

**opentext™**

# Dimensions CM

Software version: 14.7

## Reports Guide



Copyright © 2023 Open Text.

The only warranties for products and services of Open Text and its affiliates and licensors ("Open Text") are as may be set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Open Text shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.

Product version: 14.7

Last updated: December 8, 2023

# Table of Contents

---

<i>Chapter 1</i>	<b>Introduction . . . . .</b>	<b>9</b>
	Introduction to OpenText™ Dimensions CM Reports. . . . .	10
	Users of Dimensions CM Development Reports . . . . .	10
<i>Chapter 2</i>	<b>Dimensions Standard Command-Line and ASCII Reports . .</b>	<b>11</b>
	Introduction . . . . .	12
	Product Structure Reports. . . . .	12
	Report on Parts. . . . .	13
	Report on Baseline . . . . .	16
	Report on Export File. . . . .	18
	Change Management Reports . . . . .	18
	Selection Criteria. . . . .	18
	Report Types . . . . .	20
	Brief Descriptions of Report Types. . . . .	21
	Sample Catalog Report . . . . .	22
<i>Chapter 3</i>	<b>User Reports Administration. . . . .</b>	<b>25</b>
	About User Reports Administration. . . . .	26
	About User Report Definitions. . . . .	26
	About User Report Files . . . . .	26
	About the Product-Range Parameter . . . . .	27
	About Roles for Running Reports. . . . .	27
	Defining User Reports and Managing Report Files . . . . .	27
	Running User Reports. . . . .	28
	Process of Running a Report. . . . .	28
<i>Chapter 4</i>	<b>The Published Views . . . . .</b>	<b>29</b>
	Introduction . . . . .	30
	Installing Published Views. . . . .	30
	Notes About Published Views. . . . .	31
	The Published Views. . . . .	32
	pcms_archives - details on archives that have been created. . . . .	32
	pcms_archive_items - items that are contained in an archive. . . . .	33
	pcms_area_catalogue - area catalog . . . . .	34
	pcms_attr_map - attribute mapping . . . . .	35
	pcms_attributes_baseline - baselines and their attributes . . . . .	36
	pcms_attributes_chdoc - requests and their attributes. . . . .	37
	pcms_attributes_item - items and their attributes . . . . .	38
	pcms_attributes_part - parts and their attributes . . . . .	39
	pcms_attributes_product - products and their attributes . . . . .	40
	pcms_attributes_user - user-defined attributes for user. . . . .	41
	pcms_attributes_workset - projects and their attributes. . . . .	42

pcms_audit_log. . . . .	43
pcms_baseline_attributes - baseline attribute information . . . . .	44
pcms_baseline_history - baseline update history. . . . .	45
pcms_baseline_info - baseline information . . . . .	46
pcms_baseline_itemdirs - items and their file details contained in a baseline . . . . .	48
pcms_baseline_items - items in baselines . . . . .	49
pcms_baseline_norm_lc - normal lifecycle information for baselines .	50
pcms_baseline_parentage - pedigree of revised and merged baselines	51
pcms_baseline_parts - parts in baselines . . . . .	52
pcms_baseline_related_chdocs - baseline and related requests . . . .	53
pcms_baseline_revision_mods - baseline changes made by CRB. . . .	54
pcms_baseline_template - baseline templates . . . . .	55
pcms_baseline_types - baseline types . . . . .	56
pcms_bl_rtm_project_xref - baseline to RM project associations. . . .	57
pcms_bld_bline_cfgver - Dimensions Build baseline configuration version . . . . .	58
pcms_bld_build_config - Dimensions Build configuration information . . . . .	59
pcms_bld_build_config_version - Dimensions Build configuration version . . . . .	60
pcms_bld_build_env - Dimensions Build environment . . . . .	61
pcms_bld_build_job - Dimensions Build job . . . . .	62
pcms_bld_build_job_for_target - Dimensions Build jobs for targets. .	63
pcms_bld_build_job_target_hist - Dimensions Build jobs and targets history . . . . .	64
pcms_bld_platform - Dimensions Build platforms . . . . .	65
pcms_bld_target - Dimensions Build targets. . . . .	66
pcms_bld_type - Dimensions Build types . . . . .	67
pcms_bln_history_with_attr - baseline update history including attributes. . . . .	68
pcms_bln_rels2 - baseline relationships information . . . . .	69
pcms_branches - registered branch names. . . . .	70
pcms_chdoc_action_desc - request action description . . . . .	71
pcms_chdoc_attr_history - historic request attribute information . . .	72
pcms_chdoc_attachments - request attachments, primary catalog . .	73
pcms_chdoc_data - current request information . . . . .	74
pcms_chdoc_delegation - request delegations . . . . .	76
pcms_chdoc_detail_desc - request detailed description . . . . .	77
pcms_chdoc_history - request history . . . . .	78
pcms_chdoc_history_with_attr - request history with attributes . . . .	79
pcms_chdoc_norm_lc - normal lifecycle for requests . . . . .	80
pcms_chdoc_related_baselines - request and related baselines . . . .	81
pcms_chdoc_related_chdoc - related request information . . . . .	82
pcms_chdoc_related_items - requests and related items . . . . .	83
pcms_chdoc_related_parts - requests and related design parts . . . .	84
pcms_chdoc_related_worksets - request project relationships . . . . .	85
pcms_chdoc_this_action_desc - request current action description . .	86
pcms_chdoc_types - request type information . . . . .	87

pcms_chdoc_user_lists - request user lists . . . . .	88
pcms_cm_catalogue - basic request information . . . . .	89
pcms_command_statistics - command statistics . . . . .	91
pcms_customer_details - customer information . . . . .	92
pcms_customer_releases - releases forwarded to customers . . . . .	93
pcms_deployment_data - deployment data . . . . .	94
pcms_deployment_history - deployment history . . . . .	95
pcms_derived_formats - one-to-one derived build relationship . . . . .	96
pcms_environment_items - environment item definitions. . . . .	97
pcms_groups - groups definitions . . . . .	98
pcms_group_members - group members . . . . .	99
pcms_ipds - Item Process Definitions registered against a product . .	100
pcms_item_attributes - product item attribute information. . . . .	101
pcms_item_attr_history - historic product item attribute information.	102
pcms_item_data - current item information . . . . .	103
pcms_item_delegation - item delegations . . . . .	105
pcms_item_exports - items checked out by users . . . . .	106
pcms_item_history - history of items. . . . .	107
pcms_item_history_with_attr - item update history including attributes . . . . .	108
pcms_item_norm_lc - normal lifecycle for product items . . . . .	109
pcms_item_pending_actions - users and their items to be actioned. .	110
pcms_item_related_items - user-defined relations between items. . .	111
pcms_item_related_parts - items and their related design parts . . . .	112
pcms_item_revision_history - history of item revisions . . . . .	113
pcms_item_spec_data - current item specification . . . . .	114
pcms_item_types - product item type information. . . . .	115
pcms_life_cycles - lifecycles . . . . .	117
pcms_mail_notif - e-mail notifications . . . . .	118
pcms_mail_subscribers - e-mail notification subscribers. . . . .	119
pcms_make_instance_catalogue - network nodes used in the Dimensions Make process . . . . .	120
pcms_make_mo_list - made-of list for items resulting from Dimensions Make. . . . .	121
pcms_mo_list - made-of list . . . . .	122
pcms_part_attributes - design part attribute information . . . . .	123
pcms_part_attr_history - historic design part attribute information . .	124
pcms_part_data - design part information . . . . .	125
pcms_part_history - design part update history . . . . .	126
pcms_part_history_with_attr - design part update history including attributes. . . . .	127
pcms_part_types - design part type information . . . . .	128
pcms_parts_structure - structural relationships between design parts	129
pcms_privileges - privileges . . . . .	130
pcms_privilege_rules_enabled - privilege rules enabled . . . . .	131
pcms_product_attributes - product attribute information . . . . .	132
pcms_product_attr_upd_history - product attribute update history . .	133
pcms_product_catalogue - product catalog. . . . .	134

pcms_project_areas - list projects associated to a deployment area .	135
pcms_promote_history - promotion history . . . . .	136
pcms_release_data - release information . . . . .	137
pcms_release_items - items in releases . . . . .	138
pcms_release_template - release templates . . . . .	139
pcms_replicator_branches - branches replicated . . . . .	140
pcms_replicator_chdoc_types - replicator request types . . . . .	141
pcms_replicator_configs - registered replication configurations . . . . .	142
pcms_replicator_config_rels - replica definitions in a primary definition . . . . .	144
pcms_replicator_receive - inward transfers received . . . . .	145
pcms_replicator_receive_cdocs - requests received . . . . .	146
pcms_replicator_receive_items - item revisions received . . . . .	147
pcms_replicator_receive_logs - text logs received . . . . .	148
pcms_replicator_transfer - outward transfers . . . . .	149
pcms_replicator_transfer_cdocs - requests for transfer . . . . .	150
pcms_replicator_transfer_items - items for transfer . . . . .	151
pcms_replicator_transfer_logs - text logs associated with outward transfers . . . . .	152
pcms_report_chdoc_data - current request information used for reporting purposes . . . . .	153
pcms_role_sections - role sections . . . . .	155
pcms_rtm_dm_project_col_xref - RM collections associations . . . . .	156
pcms_rtm_object - requirements associations . . . . .	157
pcms_rtm_project_product_xref - Dimensions RM project association	158
pcms_rules- rule details . . . . .	159
pcms_rule_privileges - rules applicable for each privilege . . . . .	160
pcms_sec_chdoc_action_desc - secondary request action description	161
pcms_sec_chdoc_attachments - request attachments, secondary catalog . . . . .	162
pcms_sec_chdoc_attr_history - historic secondary request attribute information . . . . .	163
pcms_sec_chdoc_data - secondary request information . . . . .	164
pcms_sec_chdoc_detail_desc - secondary request detailed description	165
pcms_sec_chdoc_history - secondary request history . . . . .	166
pcms_sec_history_with_attr - secondary request history with attributes . . . . .	167
pcms_sec_status_rpt1 - secondary requests on normal lifecycle . . . . .	168
pcms_sec_this_action_desc - secondary request current action description . . . . .	169
pcms_sites - registered base database sites . . . . .	170
pcms_status_rpt1 - requests on normal lifecycle . . . . .	171
pcms_strings_db - Dimensions Make commands used to make an item	172
pcms_user_attributes - user attribute information . . . . .	173
pcms_user_attributes_def - user-defined attributes . . . . .	174
pcms_user_attr_update_history - user attribute update history . . . . .	176
pcms_user_audit_trail - user audit trail . . . . .	177
pcms_user_pending_chdoc - user and pending request information .	178
pcms_user_roles - roles allocated to users . . . . .	179

---

pcms_users_profile - registered users . . . . .	180
pcms_valid_set_view - valid sets . . . . .	181
pcms_workset_attributes - project attributes . . . . .	182
pcms_workset_branches - project version branch names . . . . .	183
pcms_workset_components - project components . . . . .	184
pcms_workset_history - project history . . . . .	185
pcms_workset_history_with_attr - project update history including attributes . . . . .	186
pcms_workset_info - project information . . . . .	187
pcms_workset_items - items in projects . . . . .	188
pcms_workset_norm_lc - normal lifecycle for projects . . . . .	189
pcms_workset_related_chdocs - project request relationships . . . . .	190
pcms_workset_structure_history - project refactoring changes . . . . .	191
pcms_full_wset_struct_history - project refactoring changes . . . . .	192
Use of the Views . . . . .	193
List of Open Requests . . . . .	193
Request Selected on Multi-Valued Attributes . . . . .	193
Requests Pending for a User and Related Parts . . . . .	194
Nested Queries . . . . .	194
Hierarchical Queries . . . . .	195





# Chapter 1

---

## **Introduction**

Introduction to OpenText™ Dimensions CM Reports

10

---

# Introduction to OpenText™ Dimensions CM Reports

Development reporting is a vital element of the development process. To provide support for such reporting, Dimensions CM offers:

- **Dimensions CM Standard Reports** (product structure and change management reports) to facilitate decision-making by transforming Dimensions CM metadata into clear, concise, and meaningful information in the form of easy-to-use standard reports. Dimensions CM Standard Reports take the form of ASCII format reports that may be run from the `dmccli` command. Refer to "[Dimensions Standard Command-Line and ASCII Reports](#)" on page 11 for details.
- **Database Published Views** reporting facilities for Dimensions CM objects such as design part, baseline, request, item, project, release, and product. Dimensions CM provides a number of database Published Views that enable you to write your own programs to extract information from the database. Refer to "[The Published Views](#)" on page 29 for a description of each view and how to install them.
- **User reports** that can be set up in the Administration Console. See "[User Reports Administration](#)" on page 25 for details. To run these user reports, you use the Dimensions CM RUR command. For details, see the *Command-Line Reference*.

Limitations Dimensions CM reports do not support external requests.

To provide support for general end-user reporting, Dimensions CM offers:

- The web client and desktop client's Report Builder tool. Use this tool to create, edit, and run end-user GUI reports (the tool replaces and extends the Query Builder tool provided with Dimensions CM prior to release 10.1.3). For details, see the *online help*.
- Through the desktop client, access to the Crystal Reports desktop, provided you have this product installed. See the *Dimensions CM online help* for information on how to access Crystal Reports.

## Users of Dimensions CM Development Reports

The different types of users are largely determined in the creation of the process model (formerly known as control plan). However, it can be expected that users of Dimensions CM reports will fall into categories similar to the following.

- **Developers**  
Users who have responsibility for the initial creation of items (program source, documents, test data, etc).
- **Team/Project Leaders**  
Users who have responsibility for part of the product structure. The items belonging to those parts are created by Developers who are part of their team.
- **Integrators**  
Users who integrate the items created by developers.
- **Managers**  
Users who have responsibility for all or part of the product and additionally have responsibility for one or more teams of developers.

## Chapter 2

---

# Dimensions Standard Command-Line and ASCII Reports

Introduction	12
Product Structure Reports	12
Change Management Reports	18

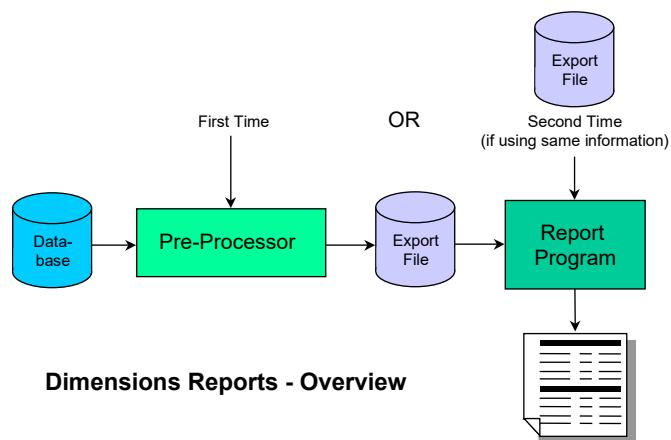
---

## Introduction

This chapter lists and describes the functions and types of reports that can be undertaken when generating command-line ASCII format reports. For full details of the Dimensions CM command-line reports and options, see the *Command-Line Reference*.

## Product Structure Reports

The product structure report programs operate with a pre-processor, as displayed in the following diagram.



The pre-processor extracts complete structure details on the required product (or specified part of it) from the database and stores them in an **export-file** within the user-area.



**IMPORTANT!** Despite the name used for it, this is not the same as an "export of the database." These 'export-files' are *not* suitable as backups of database contents, as they can be used *only* for producing reports of this snapshot of the database, and/or for comparing two snapshots from the same or different databases.

The pre-processor phase can be bypassed where an existing file already holds the required product-structure details. This provision can save a considerable amount of time where a number of different reports of the same product-structure are required.

An export file can also be retained for future use, for example, to print a report of differences between it and the product-structure as developed subsequently.

In joint-development environments, an export-file can be used in conjunction with the transfer of a product release to another database or installation. In this case, the file can be compared with the recipient's existing database to highlight any differences in the new release.

The report-option programs provide a number of variations in product-structure reporting. A report may be based on:

- Current product-structure.
- Product-structure secured by a baseline.
- Previously saved export-files, based on either a baseline or the then-current product-structure.

The differences between any two of the above structure sources can also be reported.

Product-structure reports are provided as follows.

