
Relate Request to Design Part	REQUEST_RELATE_PART	This privilege allows you to relate/unrelate a request to/from design parts.
Relate Request to Item	REQUEST_RELATE_ITEM	This privilege allows you to relate/unrelate a request to/from items.
Relate Request to Request	REQUEST_RELATE_REQUEST	This privilege allows you to relate/unrelate a request to/from other requests.
Rollback from Areas	REQUEST_ROLLBACK	**Note**: This privilege does not have any effect, and should not be used.
Update Attachments	REQUEST_UPDATE_ATTACH	This privilege allows you to add/remove attachments.
Update Request	REQUEST_UPDATE	This privilege allows you to edit attributes and add/remove action descriptions.

Index

Symbols

A

B

C

D

E

F

G

H

I

L

M

N

O

P

R

S

T

U

V

W

Z