<b>Report</b>	<b>Command</b>
■ Report Current Items	RCI
■ Report Current Parts	RCP
■ Report Design Structure	RDS
■ Compare Structures or Baselines	CMP

**Limitations** Users initiating the following reports must have a valid role for the top design part being reported. In a structure-report whose structure is to include items, when an item has two or more revisions currently at the same lifecycle state, only the latest (most recently created/updated) of these is shown.

## Report on Parts

Reports on Parts are implemented through RCI, RCP, and RDS from the command line.

These provide the following reporting functions.

- **Report on Items** The **RCI** command: This produces a report for all items found in the product-structure. If no export file is specified, the file name defaults to `item_list.out`. You may order by item identifier only.
- **Report on Parts** The **RCP** command: This produces a report for all design parts found in the product structure. You may choose to sort by part identifier or part number.
- **Report on Product Structure** The **RDS** command: This produces a structure-level report for a product-structure down to a certain number of levels (or to all levels), optionally including items, requests, user roles, and usage relationships.

The reports generated by choosing an example part `FS:CABIN.A;1`, with all report options being at their default values, are shown on the following pages:

REPORT CURRENT ITEMS

Parameters: RCI tmp\_file\_xrep\_1b73.tmp /NEW /SORT=IID  
/PART="FS:CABIN.A;1"

User : USER1

Date : 17/Mar/23 17:20

17/Mar/23 17:20    PRODUCT: FS    CURRENT ITEMS REPORT    TOP PART: CABIN    Page 1

<u>Item id</u>	<u>Part no.</u>	<u>Var</u>	<u>Item Type</u>	<u>Revision</u>	<u>Status</u>	<u>Filename</u>	<u>Ver</u>
CABIN		A	OBJ	1	COMPILED	cabin.o	1
CABIN		A	SRC	2	APPROVED	cabin.c	2
CABIN AIR		A	OBJ	1	COMPILED	cab_air.o	1
CABIN AIR		A	SRC	2	APPROVED	cab_air.c	2
CABIN DOOR		A	OBJ	1	COMPILED	cab_door.o	1
CABIN DOOR		A	SRC	2	APPROVED	cab_door.c	2
CABIN LIGHTS		A	OBJ	1	COMPILED	cablight.o	1
CABIN LIGHTS		A	SRC	2	APPROVED	cablight.c	2

REPORT CURRENT PARTS

Parameters: RCP tmp\_file\_xrep\_1b73.tmp /OLD /SORT=PID  
/PART="FS:CABIN.A;1"

User : USER1

Date : 17/Mar/23 17:21

17/Mar/23 17:21    PRODUCT: FS    CURRENT ITEMS REPORT    TOP PART: CABIN    Page 1

<u>Part id</u>	<u>Var</u>	<u>PCS</u>	<u>Part no.</u>	<u>Local Part no.</u>	<u>Category</u>	<u>Description</u>
CABIN	A	1			SUB-SYSTEM	Cabin Envrnmnt Sub-system
CABIN AIR	A	1			MODULE	Cabin air conditioning systems
CABIN DOOR	A	1			MODULE	Cabin door
CABIN AIR	A	1			MODULE	Cabin lighting systems

REPORT DESIGN STRUCTURE

Parameters: RDS \*/STRUCTURE=ALL  
/PART="FS:CABIN.A;1"

User : USER1

Date : 17/Mar/23 17:21

CABIN.A;1 (SUB-SYSTEM)

Roles -

USER1	(ANALYST)
USER1	(AUTHOR)
USER1	(CHANGE REVIEWER)
ROBERT	(ENGINEER)
USER1	(ENGINEER)
USER1	(LEAD)
USER1	(QA)
USER1	(REVIEWER)

Items -

CABIN	.A	-OBJ	;1	COMPILED	OWNER
CABIN	.A	-SRC	;2	APPROVED	OWNER

1 CABIN AIR.A;1 (MODULE)

Items

CABIN AIR	A	-OBJ	;1	COMPILED	OWNER
CABIN AIR	A	-SRC	;2	APPROVED	OWNER

1 CABIN DOOR.A;1 (MODULE)

Items

CABIN DOOR	A	-OBJ	;1	COMPILED	OWNER
CABIN DOOR	A	-SRC	;2	APPROVED	OWNER

1 CABIN LIGHTS.A;1 (MODULE)

Items

CABIN LIGHTS	A	-OBJ	;1	COMPILED	OWNER
CABIN LIGHTS	A	-SRC	;2	APPROVED	OWNER

## Report on Baseline

Reports on Baselines are implemented through RCI, RCP, and RDS from the command-line.

- **Report on Items** The **RCI** command: This produces a report for all baselined items found in the product-structure. If no export file is specified, the file name defaults to `item_list.out`. You may order by item identifier only.
- **Report on Parts** The **RCP** command: This produces a report for all baselined design parts found in the product structure. You may order by part identifier (default) or part number.
- **Report on Product Structure** The **RDS** command: This produces a structure-level report for the baselined product-structure down to a certain number of levels (or to all levels), optionally including items, requests, user roles and usage relationships.

As an example, the reports generated by choosing the baseline `FS:CABIN_BL1` are shown below:



REPORT CURRENT ITEMS  
Parameters: RCI tmp\_file\_xrep\_33c3.tmp /NEW /SORT=IID  
/BASELINE="FS:CABIN.A;1"  
User USER1  
Date 17/Mar/23 18:37

17/Mar/23 18:37 PRODUCT: FS CURRENT ITEMS REPORT BASELINE: CABIN\_BL1 Page 1

<u>Item id</u>	<u>Part no.</u>	<u>Var</u>	<u>Item Type</u>	<u>Revision</u>	<u>Status</u>	<u>Filename</u>	<u>Ver</u>
CABIN		A	OBJ	1	COMPILED	cabin.o	1
CABIN		A	SRC	2	APPROVED	cabin.c	2
CABIN AIR		A	OBJ	1	COMPILED	cab_air.o	1
CABIN AIR		A	SRC	2	APPROVED	cab_air.c	2
CABIN DOOR		A	OBJ	1	COMPILED	cab_door.o	1
CABIN DOOR		A	SRC	2	APPROVED	cab_door.c	2
CABIN LIGHTS		A	OBJ	1	COMPILED	cablight.o	1
CABIN LIGHTS		A	SRC	2	APPROVED	cablight.c	2

REPORT CURRENT PARTS  
Parameters: RCP tmp\_file\_xrep\_33c3.tmp /OLD /SORT=PID  
/BASELINE="FS:CABIN.A;1"  
User USER1  
Date 17/Mar/23 18:37

<u>Part id</u>	<u>Var</u>	<u>PCS</u>	<u>Part no.</u>	<u>Local Part no.</u>	<u>Category</u>	<u>Description</u>
CABIN	A	1			SUB-SYSTEM	Cabin Envrnmnt Sub-system
CABIN AIR	A	1			MODULE	Cabin air conditioning systems
CABIN DOOR	A	1			MODULE	Cabin door
CABIN AIR	A	1			MODULE	Cabin lighting systems

## Report on Export File

Reports on Export files are implemented through RCI, RCP, and RDS from the command line:

- **Report on Items** The **RCI** command: This produces a report for all items found in the product-structure as stored in the export file. If no export file is specified, the file name defaults to `item_list.out`. You may order by item identifier only.
- **Report on Parts** The **RCP** command: This produces a report for all design parts found in the product structure as stored in the export file. You may order by part identifier (default) or part number.
- **Report on Product Structure** The **RDS** command: This produces a structure-level report for a product-structure as stored in the export file down to a certain number of levels (or to all levels), optionally including items, request, user roles and usage relationships.

## Change Management Reports

Change Management (Request) reports designed to suit different user requirements can be generated through the command-line RPT command.

Each report is based on a specified product and request category, and these define the set of requests to be covered by the report. If a sharper focus is desired, users may specify further parameters to refine the selected set of requests.

By default, reports present information about requests, but do not show any of their text. Optionally, a detailed report may be requested, and this will include the full text of each selected request (but users should beware—this could be very bulky).

## Selection Criteria

All request reports cover a selection from the full set of requests recorded in the system. Some of the criteria used for this selection are mandatory for all reports, but there are other optional criteria that are useful in special cases.

The first criterion is the report type, which may be:

- /CATALOGUE
- /CATALOGUE/SECONDARY
- /PENDING

Most reports are provided for all types, but some (notably USER\_CHDOC) are only available in one type. The report type is combined with other selection criteria to affect the choice of requests for a report, and this is discussed below.

The other mandatory criteria are the product-id for the product owning the requests to be reported, and the request category. The request type should also be specified.

There follows a set of optional criteria, which must be matched exactly if specified (i.e. no wild cards), or which default to "any" if not specified.

<b>Part_Id</b>	<b>Comment</b>
<b>Phase</b>	This specifies the phase which all requests to be selected are currently in. So a report could cover just documents in the WORK phase.
<b>Dimensions User</b>	This may specify a user's login identity, and this is combined with the report type in selecting requests. See below.
<b>Period From</b>	This may specify a starting date, and this is combined with the report type in selecting requests. See below.
<b>Period To</b>	This may specify a finishing date, and this is combined with the report type in selecting requests. See below.

---

The report type is combined with these last three criteria in the selection of requests for the report.

For a "main/secondary catalog" report, requests will be selected (subject to the other criteria) if they have been actioned or created by the specified user within the given period. The defaults are, of course, "any user" or "any date". In the case where neither a user nor a period is given, a "catalog" report will be able to cover all requests including those that have reached the end of their lifecycles.

For a "pending" report, requests will be selected (subject to the other criteria) if they have been passed for actioning to the specified user within the given period, and are still awaiting action by that user. The defaults are, of course, "any user" or "any date". In the case where neither a user nor a period is given, a "pending" report will be able to cover all requests that are still to be actioned by some user—that is any request apart from those that have reached the end of their lifecycles.

As well as all the above, the request selection may be further refined by specifying (if required) the request identity, current lifecycle status, and the values of certain user-defined attributes including `attribute-1` (conventionally the request title).

## Report Types

The table shown below lists the available report types. The left column gives the report type and the right column gives the title. Reports must be selected from either the Catalog or projects as defined by the middle column of this table.

Report Type	Catalog or Pending	Title
CH_DOC_LIST	C	The ch_doc catalog
CH_DOC_LIST	P	The ch_docs pending
CHDOC_PART_ITEM	C	Ch_docs and all related design parts and items
CHDOC_PART_ITEM	P	Ch_docs and all related design parts and items
DEPENDENCY_REP	C	Ch_docs and their dependent requests
DEPENDENCY_REP	P	Ch_docs and their dependent requests
USER_CHDOC	P	Ch_docs outstanding for users
FULL_HISTORY	C	Ch_doc action or update history report
FULL_HISTORY	P	Ch_doc action or update history report
ACTION_HISTORY	C	Request History Report (Action History)
ACTION_HISTORY	P	Request History Report (Action History)
PART_CHDOC_ITEM_R	C	Recursive: design parts, related ch_docs, items
PART_CHDOC_ITEM_R	P	Recursive: design parts, related ch_docs, items
PART_CHDOC_ITEM_S	C	Single: design parts, related ch_docs, items
PART_CHDOC_ITEM_S	P	Single: design parts, related ch_docs, items
PARM_11_12	C	Ch_doc catalog ordered by report parameters 11 and 12
PARM_11_12	P	Ch_docs pending ordered by report parameters 11 and 12
PARM_12_11	C	Ch_doc catalog ordered by report parameters 12 and 11
PARM_12_11	P	Ch_docs pending ordered by report parameters 12 and 11
BASELINE_DETAIL	n/a	Baseline detail report

## Brief Descriptions of Report Types

Type	Description
CH_DOC_LIST	This report lists all requests which match the selection criteria.
CHDOC_PART_ITEM	This report lists each request which matches the selection criteria, and for each of these, it then lists all of that request's related design parts and items.
DEPENDENCY_REP	This report lists each request which matches the selection criteria and which has requests related to it, followed by the related requests.
USER_CHDOC	This report is only available as a "pending" report. It is arranged by user and shows the requests that have been passed to the user and that are still awaiting actioning. Each request is followed by a list of items related to that request. As with other reports, the requests listed are those that meet any given selection criteria. If, for example, two dates are specified, then the report will only list requests passed to the users between those dates.
FULL_HISTORY	This report lists all requests which match the selection criteria, and includes the action history and the update history for the listed requests.
ACTION_HISTORY	This report lists all requests which match the selection criteria, and includes the action history for the listed requests.
PART_CHDOC_ITEM_R	This report relates requests to the design part structure of a product, or of a portion of a product. Starting at the specified design part, or failing that at the top part of the product, the design parts in the structure are shown in family-tree order. Each design part is followed by a list of all related requests that match the selection criteria.
PART_CHDOC_ITEM_S	For each of the design parts which match the selection criteria, this report lists the requests related to these design parts and matching the relevant selection criteria; and for each request, it lists the related design parts and the related items. Within this report, the design parts appear in alphabetical order.
PARM_11_12	This report lists all requests which match the selection criteria, and includes user-defined attributes 2 and 3 which are identified as parameters 11 and 12. This report is sorted according to user-defined attribute 2 (parameter 11). Attributes 2 and 3 should be defined as CHARacter attributes and <b>must not exceed 25 characters in length for the purposes of this report.</b>



Micro Focus	The ch_doc catalogue			20-Mar-2023 14:03	
Ch_doc_id	Phase	Ch_doc Status	Date Created	Date Last Actioned	Severity /Priority
<b>Title</b>					
FS_PR_1	REJECTED	PROBLEM SOLVED	21-FEB-2023	21-FEB-2023	1_critical
	A Continuous Bleeping occurs when invoking print function				
FS_PR_2	AN+WORK	RAISED	24-FEB-2023	24-FEB-2023	3_mderate
	This is to set up training				
FS_PR_3	AN+WORK	RAISED	24-FEB-2023	24-FEB-2023	2_severe
	yet again				
FS_PR_4	AN+WORK	RAISED	24-FEB-2023	24-FEB-2023	2_severe
	yet again again				
FS_PR_5	AN+WORK	RAISED	14-MAR-2023	14-MAR-2023	2_severe
	user3 FS fails, core dump				
FS_PR_6	AN+WORK	RAISED	14-MAR-2023	14-MAR-2023	3_mderate
	Emilio's Change Document No.1				
FS_PR_7	AN+WORK	RAISED			4_minor
	This a test prob report from JP				
FS_PR_8	AN+WORK	RAISED			1_critical
	<user_id> FS fails, core dump				
FS_PR_9	AN+WORK	RAISED			4_minor
	junk				
FS_PR_10	AN+WORK	RAISED			2_severe
	test				

[End of Report]





## Chapter 3

---

# User Reports Administration

About User Reports Administration	26
Defining User Reports and Managing Report Files	27
Running User Reports	28

---

## About User Reports Administration

- Purpose User Reports Administration in the Administration Console enables you to:
- Create user report definitions that can be run using the RUR (Run User Report) function from the command-line interface, the desktop client, or the web client.
  - Create and manage user report files, and assign them to one or more existing report definitions.
  - View, edit, and delete existing user report definitions and user report files.
- PRIVILEGES** To set up or run user reports, you need the following privileges:
- To set up user reports: ADMIN\_REPORTMAN
  - To run user reports: PRODUCT\_RUN\_REPORT

Invocation Dimensions CM Administration Console | Database Management | User reports administration

For details about the User Reports Administration window, see the *Dimensions CM online help*.

### About User Report Definitions

A user report definition enables you to produce a custom report that can be run using the Dimensions CM RUR command. It consists of:

- Attributes that identify the operating system of the machine running Dimensions CM and the scope of the report.
- A user report file that contains the commands for executing the report.
- Up to seven optionally defined parameters that appear as prompts if the user runs the RUR command in interactive mode.

### About User Report Files

A user report file is any executable file that contains statements or commands that are executed as part of a user-defined report. For example, the user report file could be a Bourne-shell script for UNIX systems or a DOS batch file for Windows systems. For each report definition, you must designate one main (default) user report file to execute the report.

Auxiliary files The main user report file may also include names of files that supply auxiliary specifications. For example, an auxiliary file could contain SQL statements to be executed or operating system commands to be executed as a subroutine. You can include as many nested auxiliary files as you want, referring to them as simple file names in the current working directory, rather than including a unique location or pathname. You must add all auxiliary files to Dimensions CM as user report files, and assign them to the same report definition as the main user report.

When a user runs a report, Dimensions CM retrieves all the report files assigned to the report definition from the database and executes the main report file. Any auxiliary report files included in the main report file are executed as well.

## About the Product-Range Parameter

The **product-range** input parameter is a string of up to 10 characters, which may be as follows.

- Either a single specific product-id in the database; or
- A string which can represent one or more product-ids in the database, by using wildcard characters. These are the characters `_` (match single character) and/or `%` (match any characters):
  - each `_` character is considered to be matched by any one, and exactly one, character in a product-id
  - each `%` character is considered to be matched by any group of zero or more characters in a product-id.

For example, a string of `%` alone would match *all* product-ids in the database, or a string of `_____` would match all product-ids exactly five characters long.

## About Roles for Running Reports

When a user runs the RUR command, Dimensions CM checks the value that the user selects or types for the product or product range, and verifies that:

- The product range value matches at least one actual product ID in the base database.
- The product is not \$GENERIC (you cannot run reports against this product).

If any of these checks fails, Dimensions CM displays an error message and does not produce the report.

# Defining User Reports and Managing Report Files

In the User Reports Administration section of the Administration Console, you can:

- Create user report definitions.
- Edit the scope, description, and parameters of a user report definition.
- Add user reports files and assign them to user reports.
- Export and import user report files.

For details about these Administration Console functions, and an example of setting up and running a user report, see the *Dimensions CM online help*.

# Running User Reports

## Process of Running a Report

The process of running a user-defined report consists of these stages:

Stage	Description
1	The user runs the RUR command from the command-line interface, or uses the Run User Report wizard from the desktop client or web client. You can run the wizard when you select a request.
2	As part of the RUR command, the user specifies: <ul style="list-style-type: none"> <li>■ The report definition name</li> <li>■ The product ID or range of products</li> <li>■ Any parameter values for the report</li> <li>■ The name of the output file for the report</li> </ul>
3	Dimensions CM verifies the product ID and the user's role on the product(s).
4	Dimensions CM gets the report files specified for the report definition from the database.
5	Dimensions CM executes the main user report file with the statement: <ul style="list-style-type: none"> <li>■ UNIX /bin/sh &lt;command script file&gt; \ &lt;product-range&gt; &lt;other parameters&gt;</li> <li>■ Windows &lt;command script file&gt; - &lt;product-range&gt; &lt;other parameters&gt;</li> </ul>
6	Dimensions CM outputs the report to the specified file.

For help with using the RUR command, see the *Command-Line Reference*.

## Chapter 4

---

# The Published Views

Introduction	30
Installing Published Views	30
Notes About Published Views	31
The Published Views	32
Use of the Views	193

## Introduction

From time to time users may require access to the database, particularly when they need to supplement the standard reports with specially written reports. One method is to write programs in C, using the API routines described in the *Developer's API Guide* to extract information from the database.

It may be preferred to use a report writer, and several are available that enable users to interrogate a database and to format the results into useful reports. Such users need database views that can be maintained even when there are changes in the underlying database schema.

The views described in this document are of two types: those that need to be generated to incorporate definitions taken from the user's current schema and those that are supplied in fixed form.

*Schema objects other than the Views defined in this document should not be used for Report purposes, as OpenText reserves the right to modify the underlying schema at any time in the future.*

The majority of objects managed by Dimensions CM are associated with a Unique Identifier (UID) which is held in the corresponding database tables. The UID for any object is a number uniquely allocated to the object at its creation. The UID is used to join different tables referring to the object, and it has no other meaning. Many of the views return UIDs for relevant objects, so that these may be used in subsidiary queries.

## Installing Published Views

Published views are installed using the *dmdba* tool. For details on how to run this tool, see the *Administration Guide*.

The following commands are provided for managing report views:

- INSV - Install report views.
- DELV - Delete report views.
- GRTV - Grant access to views for user.
- REKV - Revoke access to views from user.
- LSGV - List users that have been granted access to the views.

For information about using these commands, see any of the following documents:

- *Dimensions CM System Administration*, in the *Administration Guide*.
- *Post-Installation Activities For Dimensions CM For Windows*, in the *Windows Installation Guide*.
- *Post-Installation Activities For Dimensions CM For UNIX*, in the *UNIX Installation Guide*.

## Notes About Published Views

Some of these views provide details on objects together with their attribute values. The full set of possible attributes is part of the generic process model in any base database, and may be inspected and modified through the Administration Console Scripting Interface. For details about the scripting interface, see the *online help*.

Because the attribute set is variable, you should delete and reinstall the views using the DELV and INSV commands described above, following any changes to the attribute set.

The number of rows returned by some views for any single object may depend on the object's current attributes. If it has single-valued attributes only, and no multi-valued attributes, one row will be returned with the field seq having the value 1. If the object has any multi-valued attributes, the number of rows will be at least as many as the highest number of values assigned to any one attribute. In all cases, the non-attribute fields are repeated in each row.

## The Published Views

### **pcms\_archives - details on archives that have been created**

This view lists the details on archives that have been created in the base database. The following fields are returned:

**product\_id**

The product which owns the archive.

**archive\_id**

The name of the archive.

**base\_seq\_no**

The uid of the baseline from which the archive was created.

**date\_time**

The create date of the archive.

**owner**

The user who created the archive.

**archive\_type**

The type of the archive which was created. This field is ARCHIVE, TOKEN or TRANSFER.

**media\_vol\_id**

The volume identifier of the media used.

**media\_label**

The label identifier of the media used.

**description**

The description of the archive.



## **pcms\_archive\_items - items that are contained in an archive**

This view lists all the items that are contained within a specific archive. The following fields are returned:

### **product\_id**

The product which owns the archive.

### **archive\_id**

The name of the archive.

### **archive\_type**

The type of the archive which was created. This field is ARCHIVE, TOKEN or TRANSFER.

### **item\_uid**

The uid of the items contained in this archive.

### **base\_seq\_no**

The uid of the baseline from which this archive was created.

### **item\_archive\_status**

The current status of the item in this archive such as online or offline.

## **pcms\_area\_catalogue - area catalog**

This view provides details on all defined areas.

### **AREA\_UID**

The UID for the area.

### **AREA\_ID**

The name of the area.

### **DIRECTORY**

The directory path that is defined for the area.

### **AREA\_TYPE**

The type of area (work, deployment, or library cache).

### **STATUS**

The current status.

### **OWNER**

The owner of the area.

### **CREATE\_DATE**

The date that the area was created.

### **ORIGINATOR**

The creator of the area.

## pcms\_attr\_map - attribute mapping

This view gives the attributes defined for any product type, request type, design part category, or product item type. The following fields are returned:

### **attr\_no**

The attribute number - an integer between 1 and 220.

### **variable**

The variable name for the attribute (uppercase). This is used (by default) as the field name for the attribute value in the generated views (see ["Notes About Published Views" on page 31](#)).

### **product\_id**

The name of a product for which the attribute is defined (uppercase).

### **type\_name**

The name of an object type for which the attribute is defined (uppercase). This may be a request type, a design part category, or a product item type.

### **data\_type**

A letter indicating the type of data held in the attribute. This may be 'C' for character data, 'N' for numeric data, or 'D' for date format data (see ["Notes About Published Views" on page 31](#)).

### **scope**

A letter indicating the corresponding class of objects: 'C' for request, 'P' for design part, 'I' for product item.

### **prompt**

The prompt string used to identify this attribute on forms displays and on displays by.

### **actual\_length**

The maximum number of characters allowable for values of this attribute.

### **vs\_uid**

The UID of a valid set against which values of this attribute are checked.

### **vs\_col\_no**

The applicable column within the valid set against which values of this attribute are checked.

## **pcms\_attributes\_baseline - baselines and their attributes**

This view returns user defined attribute information for all baselines in the main baseline catalog. The following fields are returned:

### **baseline\_uid**

The UID for the baseline concerned.

### **seq**

A number indicating the ordering of multi-valued attributes.

### **Attribute fields**

One attribute field corresponding to each possible attribute as defined through Object Type Definitions in the process model. The field name in each case is the same as the `variable_name`, and the retrieved length of the field is either the maximum length as defined in the Process Model, or 1978, whichever is the shorter.

## **pcms\_attributes\_chdoc - requests and their attributes**

This view returns user defined attribute information for all requests in the main requests catalog. The following fields are returned:

### **ch\_uid**

The UID for the request concerned.

### **seq**

A number indicating the ordering of multi-valued attributes.

## **pcms\_attributes\_item - items and their attributes**

This view returns user-defined attribute information for all product items. The following fields are returned:

### **item\_uid**

The UID for the particular item revision concerned.

### **seq**

A number indicating the ordering of multi-valued attributes.

## **pcms\_attributes\_part - parts and their attributes**

This view returns user defined attribute information for all design parts. The following fields are returned:

### **part\_uid**

The UID for the particular PCS of the design part concerned.

### **seq**

A number indicating the ordering of multi-valued attributes.

## **pcms\_attributes\_product - products and their attributes**

This view returns user defined attribute information for all products in the base database. The following fields are returned:

### **baseline\_uid**

The product part UID.

### **seq**

A number indicating the ordering of multi-valued attributes.

### **Attribute fields**

One attribute field corresponding to each possible attribute as defined through Object Type Definitions in the process model. The field name in each case is the same as the variable\_name, and the retrieved length of the field is either the maximum length as defined in the Process Model, or 1978, whichever is the shorter.



## **pcms\_attributes\_user - user-defined attributes for user**

This view gives user defined attribute information for a user. The following fields are returned:

### **user\_uid**

The UID of the user.

### **seq**

The attribute sequence number.

### **Attribute fields**

One attribute field corresponding to each possible attribute as defined through Object Type Definitions in the process model. The field name in each case is the same as the variable\_name, and the retrieved length of the field is either the maximum length as defined in the Process Model, or 1978, whichever is the shorter.

## **pcms\_attributes\_workset - projects and their attributes**

This view returns attribute information for all projects. The following fields are returned:

### **workset\_uid**

The UID of the project concerned.

### **seq**

A number indicating the ordering of multi-valued attributes.

### **Attribute fields**

One attribute field corresponding to each possible attribute as defined through Object Type Definitions in the process model. The field name in each case is the same as the variable\_name, and the retrieved length of the field is either the maximum length as defined in the Process Model, or 1978, whichever is the shorter.

## **pcms\_audit\_log**

This view returns audit logs for projects. The following fields are returned:

**audit\_seq**  
**user\_name**  
**cmd\_start\_time**  
**cmd\_end\_time**  
**elapsed\_msec**  
**srv\_hostname**  
**client\_hostname**  
**srv\_process\_id**  
**client\_ip**  
**connection\_ip**  
**app\_name**  
**cmd\_status**  
**cmd\_id**  
**cmd\_text**

## **pcms\_baseline\_attributes - baseline attribute information**

This view returns full attribute information for all product baselines. The following fields are returned:

### **baseline\_spec\_uid**

The UID for object specification.

### **baseline\_uid**

The UID for the baseline.

### **type\_uid**

The UID corresponding to the baseline type.

### **product\_id**

The name of the product to which the baseline belongs.

### **baseline\_id**

The name of the baseline.

### **baseline\_type**

The category of baseline. This can be release, design or archive (1 = Design baseline: 2 = Release Baseline: 3 = Archive Baseline).

### **seq**

A number indicating the ordering of multi-valued attributes.

### **Attribute fields**

One attribute field corresponding to each possible attribute as defined through Object Type Definitions in the process model. The field name in each case is the same as the `variable_name`, and the retrieved length of the field is either the maximum length as defined in the Process Model, or 1978, whichever is the shorter.

## **pcms\_baseline\_history - baseline update history**

This view gives update history recorded for a baseline. The following fields are returned:

**base\_seq\_no**

The baseline UID.

**current\_status**

The current status of the baseline.

**old\_status**

The previous status of the baseline.

**user\_name**

The name of the user who performed the update.

**history\_type**

The type of history record.

**date\_time**

The date and time the update was made.

**action\_no**

The action number for this operation.

**remark**

The comment for the update or type of operation performed.

## **pcms\_baseline\_info - baseline information**

This view gives overall details on baselines, one row per baseline. The following fields are returned:

### **product\_id**

The name of the product to which the baseline belongs (uppercase). The top design part in the baseline also belongs to this product.

### **baseline\_id**

The name of the baseline (uppercase).

### **base\_seq\_no**

A number distinguishing this baseline uniquely from all other baselines. This is used to join this view with other baseline views.

### **date\_time**

The date and time of creation of the baseline.

### **top\_node\_part\_id**

The name of the top design part in the baseline (uppercase).

### **top\_node\_part\_variant**

The variant of the top design part in the baseline (uppercase).

### **top\_node\_part\_pcs**

The product change status (PCS) of the top design part in the baseline (uppercase).

### **template\_id**

The name of the baseline template used to make the baseline (uppercase).

### **owner**

The user who owns the baseline (uppercase).

### **creation\_mechanism**

A letter indicating the method of creation of the baseline - B for creation of a part-scoped baseline by the CBL command, W for creation of a project baseline by the CBL command, D for creation by the CBL command using a request template, M for a merged baseline (CMB), R for a revised baseline (CRB).

### **baseline\_type**

A number indicating the category of the baseline (1 = Design baseline: 2 = Release Baseline: 3 = Archive Baseline).

### **status**

The current lifecycle status of the baseline.

### **type\_name**

The baseline type.

### **lifecycle\_id**

The name of the lifecycle defined for the baseline type.

**lc\_seq**

The sequence number of the current lifecycle status within the normal lifecycle.

**workset\_uid**

The UID of the project from which the baseline was created.

**workset**

The name of the project from which the baseline was created.

## **pcms\_baseline\_itemdirs - items and their file details contained in a baseline**

This view lists the file specifications of items that are contained in a baseline. The following fields are returned:

### **item\_uid**

The uid of the item included in a baseline.

### **base\_seq\_no**

The uid of the baseline.

### **filename**

The filename of the item included in the baseline. This does not include the directory specification.

### **dir\_fullpath**

The directory containing the item.

### **unix\_directory**

The UNIX directory containing the item.



## **pcms\_baseline\_items - items in baselines**

This view identifies the product items contained in baselines. The following fields are returned:

### **part\_uid**

The UID of a design part that is included in the baseline, and that is related to the above product item

### **item\_uid**

The UID of an item included in a baseline.

### **base\_seq\_no**

A number distinguishing the baseline uniquely from all other baselines. This is used to join this view with other baseline views.

### **dir\_uid**

The uid of the directory containing the item

### **filename**

The project filename for the product item contained within the baseline

### **dir\_fullpath**

The full directory specification for the product item within its project, as held in the Dimensions database. It comprises zero-to-many path components with a single forward slash separating each component from the next.

### **type\_of\_rel**

This is either "O" (Owned By) or "U" (Used By), reflecting if the item is owned or used by the design-part.

### **item\_spec\_uid**

The uid for the product item as a whole (all revisions) included in a baseline

## **pcms\_baseline\_norm\_lc - normal lifecycle information for baselines**

This view provides information on the normal lifecycle for types of baselines.

### **product\_id**

The name of the product to which the baselines belong.

### **baseline\_type**

The baseline type.

### **type\_uid**

The UID corresponding to the baseline type as defined on the relevant product.

### **state\_seq\_no**

The sequence number of the lifecycle state within the normal lifecycle.

### **status**

The name of a state in the normal lifecycle for the baselines of the type and product concerned.

### **lifecycle\_id**

The lifecycle ID.

## **pcms\_baseline\_parentage - pedigree of revised and merged baselines**

This view gives details on the pedigree of baselines created by the CRB and CMB functions. For a baseline created by CRB, there is just one predecessor baseline: for a baseline created by CMB, there may be any number of predecessors. The following fields are returned:

### **base\_seq\_no**

The distinguishing number of the baseline created by the CRB or CMB command.

### **parent\_product\_id**

The product owning a predecessor baseline (uppercase).

### **parent\_baseline\_id**

The name of a predecessor baseline (uppercase).

### **parent\_order**

The order of inclusion of the predecessor baseline. This is zero when the output baseline was created by CRB. For baselines created by CMB, this ranges from 1 upward.

## **pcms\_baseline\_parts - parts in baselines**

This view identifies the design parts contained in baselines and the relations between them. This view is suitable for use in hierarchical queries (see "[Hierarchical Queries](#)" on page 195). The following fields are returned:

### **parent\_part\_uid**

The UID for the parent design part in the relationship.

### **child\_part\_uid**

The UID for the child design part in the relationship.

### **base\_seq\_no**

A number distinguishing the baseline uniquely from all other baselines. This is used to join this view with other baseline views.

### **type\_of\_rel**

This indicates the type of relationship between the design parts. Currently this is either 'BREAKDOWN' or 'USAGE': but this may be changed in the future.

## **pcms\_baseline\_related\_chdocs - baseline and related requests**

This view returns the UIDs for the baseline and requests related to one another, together with information about the relationship between them. All relevant requests are returned, whether they are in the main or secondary request catalog. The following fields are returned:

### **from\_baseline\_uid**

The UID for the baseline in the relationship.

### **to\_ch\_uid**

The UID for the request in the relationship.

### **reltype\_name**

The type of relationship between the request and the baseline, for example, affected, in-response-to, or information.

### **parent\_class\_name**

The type of relationship between the request and the baseline, for example, affected, in-response-to, or information.

## **pcms\_baseline\_revision\_mods - baseline changes made by CRB**

This view gives details on modifications made to baselines by the CRB function. Each row records an item revision, the baseline and request involved, and a code to indicate whether the item revision was added to or removed from the baseline. The following fields are returned:

### **base\_seq\_no**

The distinguishing number of the baseline created by the CRB command.

### **ch\_uid**

The UID of the request used to revise the baseline, and citing the item involved.

### **product\_id**

The name of the product owning the product item involved (uppercase).

### **item\_id**

The identifier of the item involved (uppercase).

### **variant**

The variant of the item involved (uppercase).

### **item\_type**

The item type of the item involved (uppercase).

### **revision**

The specific revision of the item involved (uppercase).

### **action\_taken**

A letter indicating the action taken. 'U' indicates that the item revision was added to the baseline specified by base\_seq\_no, replacing another revision of the same item in the predecessor baseline. 'R' indicates that the item revision was removed without replacement. 'I' indicates that the item revision was added to the new baseline, and there was no revision of the item in the predecessor baseline.

## **pcms\_baseline\_template - baseline templates**

This view returns details on baseline templates. Each row applies to one template rule and gives the information displayed in the Administration Console Baseline Templates dialog. The following fields are returned:

### **template\_id**

The name of the template (uppercase).

### **date\_time**

The date and time of creation of the template.

### **item\_type**

The product item type to which this template rule applies (uppercase).

### **minimum\_status**

The lifecycle state to which this rule refers.

### **code**

The code (as shown in the Administration Console Baseline Templates dialog) used in conjunction with `minimum_status`.

### **description**

Description of the template rule.

## **pcms\_baseline\_types - baseline types**

This view returns details on baseline types. The following fields are returned:

### **type\_uid**

The UID corresponding to the baseline type as defined on the relevant product.

### **product\_id**

The name of the product to which the baseline type belongs.

### **baseline\_type**

The baseline type.

### **lifecycle\_id**

The name of the lifecycle defined for the baseline type

### **description**

The description of the baseline type as recorded in the database.



## **pcms\_bl\_rtm\_project\_xref - baseline to RM project associations**

This view provides information on associations between Dimensions CM baselines and Dimensions RM projects.

### **obj\_uid**

The unique identifier of the baseline.

### **rtm\_db\_name**

The name of the RM database.

### **rtm\_project\_id**

The unique identifier of the associated RM project.

### **rtm\_url**

The URL to the RM server.

### **product\_uid**

The unique identifier of the CM product.

## **pcms\_bld\_bline\_cfgver - Dimensions Build baseline configuration version**

This view provides details on Dimensions Build baseline configuration versions.

### **BLINE\_ID**

The ID of the baseline.

### **BUILD\_CONFIG\_VERSION\_ID**

The configuration version ID.

### **BLINE\_CFGVER\_ID**

The record ID of the relationship between the baseline and the build configuration version.

## **pcms\_bld\_build\_config - Dimensions Build configuration information**

This view provides details on Dimensions Build configurations.

### **BUILD\_CONFIG\_ID**

The ID of the Build configuration.

### **PROJECT\_ID**

The Build project ID.

### **PLATFORM\_ID**

The ID of the Build platform.

### **BUILD\_CONFIG\_CURRENT\_ID**

The current Build configuration ID.

### **BUILD\_CONFIG\_TYPE**

The ID of the build configuration type: Default/OpenMake.

### **BUILD\_CONFIG\_NAME**

The name of the Build configuration.

## **pcms\_bld\_build\_config\_version - Dimensions Build configuration version**

This view provides details on versions of Dimensions Build configurations.

### **BUILD\_CONFIG\_ID**

The ID of the Build configuration.

### **BUILD\_CONFIG\_VERSION\_COMMENT**

The check-in comment for the specific version.

### **BUILD\_CONFIG\_VERSION\_DATE**

The date that the version of the build configuration was created.

### **BUILD\_CONFIG\_VERSION\_ID**

The ID of the build configuration version.

### **BUILD\_CONFIG\_VERSION\_NUMBER**

The build configuration version number.

### **BUILD\_CONFIG\_VERSION\_USER**

The user who created the version.

## **pcms\_bld\_build\_env - Dimensions Build environment**

This view provides details on Dimensions Build environments. A Build environment represents the relationship between a Build configuration and an area in Dimensions CM.

### **BUILD\_ENV\_ID**

The ID of the Build environment (relationship between the Build configuration and the area).

### **BUILD\_CONFIG\_ID**

The ID of the Build configuration for the environment.

### **BUILD\_AREA\_ID**

The ID of the area for the environment.

### **BUILD\_ENV\_ASKPASSRUNTIME**

Whether a password is required at runtime in order to run the build.

### **BUILD\_ENV\_LOCKABLE**

Whether the Build environment can be locked.

## **pcms\_bld\_build\_job - Dimensions Build job**

This view provides details on Dimensions Build jobs.

### **BUILD\_JOB\_ID**

The ID of the Build job.

### **BUILD\_ENV\_ID**

The ID of the Build environment for the job.

### **BUILD\_AREA\_ID**

The ID of the Build area for the job.

### **BUILD\_JOB\_START\_TIME**

The start time for the Build job.

### **BUILD\_JOB\_STOP\_TIME**

The stop time for the Build job.

### **BUILD\_JOB\_USER**

The user who created the job.

### **BUILD\_JOB\_RESULT**

The result of the build job.

### **BUILD\_CONFIG\_VERSION\_ID**

The ID of the version of the Build configuration for the job.

### **CLEAN\_BUILD**

Whether the Build job should clean the target directory before running.

### **SRC\_BLINE\_ID**

The ID of the source baseline that the Build job compiles.

### **TGT\_PRJ\_ID**

The ID of the target project for the Build job.

### **BUILD\_JOB\_NOSRCTFR**

Whether the build job should transfer sources.

### **BUILD\_JOB\_NOTGTDEP**

Whether the build job should preserve targets.

## **pcms\_bld\_build\_job\_for\_target - Dimensions Build jobs for targets**

This view provides details on related Dimensions Build jobs and targets.

### **BUILD\_JOB\_FOR\_TARGET\_ID**

The ID of the relationship between the build job and the target.

### **BUILD\_JOB\_ID**

The ID of the Build job.

### **TARGET\_ID**

The ID of the target.

## **pcms\_bld\_build\_job\_target\_hist - Dimensions Build jobs and targets history**

This view provides history on specific targets in Build jobs.

### **BUILD\_JOB\_ID**

The ID of the Build job.

### **BUILD\_JOB\_TARGET\_BINDING**

The item data in Dimensions.

### **BUILD\_JOB\_TARGET\_FILE**

The target file for the build job.

### **BUILD\_JOB\_TARGET\_HISTORY\_ID**

The ID of the target history.

### **BUILD\_JOB\_TARGET\_RELPATH**

The relative path to the target.

### **BUILD\_JOB\_TARGET\_STATUS**

The current status of the build target.



## **pcms\_bld\_platform - Dimensions Build platforms**

This view provides history on Dimensions Build platform definitions.

### **PLATFORM\_ID**

The ID of the Build platform.

### **PLATFORM\_NAME**

The name of the Build platform

## **pcms\_bld\_target - Dimensions Build targets**

This view provides information on Build target definitions.

### **BUILD\_CONFIG\_ID**

The ID of the Build configuration that contains the target.

### **TARGET\_DESCRIPTION**

Description of the Build target.

### **TARGET\_FILE**

The Build target file.

### **TARGET\_ID**

The ID of the Build target.

### **TARGET\_IS\_FINAL**

Whether the Build target is final.

## **pcms\_bld\_type - Dimensions Build types**

This view provides information on Build types

### **TYPE\_ID**

The ID of the Build type.

### **TYPE\_CODE**

Build type code.

### **TYPE\_NAME**

The Build type name.

## **pcms\_bln\_history\_with\_attr - baseline update history including attributes**

This view gives update history including attribute updates for a baseline. The following fields are returned:

**part\_uid**

The part UID.

**action\_no**

The action number for this operation.

**date\_time**

The date and time the update was made.

**status**

The status of the baseline.

**user\_name**

The name of the user who performed the update.

**remark**

The comment for the update.

**history\_type**

The type of history record.

**attr\_no**

The attribute number affected by the operation (optional) - an integer between 1 and 220.

**seq**

The (optional) attribute sequence number.

**old\_attr\_val**

The old value of the attribute (if attribute tracking is enabled).

## **pcms\_bln\_rels2 - baseline relationships information**

This view provides information on baseline relationships.

### **rel\_uid**

The unique identifier of the relationship.

### **obj\_uid**

The unique identifier of the baseline.

### **related1\_uid**

The unique identifier of the first related object.

### **related2\_uid**

The unique identifier of the second related object.

### **reltype\_uid**

The unique identifier of the relationship type.

### **user\_name**

The user who created the relationship.

### **date\_time**

The date and time when the relationship was created.

## **pcms\_branches - registered branch names**

This view lists the version branch names registered in this base database.

### **branch\_uid**

A unique integer identifying the branch.

### **branch\_name**

The name of the version branch.

### **create\_date**

The date the branch was created.

### **site\_uid**

The uid of the base database site (see **site\_uid** in **pcms\_sites - registered base database sites** on [page 170](#)) that owns the branch. A null site\_uid means the branch is owned by this base database.

### **description**

The branch description.

## **pcms\_chdoc\_action\_desc - request action description**

This view returns the action description texts for those requests in the main request catalog for which this is held in the database. The following fields are returned:

### **ch\_uid**

The UID for the request concerned.

### **ch\_doc\_id**

The request identity (uppercase). This comprises the product\_id, the ch\_doc\_type and the doc\_seq number.

### **file\_version**

A number corresponding to the action at which the corresponding action description was entered.

### **seq**

This is not currently used.

### **data\_size**

The number of characters in the action description text held for the request.

### **data**

The action description text held for the request. This field is of the CLOB datatype.

## **pcms\_chdoc\_attr\_history - historic request attribute information**

This view returns historic attribute values that were updated when actioning a request, for example, specifying /ATTRIBUTES when executing the AC command. The following fields are returned:

### **ch\_uid**

The UID for the request concerned.

### **type\_uid**

The UID corresponding to the request type as defined on the relevant product for the corresponding request.

### **product\_id**

The name of the product to which the request belongs (uppercase).

### **ch\_doc\_type**

The request type (uppercase).

### **doc\_seq**

The sequential number of the request. This field is numeric, and is often used to sort a list of requests.

### **ch\_doc\_id**

The request identity (uppercase). This comprises the product\_id, the ch\_doc\_type and the doc\_seq number.

### **action\_no**

The action number at which the attribute set was applicable to the request.

### **seq**

A number indicating the ordering of multi-valued attributes.

### **Attribute fields**

One attribute field corresponding to each possible attribute as defined through Object Type Definitions in the process model. The field name in each case is the same as the variable\_name, and the retrieved length of the field is either the maximum length as defined in the Process Model, or 1978, whichever is the shorter.



## **pcms\_chdoc\_attachments - request attachments, primary catalog**

This view returns details for request attachments in the main request catalog, excluding any HELD requests. The following fields are returned:

### **ch\_uid**

The UID for the request concerned.

### **attachment\_uid**

The UID for the attachment.

### **filename**

The filename of the file attachment.

### **date\_time**

The date and time at which the attachment was created.

### **user\_name**

The user who created the attachment.

### **file size**

The size of the attachment in bytes.

### **description**

The description for the attachment.

## **pcms\_chdoc\_data - current request information**

This view returns full information for all requests in the main request catalog, excluding any HELD requests. The following fields are returned:

### **ch\_uid**

The UID for the request concerned.

### **type\_uid**

The UID corresponding to the request type as defined on the relevant product for the corresponding request.

### **product\_id**

The name of the product to which the request belongs (uppercase).

### **super\_type**

A number used to group request types.

### **ch\_doc\_type**

The request type (uppercase).

### **doc\_seq**

The sequential number of the request. This field is numeric, and is often used to sort a list of requests.

### **ch\_doc\_id**

The request identity (uppercase). This comprises the product\_id, the ch\_doc\_type and the doc\_seq number.

### **status**

The current lifecycle status of the request (uppercase).

### **create\_date**

The date and time of creation for the request.

### **action\_no**

The number of times that the request has been actioned.

### **cm\_phase**

A number used to indicate the current phase of the request.

### **originator**

The login identity of the user who originated the request (uppercase).

### **action\_date**

The date and time at which the request was last actioned.

### **update\_date**

The date and time at which the request was last updated.

### **owner\_site**

The site that owns the request.

**delegated\_site**

The site to which the request has been delegated.

**locked**

This is 'Y' or 'N' to indicate whether the request ownership is locked or not.

**reference\_only**

This is 'Y' or 'N' to indicate whether the request is reference only (cannot be updated).

**user\_name**

The login identity of the last user to action the request.

**seq**

A number indicating the ordering of multi-valued attributes.

**Attribute fields**

One attribute field corresponding to each possible attribute as defined through Object Type Definitions in the process model. The field name in each case is the same as the `variable_name`, and the retrieved length of the field is either the maximum length as defined in the Process Model, or 1978, whichever is the shorter.

## **pcms\_chdoc\_delegation - request delegations**

This view identifies current request delegations. The following fields are returned:

### **ch\_uid**

The UID of a request that has a delegation in force.

### **from\_user\_name**

The name of the user who set up the delegation (uppercase).

### **to\_user\_name**

The name of the user to whom a role has been delegated (uppercase).

### **role**

The role that has been delegated (uppercase).

### **capability**

The capability that has been delegated (uppercase): 'L' for leader, 'P' for primary, 'S' for secondary.

### **date\_time**

The date and time when the delegation was set up.

## **pcms\_chdoc\_detail\_desc - request detailed description**

This view returns the detailed description text for those requests in the main request catalog for which this is held in the database. The following fields are returned:

### **ch\_uid**

The UID for the request concerned.

### **ch\_doc\_id**

The request identity (uppercase). This comprises the product\_id, the ch\_doc\_type and the doc\_seq number.

### **file\_version**

A number corresponding to the number of times the text has been edited.

### **seq**

This is not currently used.

### **data\_size**

The number of characters in the detail description text held for the request.

### **data**

The detail description text held for the request. This field is of the LONG datatype.

## **pcms\_chdoc\_history - request history**

This view returns the history recorded for requests in the main request catalog. The following fields are returned:

### **ch\_uid**

The UID for the request concerned.

### **ch\_doc\_id**

The request identity (uppercase). This comprises the product\_id, the ch\_doc\_type and the doc\_seq number.

### **action\_no**

The action number at the time of the request history entry.

### **cm\_phase**

The phase of the request at the time of the history entry.

### **date\_amend**

The date and time of the history entry.

### **status**

The lifecycle status of the request (uppercase) at the time of the history entry.

### **user\_name**

The user responsible for the history entry.

### **action\_note**

Descriptive text for the history entry.

### **history\_type**

The type of history entry. Values are A for action, D for delegation and U for update.

---

## pcms\_chdoc\_history\_with\_attr - request history with attributes

This view returns the history recorded for requests in the main request catalog together with recorded historic attribute values. The following fields are returned:

**ch\_uid**

The UID for the request concerned.

**ch\_doc\_id**

The request identity (uppercase). This comprises the product\_id, the ch\_doc\_type and the doc\_seq number.

**action\_no**

The action number at the time of the request history entry.

**cm\_phase**

The phase of the request at the time of the history entry.

**date\_amend**

The date and time of the history entry.

**status**

The lifecycle status of the request (uppercase) at the time of the history entry.

**user\_name**

The user responsible for the history entry.

**action\_note**

Descriptive text for the history entry.

**history\_type**

The type of history entry. Values are A for action, D for delegation and U for update.

**hist\_uid**

The UID for the history entry.

**attr\_no**

The attribute number for an updated attribute. This will be NULL for any history entry that does not correspond to an updated attribute.

**seq**

A number indicating the ordering of multi-valued attributes. This will be NULL for any history entry that does not correspond to an updated attribute.

**old\_attr\_val**

The superseded value of an updated attribute. This will be NULL for any history entry that does not correspond to an updated attribute.

## **pcms\_chdoc\_norm\_lc - normal lifecycle for requests**

This view returns information about the normal lifecycle for requests of any type. Each row corresponds to one transition within a lifecycle. The following fields are returned:

### **product\_id**

The name of the product to which the requests belong (uppercase).

### **ch\_doc\_type**

The request type (uppercase).

### **type\_uid**

The UID corresponding to the request type as defined on the relevant product.

### **state\_seq\_no**

The sequence number of the lifecycle state within the normal lifecycle.

### **status**

The name of a state on the normal lifecycle for requests of the type and product concerned (uppercase).

### **lifecycle\_id**

The name of the lifecycle for the request type.



## **pcms\_chdoc\_related\_baselines - request and related baselines**

This view returns the UIDs for the request and baselines documents related to one another, together with information about the relationship between them. All relevant baselines are returned, regardless of whether the related requests are in the main or secondary request catalog. The following fields are returned:

### **from\_ch\_uid**

The UID for the request in the relationship.

### **to\_baseline\_uid**

The UID for the baseline in the relationship.

### **reltype\_name**

The type of relationship between the request and the baseline, for example, affected, in-response-to, or information.

### **parent\_class\_name**

The type of relationship between the request and the baseline, for example, affected, in-response-to, or information.

## **pcms\_chdoc\_related\_chdoc - related request information**

This view returns the UIDs for requests related to one another, together with information relating to the relationship between them. All relevant requests are returned, whether they are in the main or secondary request catalog. The following fields are returned:

### **from\_ch\_uid**

The UID for the parent request in the relationship.

### **to\_ch\_uid**

The UID for a request related as a child to the request identified by Dimensions CM from\_ch\_uid.

### **reltype\_name**

The relationship subclass applicable to the relationship between the requests identified by Dimensions CM from\_ch\_uid and to\_ch\_uid. If no subclass has been defined, this will default to the value of parent\_class\_name - either "Info" or "Dependent".

### **parent\_class\_name**

The system-defined relationship class name applicable to the relationship between the requests identified by Dimensions CM from\_ch\_uid and to\_ch\_uid. (This is either "Info" or "Dependent").

## **pcms\_chdoc\_related\_items - requests and related items**

This view returns the UIDs for requests and product items related to one another, together with information relating to the relationship between them. All relevant requests are returned, whether they are in the main or secondary request catalog. The following fields are returned:

### **from\_ch\_uid**

The UID for the request in the relationship (which is always in the parent position).

### **to\_item\_uid**

The UID for the product item in the relationship (which is always in the child position).

### **reltype\_name**

The relationship subclass applicable to the relationship between the request identified by `from_ch_uid` and the product item identified by `to_item_uid`.

### **parent\_class\_name**

The system-defined relationship class name applicable to the relationship between the request identified by `from_ch_uid` and the product item identified by `to_item_uid`.

## **pcms\_chdoc\_related\_parts - requests and related design parts**

This view returns the UIDs for requests and design parts related to one another, together with information relating to the relationship between them. All relevant requests are returned, whether they are in the main or secondary request catalog. The following fields are returned:

### **from\_ch\_uid**

The UID for the request in the relationship (which is always in the parent position).

### **to\_part\_uid**

The UID for the design part in the relationship (which is always in the child position).

### **reltype\_name**

The relationship subclass applicable to the relationship between the request identified by `from_ch_uid` and the design part identified by `to_part_uid`. Currently this will have the same value as `parent_class_name` - i.e. "Affected".

### **parent\_class\_name**

The system-defined relationship class name applicable to the relationship between the request identified by `from_ch_uid` and the design part identified by `to_part_uid`. (This is "Affected".)

## pcms\_chdoc\_related\_worksets - request project relationships

This view provides information about projects that are related to requests. The following fields are returned:

### **from\_ch\_uid**

The UID of the request.

### **to\_workset\_uid**

The UID of the related project.

### **reltype\_name**

The type of relationship between the change document and the project. This may be **Affected**, **In Response To**, **Dependent**, or **Info**.

### **parent\_class\_name**

The name of the parent project class.

## **pcms\_chdoc\_this\_action\_desc - request current action description**

This view returns the current action description text for those requests in the main request catalog for which this is held in the database. The following fields are returned:

### **ch\_uid**

The UID for the request concerned.

### **ch\_doc\_id**

The request identity (uppercase). This comprises the product\_id, the ch\_doc\_type and the doc\_seq number.

### **file\_version**

A number corresponding to the number of times the text has been edited.

### **seq**

This is not currently used.

### **data\_size**

The number of characters in the current action description text held for the request.

### **data**

The current action description text held for the request. This field is of the LONG datatype.

## **pcms\_chdoc\_types - request type information**

This view returns information relating to the definition of request types on particular products. The following fields are returned:

### **type\_uid**

The UID corresponding to the request type as defined on the relevant product.

### **product\_id**

The name of the product to which the request type belongs (uppercase).

### **ch\_doc\_type**

The request type (uppercase).

### **lifecycle\_id**

The name of the lifecycle defined for the request type (uppercase).

### **description**

The description of the request type as recorded in the database.

### **cntrl**

This is 'Y' or 'N' to indicate whether Change Management Rules are enabled or not.

### **relate\_state**

The minimum state the request must reside in before it can be related to items.

### **extract\_state**

The minimum state the request must reside in before a new item revision can be created in response to it.

### **freeze\_state**

The minimum state the request must be in to become frozen.

### **close\_state**

The minimum state the request must be in for it to be considered closed.

## **pcms\_chdoc\_user\_lists - request user lists**

This view returns the requests recorded in Request User Lists. The following fields are returned:

### **user\_name**

The login identity of the user who owns the Request User List.

### **list\_name**

The name of the Request User List.

### **ch\_uid**

The UID for the request concerned.



---

## pcms\_cm\_catalogue - basic request information

This view returns basic information for all requests in the system, whether they are in the main or secondary request catalog. The following fields are returned:

**ch\_uid**

The UID for the request concerned.

**type\_uid**

The UID corresponding to the request type as defined on the relevant product for the corresponding request.

**product\_id**

The name of the product to which the request belongs (uppercase).

**super\_type**

A number used to group request types.

**ch\_doc\_type**

The request type (uppercase).

**doc\_seq**

The sequential number of the request. This field is numeric, and is often used to sort a list of requests.

**ch\_doc\_id**

The request identity (uppercase). This comprises the product\_id, the ch\_doc\_type and the doc\_seq number.

**status**

The current lifecycle status of the request (uppercase).

**create\_date**

The date and time of creation for the request.

**action\_no**

The number of times that the request has been actioned.

**cm\_phase**

A number used to indicate the current phase of the request.

**originator**

The login identity of the user who originated the request (uppercase).

**action\_date**

The date and time at which the request was last actioned.

**update\_date**

The date and time at which the request was last updated.

**user\_name**

The login identity of the last user to action the request.

**owner\_site**

The site that owns the request.

**delegated\_site**

The site to which the request has been delegated.

**locked**

This is 'Y' or 'N' to indicate whether the request ownership is locked or not.

**reference\_only**

This is 'Y' or 'N' to indicate whether the request is reference only (cannot be updated).

**cat\_loc**

This is 'M' for a request in the main catalog or 'S' for a request in the secondary catalog.

**title**

The value of the first attribute field for the request: that is, the value of attribute number 1. This is normally used to hold the title of the request. If attribute definitions have been changed in the Process Model, or if it is desired to access the values of other attributes, the **pcms\_chdoc\_data** view ([page 74](#)) should be used instead (see also "[Notes About Published Views](#)" on [page 31](#)).

## **pcms\_command\_statistics - command statistics**

This view gives statistics for Dimensions CM commands that have been run against the database. The following fields are returned:

### **command**

The command that was run. This is just the primary command identifier, such as CI, EI, LWS, etc.

### **user\_name**

The login identifier of the person who ran the command.

### **no\_successes**

The number of times this command has been run successfully by this user.

### **no\_failures**

The number of times this command has been run unsuccessfully by this user.

### **date\_last\_run**

The last time the user ran this command.

## **pcms\_customer\_details - customer information**

This view gives details on customers as stored in the database. The following fields are returned:

### **customer\_uid**

The UID for the customer.

### **customer\_name**

Customer's name.

### **customer\_loc**

Customer's location.

### **customer\_project**

Customer's project details.

### **contact**

Contact name.

## **pcms\_customer\_releases - releases forwarded to customers**

This view identifies releases that have been forwarded to customers. The following fields are returned:

### **release\_uid**

The UID of a release forwarded to a customer.

### **customer\_uid**

The UID of the customer to whom the release has been forwarded.

### **date\_time**

The date and time of forwarding the release.

### **customer\_name**

Customer's name.

### **customer\_loc**

Customer's location.

### **customer\_project**

Customer's project details.

### **contact**

Contact name.

## **pcms\_deployment\_data - deployment data**

This view was introduced at Dimensions CM 12.1 and provides historical information on deployment. Together with "[pcms\\_promote\\_history - promotion history](#)" on page 136 it replaces the previous `pcms_deployment_history` view. The following fields are returned:

### **obj\_uid**

The UID of the object that was deployed. This is of type NUMBER.

### **obj\_spec\_uid**

The specification UID of the object that was deployed. This is of type NUMBER.

### **obj\_type**

The type of object that was deployed. This is of type VARCHAR2 (8) and can be ITEM, BASELINE, or CHDOC.

### **workset\_uid**

The UID of the project containing the object that was deployed. This is of type NUMBER.

### **area\_uid**

The UID of the area containing the object that was deployed. This is of type NUMBER.

### **stage\_uid**

The UID of the stage that the object was in when it was deployed. This is of type NUMBER.

### **user\_name**

The name of the user that deployed the object. This is of type VARCHAR2 (100).

### **date\_time**

The date and time that the object was deployed. This is of type DATE.

## **pcms\_deployment\_history - deployment history**

Starting with Dimensions CM 12.1, this view —which was for the pre-12.1 version of deployment— is no longer available. It has been replaced by "[pcms\\_promote\\_history - promotion history](#)" on page 136 and "[pcms\\_deployment\\_data - deployment data](#)" on page 94.

## **pcms\_derived\_formats - one-to-one derived build relationship**

This view lists the derived item formats that are used by the Dimensions Build process when doing a one-to-one (i.e. source-to-object) build. The following fields are returned:

### **product\_id**

The name of the product against which this definition is registered.

### **from\_item\_type**

The name of the source item type.

### **to\_item\_type**

The name of the target item type.

### **from\_format**

The name of the source item format.

### **to\_format**

The name of the target item format.



---

## **pcms\_environment\_items - environment item definitions**

This view returns details on environment item definitions. Each row defines a relationship between a primary item revision and an environment item revision. Users are advised that this view may be modified in the future by the omission of fields used to specify product items, retaining only UID fields for this purpose. The following fields are currently returned:

### **item\_spec\_uid**

The UID for the primary product item as a whole (all revisions).

### **item\_uid**

The UID for the particular revision of the primary item.

### **env\_item\_spec\_uid**

The UID for the environment product item as a whole (all revisions).

### **product\_id**

The name of the product owning the primary item (uppercase).

### **item\_id**

The identity of the primary item (uppercase).

### **variant**

The variant of the primary item (uppercase).

### **item\_type**

The item type of the primary item (uppercase).

### **revision**

The revision of the primary item (uppercase).

### **env\_product\_id**

The name of the product owning the environment item (uppercase).

### **env\_item\_id**

The identity of the environment item (uppercase).

### **env\_variant**

The variant of the environment item (uppercase).

### **env\_item\_type**

The item type of the environment item (uppercase).

### **tag**

The tag associated with the environment definition (if any).

## **pcms\_groups - groups definitions**

This view lists the groups that have been registered. The following fields are returned:

### **group\_uid**

The UID of the group.

### **group\_name**

The name of the group.

### **create\_date**

When the group was created.

### **update\_date**

When the group was last updated.

### **description**

The description of the group.

## **pcms\_group\_members - group members**

This view lists the users that are members in a group. The following fields are returned:

### **group\_uid**

The UID of the group.

### **user\_uid**

The UID of the user in the group.

### **user\_name**

The name of the user in the group.

### **date\_time**

When the user was added to the group.

## **pcms\_ipds - Item Process Definitions registered against a product**

This view lists the Item Process Definitions (IPD) that have been registered against a product. The following fields are returned:

### **product\_id**

The name of the product to which the IPD belongs.

### **i\_format**

The input format of the IPD.

### **o\_format**

The output format of the IPD.

### **ipd\_id**

The IPD identifier.

### **ipd\_src**

The **item\_uid** which is used to contain this definition.

---

## pcms\_item\_attributes - product item attribute information

This view returns full attribute information for all product items. The following fields are returned:

**item\_spec\_uid**

The UID for the product item as a whole (all revisions).

**item\_uid**

The UID for the particular item revision concerned.

**type\_uid**

The UID corresponding to the item type as defined on the relevant product for the corresponding item.

**product\_id**

The name of the product to which the item belongs (uppercase).

**item\_id**

The identifier of the item (uppercase).

**variant**

The variant of the item (uppercase).

**item\_type**

The name of the item type (uppercase). Note that this gives the name associated with type\_uid.

**revision**

The revision of the item (uppercase).

**seq**

A number indicating the ordering of multi-valued attributes.

**Attribute fields**

One attribute field corresponding to each possible attribute as defined through Object Type Definitions in the process model. The field name in each case is the same as the variable\_name, and the retrieved length of the field is either the maximum length as defined in the Process Model, or 1978, whichever is the shorter.

## **pcms\_item\_attr\_history - historic product item attribute information**

This view returns historic attribute values for all product items. The following fields are returned:

### **item\_spec\_uid**

The UID for the product item as a whole (all revisions).

### **item\_uid**

The UID for the particular item revision concerned.

### **type\_uid**

The UID corresponding to the item type as defined on the relevant product for the corresponding item.

### **product\_id**

The name of the product to which the item belongs (uppercase).

### **item\_id**

The identifier of the item (uppercase).

### **variant**

The variant of the item (uppercase).

### **item\_type**

The name of the item type (uppercase). Note that this gives the name associated with type\_uid.

### **revision**

The revision of the item (uppercase).

### **action\_no**

The action number at which the attribute set was applicable to the product item.

### **seq**

A number indicating the ordering of multi-valued attributes.

### **Attribute fields**

One attribute field corresponding to each possible attribute as defined through Object Type Definitions in the process model. The field name in each case is the same as the variable\_name, and the retrieved length of the field is either the maximum length as defined in the Process Model, or 1978, whichever is the shorter.

## pcms\_item\_data - current item information



### NOTE

The Dimensions CM 2009 R1 or later API supports large files (4GBytes or greater). For example, you can check in and fetch back a DVD ISO image. The length of the corresponding item library file in such circumstances will need to exceed 4 GB. To provide support for such item library files, internally Dimensions CM 2009 R1 or later uses 64-bit integers to represent library file lengths. Those 64-bit library lengths are stored as pairs of low-order 32-bits and high-order 32-bits.

This published view has been enhanced to reflect this by providing a new column, `lib_file_length_hi`, containing the high-order 32-bits of the file length. The low-order 32-bits of the file length continue to be stored in the `lib_file_length` column as in releases of Dimensions CM prior to 2009 R1.

This also means that the `lib_file_length` and `file_length` item attributes can potentially refer to 64-bit values. If you store files larger than 4 GB in Dimensions CM, and use DTK applications or event callouts, then you need to ensure that your code uses correct C functions for converting attribute values into 64-bit integers and vice versa. Refer to your compiler/C library vendor reference for information on how to convert 64-bit integers into null-terminated strings and vice versa.

This view returns information for all product-items in the item catalog. The following fields are returned:

### **item\_spec\_uid**

The UID for the product item as a whole (all revisions).

### **item\_uid**

The UID for the particular item revision concerned.

### **type\_uid**

The UID corresponding to the item type as defined on the relevant product for the corresponding item.

### **product\_id**

The name of the product to which the item belongs (uppercase).

### **item\_id**

The identifier of the item (uppercase).

### **variant**

The variant of the item (uppercase).

### **item\_type**

The name of the item type (uppercase). Note that this gives the name associated with `type_uid`.

### **revision**

The revision of the item (uppercase).

**lib\_filename**

The library filename of the item. See, also, the note at the beginning of this section.

**description**

Description of the item.

**status**

The current status of the item (uppercase).

**file\_version**

This is an integer specifying the version of the file stored in the item library. This information can be used to derive the most recently modified or created revision. This corresponds to the highest file version.

**create\_date**

Date of creation of the item revision as stored by the metadata tag date.

**revised\_date**

The date (in Julian format) that the operating-system date-time attributes of the operating-system file associated to the item revision was last revised.

**originator**

The login identity (uppercase) of the user who created the item revision.

**format**

The format of the item (uppercase).

**on\_line**

This is either 'Y' or 'N' to indicate whether the item file is online in its item library, or has been placed offline by means of Dimensions ART.

**lib\_file\_length**

The library file length of the item (low order 32-bits). See, also, the note at the beginning of this section.

**lib\_checksum**

The current checksum of the item.

**lib\_file\_length\_hi**

The library file length of the item (high order 32-bits). See, also, the note at the beginning of this section.

**line\_count**

The line count is determined on check in for text files and Unicode files (UTF-8/UTF-16LE, UTF-16BE, UTF-32LE, UTF-32BE-encoded files).



## **pcms\_item\_delegation - item delegations**

This view identifies current product item delegations. The following fields are returned:

**item\_uid**

The UID of a product item that has a delegation in force.

**from\_user\_name**

The name of the user who set up the delegation (uppercase).

**to\_user\_name**

The name of the user to whom a role has been delegated (uppercase).

**role**

The role that has been delegated (uppercase).

**capability**

The capability that has been delegated (uppercase): 'L' for leader, 'P' for primary, 'S' for secondary.

**date\_time**

The date and time when the delegation was set up.

## **pcms\_item\_exports - items checked out by users**

This view returns details on product items currently checked out by users. The following fields are returned:

### **item\_uid**

The UID of a product item that is currently checked out.

### **workset\_uid**

The UID of the project within which the item is currently checked out.

### **user\_name**

The name of the user who checked out the item.

### **date\_time**

The date and time at which the item was checked out.

### **user\_filename**

The file into which the product item has been checked out, as specified by the user at the time of checking out.

## **pcms\_item\_history - history of items**

This view gives the history recorded for product items. The following fields are returned:

### **item\_spec\_uid**

The UID for the product item as a whole (all revisions).

### **item\_uid**

The UID for the particular item revision concerned.

### **prev\_item\_uid**

The UID of the previous revision of this item. This field is relevant when the history entry relates to the creation of a new item revision or to the merging of one revision into another. In other cases, it is either zero or equal to the item\_uid field.

### **current\_status**

The lifecycle status of the product item (uppercase) at the time of the history entry.

### **old\_status**

In cases where the history entry relates to the actioning of the item revision from one lifecycle state to another, this is the state (in uppercase) immediately before the item was actioned. In other cases, this field is null.

### **user\_name**

The name of the user responsible for the history entry (in uppercase).

### **date\_time**

The date and time of the history entry.

### **action\_no**

The action number at the time of the item history entry.

### **remark**

Descriptive text for the history entry.

### **history\_type**

The type of history entry. Values are 'null' for action, 'A' for archived, 'D' for delegation, 'E' for extracted, 'M' for merged, 'U' for update, 'C' for create.

## **pcms\_item\_history\_with\_attr - item update history including attributes**

This view gives update history including attribute updates for an item. The following fields are returned:

**item\_uid**

The part UID.

**action\_no**

The action number for this operation.

**date\_time**

The date and time the update was made.

**status**

The status of the item.

**user\_name**

The name of the user who performed the update.

**remark**

Further details regarding the type of operation performed.

**history\_type**

The type of history record.

**attr\_no**

The attribute number affected by the operation (optional) - an integer between 1 and 220.

**seq**

The (optional) attribute sequence number.

**old\_attr\_val**

The old value of the attribute (if attribute tracking is enabled).

## **pcms\_item\_norm\_lc - normal lifecycle for product items**

This view returns information about the normal lifecycle for product items of any type. Each row corresponds to one transition within a lifecycle. The following fields are returned:

### **product\_id**

The name of the product to which the requests belong (uppercase).

### **item\_type**

The product item type (uppercase).

### **type\_uid**

The UID corresponding to the item type as defined on the relevant product.

### **state\_seq\_no**

The sequence number of the lifecycle state within the normal lifecycle.

### **status**

The name of a state on the normal lifecycle for items of the type and product concerned (uppercase).

### **lifecycle\_id**

The name of the lifecycle.

## **pcms\_item\_pending\_actions - users and their items to be actioned**

This view identifies users and their pending product items. The following fields are returned:

### **item\_uid**

The UID of a product item revision pending for the user given by user\_name.

### **from\_status**

The current lifecycle state (in uppercase) of the item revision given by item\_uid.

### **to\_status**

The next lifecycle state to which the item may be actioned (uppercase).

### **user\_name**

The identity of a user for whom the item revision identified by item\_uid is pending (uppercase).

### **role**

The role applicable to the item revision identified by item\_uid for the user given by user\_name (uppercase).

### **date\_time**

The date and time at which the item revision was placed in the user's inbox.

## **pcms\_item\_related\_items - user-defined relations between items**

This view returns the UIDs for product items that are related to one another, together with information relating to the relationship between them. The following fields are returned:

### **from\_item\_uid**

The UID for the parent item in the relationship.

### **to\_item\_uid**

The UID for the child item in the relationship.

### **user\_name**

The name of the user who created the relationship (uppercase).

### **relation**

The name of the relation between the product items.

### **rel\_description**

Description of the relation between the product items, as entered by the DIR function.

### **rel\_comment**

Comment on the relation between product items.

## **pcms\_item\_related\_parts - items and their related design parts**

This view returns the UIDs for items and design parts related to one another, together with information relating to the relationship between them. The following fields are returned:

### **item\_uid**

The UID for the item in the relationship.

### **part\_uid**

The UID for the design part in the relationship.

### **type\_of\_rel**

This is either "O" (Owned By) or "U" (Used By), reflecting if the item is owned or used by the design-part.



## **pcms\_item\_revision\_history – history of item revisions**

### **item\_spec\_uid**

The uid for the product item as a whole (all revisions).

### **item\_uid**

The uid for the particular item revision concerned

### **prev\_item\_uid**

The UID of the previous revision of this item. This field is relevant when the history entry relates to the creation of a new item revision or to the merging of one revision into another. In other cases, it is either zero or equal to the item\_uid field.

### **user\_name**

The name of the user responsible for the history entry (in uppercase)

### **date\_time**

The date and time of the history entry

### **status**

The lifecycle status of the product item (uppercase) at the time of the history entry

### **remark**

Descriptive text for the history entry

### **history\_type**

The item revision history type C, M or Z, where:

- C = create
- M = merged
- Z = deleted

## **pcms\_item\_spec\_data - current item specification**

This view returns information for all product-item specifications in the item catalog. The following fields are returned:

### **item\_spec\_uid**

The UID for the product item (all revisions).

### **type\_uid**

The UID corresponding to the item type as defined on the relevant product for the corresponding item.

### **product\_id**

The name of the product to which the item belongs (uppercase).

### **item\_id**

The identifier of the item (uppercase).

### **variant**

The variant of the item (uppercase).

### **item\_type**

The name of the item type (uppercase). Gives the name associated with type\_uid.

### **lib\_filename**

The library filename of the item.

### **description**

Description of the item.

---

## pcms\_item\_types - product item type information

This view returns information relating to the definition of product item types on particular products. The following fields are returned:

**type\_uid**

The UID corresponding to the product item type as defined on the relevant product.

**product\_id**

The name of the product to which the product item type belongs (uppercase).

**item\_type**

The product item type (uppercase).

**lifecycle\_id**

The name of the lifecycle defined for the product item type (uppercase).

**description**

The description of the product item type as recorded in the database.

**library**

The library used by to store the corresponding product items.

**node\_name**

The name of the network node where the item library is held.

**delta\_lib**

This indicates whether the item library is a delta library: it is 'Y' for a delta library, 'N' or null for a non-delta library.

**multi\_extract**

This is 'Y' or 'N' to indicate whether multiple extracts are allowed, or not.

**comment\_mandatory**

This is 'Y' or 'N' to indicate whether a user comment is mandatory when an edited item is returned, or not.

**autogen\_id**

This is 'Y' or 'N' to indicate whether the item identity is automatically generated, or not.

**header\_substitution**

This is 'Y' or 'N' to indicate whether header substitution is enabled, or not.

**compress\_file**

This is 'Y' or 'N' to indicate whether item files will be compressed in the item library, or not. This is applicable when the library is held on a UNIX node.

**overwrite\_initial**

This is 'Y' or 'N' to indicate whether revisions of an item can be edited at the initial state without producing a new revision, or not.

**primary\_cons**

This is 'Y' or 'N' to indicate whether it is a requirement for actioning items that there should be exactly one user with a primary role at the resulting state, or not.

**leader\_cons**

This is 'Y' or 'N' to indicate whether it is a requirement for actioning items that there should be exactly one user with a leader role at the resulting state, or not.

**all\_user\_overwrite**

This is 'Y' or 'N' to indicate whether (if the `overwrite_initial` flag is 'Y') an item revision may be edited at the initial state by any user with the appropriate role (Y), or only by the originator (N).

**checksum\_on**

This is 'Y' or 'N' to indicate whether checksum calculation is enabled, or not.

**use\_gui\_editor**

This is 'Y' or 'N' to indicate whether the inline editor will be used for items of this type, or not.

**cntrl**

This is 'Y' or 'N' to indicate whether Change Management Rules are enabled or not.

**chdoc\_req**

This is 'Y' or 'N' to indicate if a valid request is required before an item of this type can be created.

## **pcms\_life\_cycles - lifecycles**

This view returns details on lifecycles. Each row corresponds to one transition within a lifecycle. The following fields are returned:

### **lifecycle\_id**

The name of the lifecycle.

### **doc\_status**

The name of the state at the start of the transition.

### **role**

The role for which the transition is enabled.

### **next\_doc\_status**

The name of the state at the end of the transition.

### **norm\_lc**

This is 'Y' if the transition is part of the normal lifecycle.

## **pcms\_mail\_notif - e-mail notifications**

This view provides details on e-mail notifications. The following fields are returned:

### **notif\_uid**

The UID of the notification.

### **notif\_id**

The name of the e-mail notification.

### **create\_date**

The date that the e-mail notification was created.

### **update\_date**

When the notification was last updated.

### **notif\_type**

The type of event that the notification applies to.

### **type\_uid**

The UID of the notification type.

### **type\_flag**

This object class that the notification applies to.

### **description**

The description of the notification.

## **pcms\_mail\_subscribers - e-mail notification subscribers**

This view provides information on which users and groups have subscribed to e-mail notifications. The following fields are returned:

### **notif\_uid**

The UID of the of the e-mail notification that the user has subscribed to.

### **subscriber\_uid**

The UID of the subscribed user.

### **subscriber\_type**

The type of subscriber, such as a user or a group.

### **who\_assigned**

The name of the user who assigned the notification.

## **pcms\_make\_instance\_catalogue - network nodes used in the Dimensions Make process**

This view details the network node name and operating system on which an item preserved by Dimensions Make was created. The following fields are returned:

### **obj\_uid**

The uid of the network node on which items were made. This uid refers to the **build\_uid** in the view **pcms\_make\_mo\_list** on [page 121](#).

### **node\_name**

This details the name of the node that preserved the item.

### **software\_name**

This details the name of the operating system.

### **software\_version**

This details the version of the operating system.



---

## pcms\_make\_mo\_list - made-of list for items resulting from Dimensions Make

This view lists the input and derived output items that were created as a result of a build via Dimensions Make. The following fields are returned:

### from\_uid

The uid for the product item revision used as an input to Dimensions Make.

### to\_uid

The uid for the product item generated as an output by Dimensions Make.

### from\_fv

The file version of the input product item revision.

### to\_fv

The file version of the output item revision, refer to the VIEW **pcms\_item\_data** on [page 103](#) for more details.

### from\_workset\_uid

The uid for the project from which the input item came.

### to\_workset\_uid

The uid for the project to which the item resulting from the Dimensions Make process was added.

### rule\_uid

This uid refers to the **obj\_uid** in **pcms\_strings\_db** on [page 172](#).

### build\_uid

This uid refers to the **obj\_uid** in **pcms\_make\_instance\_catalogue** on [page 120](#).

## pcms\_mo\_list - made-of list

This view returns information about the use of product items by Configuration Build, but excludes items built by Dimensions CM. Each row corresponds to one product item as an input when another output product item is built. This view is suitable for use in hierarchical queries ([page 195](#)). The following fields are returned:

### from\_uid

The UID for the product item revision used as an input to configuration build.

### to\_uid

The UID for the product item revision generated as an output by configuration build.

### from\_fv

The file version of the input product item revision. (See the description of the **file\_version** field on [page 104](#).)

### to\_fv

The file version of the output product item revision. (See the description of the **file\_version** field on [page 104](#).)

### from\_workset\_uid

The uid for the project from which the input item came

### to\_workset\_uid

The uid for the project to which the item resulting from the process was added.

### rule\_uid

This uid refers to the obj\_uid in **pcms\_strings\_db** on [page 172](#) (pcms\_strings\_db - Dimensions Make commands used to make an item).

### build\_uid

This uid refers to the obj\_uid in **pcms\_make\_instance\_catalogue** on [page 120](#) (pcms\_make\_instance\_catalogue - network nodes used in the Dimensions Make process).

---

## pcms\_part\_attributes - design part attribute information

This view returns full attribute information for all design parts. The following fields are returned:

### **part\_spec\_uid**

The UID for the design part as a whole (all PCSs).

### **part\_uid**

The UID for the particular PCS of the design part concerned.

### **type\_uid**

The UID corresponding to the part category of the design part.

### **product\_id**

The name of the product to which the design part belongs (uppercase).

### **part\_id**

The part identity for the design part (uppercase).

### **variant**

The variant for the design part (uppercase).

### **pcs**

The PCS for the design part (uppercase).

### **part\_no**

The standard part number for this design part. This field is null if no standard part number has been allocated to the design part.

### **local\_no**

The local part number for this design part. This field is null if no local part number has been allocated to the design part.

### **seq**

A number indicating the ordering of multi-valued attributes.

### **Attribute fields**

One attribute field corresponding to each possible attribute as defined through Object Type Definitions in the process model. The field name in each case is the same as the `variable_name`, and the retrieved length of the field is either the maximum length as defined in the Process Model, or 1978, whichever is the shorter.

## **pcms\_part\_attr\_history - historic design part attribute information**

This view returns historic attribute values for all design parts. The following fields are returned:

### **part\_spec\_uid**

The UID for the design part as a whole (all PCSs).

### **part\_uid**

The UID for the particular PCS of the design part concerned.

### **type\_uid**

The UID corresponding to the part category of the design part.

### **product\_id**

The name of the product to which the design part belongs (uppercase).

### **part\_id**

The part identity for the design part (uppercase).

### **variant**

The variant for the design part (uppercase).

### **pcs**

The PCS for the design part (uppercase).

### **action\_no**

The action number at which the attribute set was applicable to the design part.

### **part\_no**

The standard part number for this design part. This field is null if no standard part number has been allocated to the design part.

### **local\_no**

The local part number for this design part. This field is null if no local part number has been allocated to the design part.

### **seq**

A number indicating the ordering of multi-valued attributes.

### **Attribute fields**

One attribute field corresponding to each possible attribute as defined through Object Type Definitions in the process model. The field name in each case is the same as the *variable\_name*, and the retrieved length of the field is either the maximum length as defined in the Process Model, or 1978, whichever is the shorter.

---

## pcms\_part\_data - design part information

This view returns information for all design parts in the system. It covers all PCSs of design parts, whether OPEN or CLOSED, as well as any SUSPENDED parts. The following fields are returned:

**part\_spec\_uid**

The UID for the design part as a whole (all PCSs).

**part\_uid**

The UID for the particular PCS of the design part concerned.

**type\_uid**

The UID corresponding to the part category of the design part.

**product\_id**

The name of the product to which the design part belongs (uppercase).

**part\_id**

The part identity for the design part (uppercase).

**variant**

The variant for the design part (uppercase).

**pcs**

The PCS for the design part (uppercase).

**category**

The category of the design part (uppercase). Note that this gives the name associated with type\_uid above.

**date\_time**

The date and time at which the design part was created.

**status**

The status of the design part (uppercase).

**owner\_id**

The login identity of the user who created the design part concerned (uppercase).

**part\_no**

The standard part number for this design part. This field is null if no standard part number has been allocated to the design part.

**local\_no**

The local part number for this design part. This field is null if no local part number has been allocated to the design part.

**description**

Description of the design part.

## **pcms\_part\_history - design part update history**

This view gives update history for a design part. The following fields are returned:

**part\_uid**

The part UID.

**current\_status**

The current status of the design part.

**old\_status**

The previous status of the design part.

**user\_name**

The name of the user who performed the update.

**history\_type**

The type of history record.

**date\_time**

The date and time the update was made.

**action\_no**

The action number for this operation.

**remark**

The comment for the update.

---

## **pcms\_part\_history\_with\_attr - design part update history including attributes**

This view gives update history including attribute updates for a design part. The following fields are returned:

**part\_uid**

The part UID.

**action\_no**

The action number for this operation.

**date\_time**

The date and time the update was made.

**status**

The status of the design part.

**user\_name**

The name of the user who performed the update.

**remark**

The comment for the update.

**history\_type**

The update type.

**hist\_uid**

The history uid for this operation (optional).

**attr\_no**

The attribute number (optional) - an integer between 1 and 220.

**seq**

The attribute sequence number (optional).

**old\_attr\_val**

The old value of the attribute.

## **pcms\_part\_types - design part type information**

This view returns information relating to the definition of design part categories on particular products. The following fields are returned:

### **type\_uid**

The UID corresponding to the design part category as defined on the relevant product.

### **product\_id**

The name of the product to which the design part category belongs (uppercase).

### **part\_type**

The design part category (uppercase).

### **lifecycle\_id**

The name of the lifecycle defined for the design part category (uppercase).

### **description**

The description of the design part category as recorded in the database.



## **pcms\_parts\_structure - structural relationships between design parts**

This view returns the UIDs for design parts related to one another, together with information relating to the relationship between them. This view is suitable for use in hierarchical queries (see [page 195](#)). The following fields are returned:

### **parent\_part\_uid**

The UID for the parent design part in the relationship.

### **child\_part\_uid**

The UID for the child design part in the relationship.

### **type\_of\_rel**

This indicates the type of relationship between the design parts. Currently this is a number - 1 for a breakdown relationship or 2 for a usage relationship: but this may be changed in the future.

## **pcms\_privileges - privileges**

This view returns details about privileges. The following fields are returned:

### **priv\_no**

The number of the privilege.

### **priv\_id**

The name of the privilege.

### **priv\_type**

The type of privilege. This can be any of the following:

- 1 (part)
- 2 (item)
- 3 (base database)
- 4 (baseline)
- 8 (request)
- 12 (product)
- 16 (release)
- 64 (project)

### **description**

The description of the privilege.

## **pcms\_privilege\_rules\_enabled - privilege rules enabled**

This view returns details about privilege rules that are enabled. The following fields are returned:

**priv\_no**

The number of the privilege.

**priv\_id**

The name of the privilege.

**rule\_no**

The number of the privilege rule.

**rule\_id**

The name of the privilege rule.

**name**

The name of the role, user, or group.

**product\_id**

The name of the product.

## **pcms\_product\_attributes - product attribute information**

This view returns full attribute information for all products. The following fields are returned:

### **part\_spec\_uid**

The product part spec UID.

### **part\_uid**

The product part UID.

### **type\_uid**

The UID corresponding to the product type.

### **product\_id**

The name of the product.

### **variant**

The product variant.

### **PCS**

The product PCS.

### **part\_no**

The product part number.

### **local\_no**

The product local number.

### **seq**

A number indicating the ordering of multi-valued attributes.

### **Attribute fields**

One attribute field corresponding to each possible attribute as defined through Object Type Definitions in the process model. The field name in each case is the same as the `variable_name`, and the retrieved length of the field is either the maximum length as defined in the Process Model, or 1978, whichever is the shorter.

## **pcms\_product\_attr\_upd\_history - product attribute update history**

This view gives historic attribute values for a product. The following fields are returned:

**part\_uid**

The UID of the product.

**product\_id**

The product ID.

**date\_amend**

The date and time the update was made.

**user\_name**

The name of the user who performed the update.

**action\_note**

The comment for the update.

**history\_type**

The update type.

**hist\_uid**

The history UID.

**attr\_no**

The attribute number - an integer between 1 and 220.

**seq**

The sequence number.

**old\_attr\_val**

The old value of the attribute.

## **pcms\_product\_catalogue - product catalog**

This view provides information on all products in the system. The following fields are returned:

**product\_id**

The name of the product.

**product\_uid**

The UID of the product.

**description**

The description of the product.

**originator**

The name of the user who created the product.

**create\_date**

The date the product was created.

**update\_date**

The date that the product was last updated.

## **pcms\_project\_areas - list projects associated to a deployment area**

This view lists projects associated to a deployment area. The following fields are returned:

### **project\_uid**

The unique identifier of the CM project.

### **area\_uid**

The UID for the area.

### **area\_offset**

The offset for the area.

### **stage\_uid**

The UID for the stage.

## **pcms\_promote\_history - promotion history**

This view was introduced at Dimensions CM 12.1 and provides historical information on promotion. Together with "[pcms\\_deployment\\_data - deployment data](#)" on page 94 it replaces the previous `pcms_deployment_history` view. The following fields are returned:

### **obj\_uid**

The UID of the object that was promoted. This is of type NUMBER.

### **obj\_spec\_uid**

The specification UID of the object that was promoted. This is of type NUMBER.

### **obj\_type**

The type of object that was promoted. This of type VARCHAR2 (8) and can be ITEM, BASELINE, or CHDOC.

### **workset\_uid**

The UID of the project containing the object that was promoted. This is of type NUMBER.

### **old\_stage**

The previous lifecycle stage that the object was in before it was promoted. This is of type VARCHAR2 (512).

### **new\_stage**

The stage that the object was promoted to. This is of type VARCHAR2 (512).

### **user\_name**

The name of the user that promoted the object. This of type VARCHAR2 (100).

### **date\_time**

The date and time that the object was promoted. This is of type DATE.

### **action\_no**

The action number for this operation. This is of type NUMBER.

### **remark**

The remark that accompanied the action. This is of type VARCHAR2 (4000).



## **pcms\_release\_data - release information**

This view gives overall details on releases, one row per release. The following fields are returned:

**release\_uid**

The UID for the release.

**product\_id**

The name of the product to which the release belongs (uppercase).

**release\_id**

The name of the release (uppercase).

**release\_dir**

The target release directory.

**user\_name**

The user who ran the release command.

**template\_id**

The name of the release template used to make the release (uppercase).

**description**

Description of the release.

**date\_time**

The date and time when the release was made.

**base\_seq\_no**

The distinguishing number for the baseline from which the release was made.

## **pcms\_release\_items - items in releases**

This view identifies the product items comprised in a release. The following fields are returned:

### **release\_uid**

The UID for the release.

### **item\_uid**

The UID of an item revision in the release.

### **sub\_dir**

The subdirectory into which the product was released.

## **pcms\_release\_template - release templates**

This view returns details on release templates. Each row represents one rule within a template, and comprises the information shown in the Administration Console Release Templates dialog. The following fields are returned:

### **template\_id**

The name of the release template (uppercase).

### **part\_id**

The part identity of the design part to which this rule applies, or '\*' to indicate any part identity (uppercase).

### **variant**

The design part variant to which this rule applies, or '\*' to indicate any variant (uppercase).

### **item\_type**

The item type for items to be selected for a release by this rule (uppercase).

### **sub\_directory**

The subdirectory into which items selected by this rule are to be released.

## **pcms\_replicator\_branches - branches replicated**

For a replication configuration, this view lists the branches that are replicated by a primary or a replica base database site.

### **config\_uid**

A unique integer identifying the primary or the replica replication definition.

### **branch\_uid**

The uid of the branch: join with **branch\_uid** in **pcms\_branches - registered branch names** on [page 70](#) for more details.

### **date\_time**

The date the branch association was created.

## **pcms\_replicator\_chdoc\_types - replicator request types**

For a replication configuration, this view lists the types of requests that are replicated by a primary or a replica base database site.

### **config\_uid**

The unique identifier for the replication configuration.

### **type\_uid**

The UID of the request types involved in the replication configuration.

### **date\_time**

The last update date of the configuration.

## **pcms\_replicator\_configs - registered replication configurations**

This view lists the Replication Configurations that have been registered in this base database.

### **config\_uid**

A unique integer identifying the primary or the replica replication definition.

### **config\_id**

The name of the replication configuration as set up in Replicator Administration Tool.

### **site\_uid**

The uid of the base database site acting as the primary or the replica site: join with **site\_uid** in **pcms\_sites - registered base database sites** on [page 170](#) for more details.

### **workset**

The spec of the primary or the replica project to and from which items are to be replicated. (This is only applicable for item replication.)

### **active**

Value is either "Y" or "N", denoting whether the primary or the replica site is enabled for replication. If this flag is set to "N" for the primary site of a configuration then no replication will occur in the context of that configuration.

### **replicate\_back**

Valid values are "Y" and "N". This field is relevant only for a replica definition and specifies whether the replica is to back replicate items to the primary when replication is invoked.

### **create\_date**

The creation date for the primary/replica definition.

### **update\_date**

The date of the last modification to the details on the primary/replica definition.

### **originator**

The operating system login id of the user who created this definition.

### **description**

A description of the primary/replica definition.

### **primary**

This is 'Y' or 'N' to indicate whether this is a primary replication definition or a replica definition.

### **config\_type**

The type of replication represented by the configuration, either ITEM/BASELINE/UNKNOWN or REQUEST.

**no\_network**

Online or offline (air gap) replication.

## **pcms\_replicator\_config\_rels - replica definitions in a primary definition**

For each primary definition in a replication configuration, this view lists the corresponding replica definitions.

### **master\_config\_uid**

The uid of the primary replication definition.

### **subord\_config\_uid**

The uid of the replica replication definition.

### **date\_time**

The date the replica definition was created.



---

## pcms\_replicator\_receive - inward transfers received

This view lists all the inward transfers that have been received by this base database from other base databases.

### receive\_uid

A unique integer identifying this inward transfer: join with **receive\_uid** in **pcms\_replicator\_receive\_items - item revisions received** on [page 147](#), **pcms\_replicator\_receive\_cdocs - requests received** on [page 146](#), and **pcms\_replicator\_receive\_logs - text logs received** on [page 148](#) for more details.

### config\_uid

The uid of the primary or replica definition identifying the context in which this inward transfer occurred: join with **config\_uid** in **pcms\_replicator\_configs - registered replication configurations** on [page 142](#) for more details.

### log\_uid

A unique integer identifying the transfer log: join with **log\_uid** in **pcms\_replicator\_receive\_items** on [page 147](#), **pcms\_replicator\_receive\_cdocs** on [page 146](#), and **pcms\_replicator\_receive\_logs** on [page 148](#) for more details.

### receive\_date

The date the inward transfer occurred.

### source\_site\_uid

The uid of the source base database site from which items were sent: join with **site\_uid** in **pcms\_sites - registered base database sites** on [page 170](#) for more details.

### source\_workset

The spec of the project on the source site from which items were replicated. (This is only applicable for item replication.)

### reference\_site

This is 'Y' or 'N' to indicate whether this site is a reference site (replicated requests cannot be updated).

## **pcms\_replicator\_receive\_cdocs -requests received**

This view lists all the requests received successfully by this base database from other base database sites.

### **receive\_uid**

The uid of the inward transfer: join with **receive\_uid** in **pcms\_replicator\_receive - inward transfers received** on [page 145](#) for more information.

### **log\_uid**

The uid of the transfer log for this request: join with **log\_uid** in **pcms\_replicator\_receive\_logs - text logs received** on [page 148](#) and **pcms\_replicator\_transfer\_logs** on [page 152](#) for more details.

### **ch\_uid**

The uid of the request which was received successfully: join with **ch\_uid** in the standard view **pcms\_chdoc\_data - current request information** on [page 74](#) for more details on the received request.

### **status**

Value is either "SUCCESS" or "FAIL", indicating whether the request was transferred successfully.

## **pcms\_replicator\_receive\_items - item revisions received**

This view lists all the item revisions received successfully by this base database from other base database sites.

### **receive\_uid**

The uid of the inward transfer: join with **receive\_uid** in **pcms\_replicator\_receive - inward transfers received** on [page 145](#) for more information.

### **log\_uid**

The uid of the transfer log for this item: join with **log\_uid** in **pcms\_replicator\_receive\_logs - text logs received** on [page 148](#) and **pcms\_replicator\_transfer\_logs** on [page 152](#) for more details.

### **item\_uid**

The uid of the item revision which was received successfully: join with **item\_uid** in the standard view **pcms\_item\_data - current item information** on [page 103](#) for more details on the received item revision.

### **receive\_date**

The date the inward transfer occurred.

### **source**

The source base database from which items were sent.

## pcms\_replicator\_receive\_logs - text logs received

This view lists the text logs associated with transfers received by this base database from other base databases.

### receive\_uid

The uid of the inward transfer: join with **receive\_uid** in **pcms\_replicator\_receive - inward transfers received** on [page 145](#) for more details.

### log\_uid

The uid of the inward transfer log for this item: join with **log\_uid** in **pcms\_replic\_intrans** for more details.

### log\_type

Specifies the type of the log:

Field Value	Meaning
LOG_ITEM	Indicates that this log is concerned with the transfer of an item revision and details any warning or error messages that may have occurred during the transfer.
LOG_CHDOC	Indicates that this log is concerned with the transfer of a request and details any warning or error messages that may have occurred during the transfer.
LOG_BASELINE	Indicates that this log is concerned with the transfer of a baseline and details any warning or error messages that may have occurred during the transfer.
LOG_INFO	Indicates that this log is concerned with any general warnings or errors that occurred during the replication process.

### obj\_uid

If "log\_type" is "LOG\_ITEM" this field identifies the uid of the item: join with "item\_uid" in the standard view **pcms\_item\_data - current item information** on [page 103](#) for details on the item revision.

If "log\_type" is "LOG\_CHDOC", this field identifies the uid of the request: join with "ch\_uid" in the standard view **pcms\_chdoc\_data - current request information** on [page 74](#) for details of the request.

If "log\_type" is "LOG\_BASELINE", this field identifies the uid of the baseline: join with "base\_seq\_no" in the standard view **pcms\_baseline\_info - current baseline information** on [page 46](#) for details of the baseline.

### line\_order\_seq

This field is supplied for convenience when using ORDER BY clauses. It can be used to show the line order of the transfer log. The values themselves have no intrinsic meaning.

### log\_text

The log text.

---

## pcms\_replicator\_transfer - outward transfers

This view lists all the outward transfers that occurred from this base database to other base database sites.

### transfer\_uid

A unique integer identifying this outward transfer: join with **transfer\_uid** in **pcms\_replicator\_transfer\_items - items for transfer** on [page 151](#), **pcms\_replicator\_transfer\_cdocs - requests for transfer** on [page 150](#), and **pcms\_replicator\_transfer\_logs - text logs associated with outward transfers** on [page 152](#) for more details.

### config\_uid

The uid of the primary or replica definition identifying the context in which this outward transfer occurred.

### log\_uid

A unique integer identifying the transfer log: join with **log\_uid** in **pcms\_replicator\_transfer\_items** on [page 151](#), **pcms\_replicator\_transfer\_cdocs** on [page 150](#), and **pcms\_replic\_outtrans\_logs** for more details.

### transfer\_date

The date the outward transfer occurred.

### target\_site\_uid

The uid of the target base database site to which items or requests were replicated: join with **site\_uid** in **pcms\_sites - registered base database sites** on [page 170](#) for more details.

### target\_workset

The spec of the project on the target site into which items were replicated. (This is only applicable for item replication.)

### reference\_site

This is 'Y' or 'N' to indicate whether the site to which requests were replicated is a reference site (replicated requests cannot be updated).

## **pcms\_replicator\_transfer\_cdocs - requests for transfer**

This view lists all the requests attempted for transfer from this base database to other base database sites, and for each request indicates whether the attempted transfer was successful.

### **transfer\_uid**

The uid of the transfer: join with **transfer\_uid** in **pcms\_replicator\_transfer - outward transfers** on [page 149](#) for more information.

### **log\_uid**

The uid of the transfer log for this request: join with **log\_uid** in **pcms\_replicator\_transfer\_logs - text logs associated with outward transfers** on [page 152](#) for more details.

### **ch\_uid**

The uid of the request which was attempted to be transferred: join with **ch\_uid** in the standard view **pcms\_chdoc\_data - current request information** on [page 74](#) for more details about the request.

### **status**

Value is either "SUCCESS" or "FAIL", indicating whether the item revision was transferred successfully.

## **pcms\_replicator\_transfer\_items - items for transfer**

This view lists all the items attempted for transfer from this base database to other base database sites, and for each item indicates whether the attempted transfer was successful.

### **transfer\_uid**

The uid of the transfer: join with **transfer\_uid** in **pcms\_replicator\_transfer - outward transfers** on [page 149](#) for more information.

### **log\_uid**

The uid of the transfer log for this item: join with **log\_uid** in **pcms\_replicator\_transfer\_logs - text logs associated with outward transfers** on [page 152](#) for more details.

### **item\_uid**

The uid of the item revision which was attempted to be transferred: join with **item\_uid** in the standard view **pcms\_item\_data - current item information** on [page 103](#) for more details about the item revision.

### **status**

Value is either "SUCCESS" or "FAIL", indicating whether the item revision was transferred successfully.

## pcms\_replicator\_transfer\_logs - text logs associated with outward transfers

This view lists the text logs associated with outward transfers that occurred from this base database to other base databases.

### transfer\_uid

The uid of the transfer: join with **transfer\_uid** in **pcms\_replicator\_transfer - outward transfers** on [page 149](#) for more details.

### log\_uid

The uid of the transfer log for this object.

### log\_type

Specifies the type of the log:

Field Value	Meaning
LOG_ITEM	Indicates that this log is concerned with the transfer of an item revision and details any warning or error messages that may have occurred during the transfer.
LOG_CHDOC	Indicates that this log is concerned with the transfer of a request and details any warning or error messages that may have occurred during the transfer.
LOG_BASELINE	Indicates that this log is concerned with the transfer of a baseline and details any warning or error messages that may have occurred during the transfer.
LOG_INFO	Indicates that this log is concerned with any general warnings or errors which occurred during the replication process

### obj\_uid

If "log\_type" is "LOG\_ITEM", this field identifies the uid of the item revision: join with "item\_uid" in the standard view **pcms\_item\_data - current item information** on [page 103](#) for details of the item.

If "log\_type" is "LOG\_CHDOC", this field identifies the uid of the request: join with "ch\_uid" in the standard view **pcms\_chdoc\_data - current request information** on [page 74](#) for details of the request.

If "log\_type" is "LOG\_BASELINE", this field identifies the uid of the baseline: join with "base\_seq\_no" in the standard view **pcms\_baseline\_info - current baseline information** on [page 46](#) for details of the baseline.

### line\_order\_seq

This field is supplied for convenience when using ORDER BY clauses. It can be used to show the line order of the transfer log. The values themselves have no intrinsic meaning.

### log\_text

The log text.



---

## **pcms\_report\_chdoc\_data - current request information used for reporting purposes**

This view returns full information for all requests in the main request catalog, excluding any HELD requests. The following fields are returned:

### **chide**

The UID for the request concerned.

### **type\_uid**

The UID corresponding to the request type as defined on the relevant product for the corresponding request.

### **product\_id**

The name of the product to which the request belongs (uppercase).

### **supportable**

A number used to group request types.

### **ch\_doc\_type**

The request type (uppercase).

### **doc\_seq**

The sequential number of the request. This field is numeric, and is often used to sort a list of requests.

### **ch\_doc\_id**

The request identity (uppercase). This comprises the product\_id, the ch\_doc\_type and the doc\_seq number.

### **status**

The current lifecycle status of the request (uppercase).

### **create\_date**

The date and time of creation for the request.

### **action\_no**

The number of times that the request has been actioned.

### **cm\_phase**

A number used to indicate the current phase of the request.

### **originator**

The login identity of the user who originated the request (uppercase).

### **action\_date**

The date and time at which the request was last actioned.

### **update\_date**

The date and time at which the request was last updated.

**user\_name**

The login identity of the last user to action the request.

**seq**

A number indicating the ordering of multi-valued attributes.

**Attribute fields**

One attribute field corresponding to each possible attribute as defined through Object Type Definitions in the process model. The field name in each case is the same as the `variable_name`, and the retrieved length of the field is either the maximum length as defined in the Process Model, or 1978, whichever is the shorter.

---

## pcms\_role\_sections - role sections

This view returns information about Role Sections. These are created and maintained by the Administration Console Role Definitions and Assignments dialog, and they both define groupings of attributes for displays and specify whether attributes are updatable and/or mandatory.

For details about the Administration Console functions, see the *online help*.

The following fields are returned:

**product\_id**

The name of the product concerned (uppercase).

**type\_name**

The name of the object type concerned (uppercase).

**type\_flag**

This indicates the kind of object concerned. Values are B for baseline, C for request, I for product item, or P for design part.

**lifecycle\_id**

The name of the lifecycle defined for the object type (uppercase).

**type\_uid**

The UID corresponding to the object type.

**attr\_no**

The attribute number for the attribute concerned (a number between 1 and 220).

**role**

The role for which this rule is defined (uppercase).

**from\_status**

The current status applicable to this rule (uppercase).

**mandatory**

This is 'Y' to indicate that this attribute is mandatory in the circumstances defined in this rule: otherwise it is 'N', blank or null.

**updateable**

This is 'Y' to indicate that this attribute may be updated in the circumstances defined in this rule: otherwise it is 'N', blank or null.

**displayed**

This is 'Y' to indicate that this attribute is displayed in the role section defined in this rule: otherwise it is 'N', blank or null.

**to\_status**

The new status applicable to this rule (uppercase).

## **pcms\_rtm\_dm\_project\_col\_xref - RM collections associations**

This view provides information on associations between Dimensions RM collections and Dimensions CM projects.

### **dm\_project\_uid**

The unique identifier of the CM project.

### **rtm\_collection**

The associated RM collection.

### **active**

This column is not currently used.

### **baseline**

Whether the RM collection is in fact a Dimensions RM baseline.

## **pcms\_rtm\_object - requirements associations**

This view provides information on associations between Dimensions RM requirements and Dimensions CM requests.

### **REQ\_OBJ\_UID**

The unique identifier of the RM requirement.

### **RTM\_CLASS\_ID**

The unique identifier of the RM requirement class.

### **RTM\_PROJ\_ID**

The RM requirement project unique identifier.

### **RTM\_OBJECT\_ID**

The RM requirement ID. This identifies the version of the requirement.

### **REQUEST\_UID**

The associated request UID.

### **SUSPECT**

This column is not currently used.

## **pcms\_rtm\_project\_product\_xref - Dimensions RM project association**

This view provides information on associations between Dimensions RM projects and Dimensions CM products.

### **RTM\_DB**

The RM database name.

### **RTM\_PROJECT\_ID**

The name of the RM project.

### **RTM\_URL**

The Dimensions RM server URL.

### **PRODUCT\_UID**

The unique identifier of the CM product that is associated with the RM project.

### **PRODUCT\_ID**

The name of the CM product that is associated with the RM project.

## **pcms\_rules- rule details**

This view returns details on rules, including the rule number, ID, and description.

### **rule\_no**

The rule number.

### **rule\_id**

The rule name.

### **description**

Description of the rule.

## **pcms\_rule\_privileges - rules applicable for each privilege**

This view returns details on which rules are applicable for each privilege, including the privilege number, privilege ID, rule number, and rule ID.

### **priv\_no**

The privilege number.

### **priv\_id**

The privilege name.

### **rule\_no**

The rule number.

### **rule\_id**

The rule name.



## **pcms\_sec\_chdoc\_action\_desc - secondary request action description**

This view returns the action description texts for those requests in the secondary request catalog for which this is held in the database. The fields returned by this view have the same names and meaning as in the **pcms\_chdoc\_action\_desc** view ([page 71](#)).

## **pcms\_sec\_chdoc\_attachments - request attachments, secondary catalog**

This view returns details for request attachments in the secondary request catalog. The fields returned by this view have the same names and meaning as in the **pcms\_chdoc\_attachments** view ([page 73](#)).

### **ch\_uid**

The UID for the request concerned.

### **attachment\_uid**

The UID for the attachment.

### **filename**

The filename of the file attachment.

### **date\_time**

The date and time at which the attachment was created.

### **user\_name**

The user who created the attachment.

### **file size**

The size of the attachment in bytes.

### **description**

The description for the attachment.

## **pcms\_sec\_chdoc\_attr\_history - historic secondary request attribute information**

This view returns historic attribute values for all requests in the secondary request catalog. The fields returned by this view have the same names and meaning as in the **pcms\_chdoc\_attr\_history** view ([page 73](#)).

## **pcms\_sec\_chdoc\_data - secondary request information**

This view returns full information for all requests in the secondary request catalog. The fields returned by this view have the same names and meaning as in the **pcms\_chdoc\_data** view (on [page 74](#)).

## **pcms\_sec\_chdoc\_detail\_desc - secondary request detailed description**

This view returns the detailed description text for those requests in the secondary request catalog for which this is held in the database. The fields returned by this view have the same names and meaning as in the **pcms\_chdoc\_detail\_desc** view ([page 77](#)).

## **pcms\_sec\_chdoc\_history - secondary request history**

This view returns the history recorded for requests in the secondary request catalog. The fields returned by this view have the same names and meaning as in the **pcms\_chdoc\_history** view ([page 78](#)).

## **pcms\_sec\_history\_with\_attr - secondary request history with attributes**

This view returns the history recorded for requests in the secondary request catalog together with recorded historic attribute values. The fields returned by this view have the same names and meaning as in the **pcms\_chdoc\_history\_with\_attr** view ([page 79](#)).

## **pcms\_sec\_status\_rpt1 - secondary requests on normal lifecycle**

This view returns information about requests in the secondary request catalog currently on their normal lifecycle. The fields returned by this view have the same names and meaning as in the **pcms\_status\_rpt1** view ([page 168](#)).



## **pcms\_sec\_this\_action\_desc - secondary request current action description**

This view returns the current action description text for those requests in the secondary request catalog for which this is held in the database. The fields returned by this view have the same names and meaning as in the **pcms\_chdoc\_this\_action\_desc** view ([page 86](#)).

## **pcms\_sites - registered base database sites**

This view lists the base database sites that have been registered in the database using the Administration Console.

### **site\_uid**

A unique integer identifying the base database site.

### **node\_name**

The name of the network node.

### **base\_db**

The name of the base database.

### **oracle\_sid**

The identifier of the oracle database instance hosting the base database.

### **site\_id**

A lexical identifier constructed from `node_name`, `base_db` and data source name (`dsn`) denoting the base database.

The format is `<node_name>:<base_db>@<dsn>`.

---

## **pcms\_status\_rpt1 - requests on normal lifecycle**

This view returns information about requests in the main request catalog currently on their normal lifecycle. The following fields are returned:

### **ch\_uid**

The UID for the request concerned.

### **ch\_doc\_id**

The request identity (uppercase). This comprises the product\_id, the ch\_doc\_type and the doc\_seq number.

### **type\_uid**

The UID corresponding to the request type as defined on the relevant product for the corresponding request.

### **ch\_doc\_type**

The request type (uppercase).

### **status**

The current lifecycle status of the request (uppercase).

### **norm\_lc\_seqno**

The sequence number of the current lifecycle status within the normal lifecycle.

### **product\_id**

The name of the product to which the request belongs (uppercase).

### **lifecycle\_id**

The name of the lifecycle applicable to the request.

### **create\_date**

The date and time of creation for the request.

### **action\_date**

The date and time at which the request was last actioned.

### **update\_date**

The date and time at which the request was last updated.

## **pcms\_strings\_db - Dimensions Make commands used to make an item**

This view details the Dimensions Make command which was used to create a specific output item. This view is used only by *pcmsmake* or *pnmake*. The following fields are returned:

### **obj\_uid**

The uid of the command string which was used to build an item. This uid refers to **rule\_uid** in view **pcms\_make\_mo\_list** on [page 121](#).

### **obj\_size**

The size of the full Dimensions Make command string.

### **string**

A partial listing of the Dimensions Make command string.

### **full\_string**

A full listing of the Dimensions Make command string. This field is represented as a LONG datatype and is used in conjunction with **obj\_size**.

## **pcms\_user\_attributes - user attribute information**

This view returns full attribute information for all users. The following fields are returned:

**user\_name**

The username.

**user\_uid**

The user UID.

**pcms\_db**

The user database name.

**base\_db**

The base database name.

**group\_id**

The user properties.

**site**

The user site.

**dept**

The user department.

**full\_name**

The user full name.

**phone**

The user telephone number.

**seq**

A number indicating the ordering of multi-valued attributes.

**Attribute fields**

One attribute field corresponding to each possible attribute as defined through Object Type Definitions in the process model. The field name in each case is the same as the `variable_name`, and the retrieved length of the field is either the maximum length as defined in the Process Model, or 1978, whichever is the shorter.

## **pcms\_user\_attributes\_def - user-defined attributes**

This view identifies default user-defined attributes. The following fields are returned:

### **attr\_no**

The attribute number - an integer between 1 and 120.

### **attr\_scope**

A letter indicating the corresponding class of object: 'C' for request, 'P' for design part, 'I' for item, 'B' for baseline.

### **attr\_type**

The type of attribute: 'SVA' represents single-valued attribute and 'MVA' represents multi-valued attribute.

### **attr\_data\_type**

Text indicating the type of data held in the attribute: 'CHAR' for character data, 'NUMBER' for numeric data, or 'DATE' for date format data.

### **product\_id**

The name of the product for which the attribute is defined.

### **type\_name**

The name of the object type for which the attribute is defined. This may be a request type, a design part category, a product item type or a baseline type.

### **type\_class**

The object class. This can be 'CHDOC' for request, 'PART' for design part, 'BASELINE' for baseline or 'ITEM' for item type.

### **user\_prompt**

The prompt string used to identify this attribute on client displays.

### **help\_message**

The prompt string used to help the user identify the attribute.

### **displayed**

This is used to indicate whether the attribute is displayed or not.

### **display\_length**

A number indicating the display length for the attribute field.

### **maximum\_length**

A number indicating the maximum length for the attribute field.

### **order\_of\_display**

This sets the order in which the attributes are displayed.

### **height**

A number indicating the number of lines displayed. This is only valid for multi-valued attributes.

**default\_value**

The default value for the attribute, as defined in the Administration Console.

**all\_revisions\_same\_value**

Flag which can be Y or N which sets whether the attribute value is the same for subsequent revisions.

## **pcms\_user\_attr\_update\_history - user attribute update history**

This view gives historic attribute values for a user. The following fields are returned:

**user\_uid**

The UID of the user.

**user\_name**

The user name.

**date\_amend**

The date and time the update was made.

**amended\_by**

The name of the user who performed the update.

**action\_note**

The comment for the update.

**hist\_uid**

The history uid.

**attr\_no**

The attribute number - an integer between 1 and 220.

**seq**

The sequence number.

**old\_attr\_val**

The old value of the attribute.



---

## pcms\_user\_audit\_trail - user audit trail

This view gives an audit trail of operations requiring authentication (electronic signature) that were attempted by a user. The following fields are returned:

### **audit\_uid**

A number that uniquely identifies the audit event.

### **user\_uid**

The UID of the user who performed the operation that created the audit event.

### **obj\_uid**

The UID of the operand if the operation involved a request or item, otherwise -1.

### **obj\_type**

The type of the operand. The value of this is one of "ITEM", "CHDOC", "ATTRIBUTE", or "UNKNOWN".

### **lifecycle\_id**

The lifecycle ID if the operation involved changing the sensitivity of a lifecycle state or deleting a lifecycle state, otherwise "\$UNDEFINED".

### **status**

If lifecycle\_id is not "\$UNDEFINED" this is the lifecycle status that was modified.

If lifecycle\_id is "\$UNDEFINED" and this column is not "\$UNDEFINED" an action operation was attempted, and this column contains the target state.

If this column is "\$UNDEFINED" an attribute operation was attempted.

### **success**

Y if authentication succeeded, N if it did not.

### **attr\_no**

The number of the attribute of any request or item that was updated, otherwise 0.

### **date\_time**

The date and time the event was recorded.

### **remark**

A short description of the operation attempted.

## pcms\_user\_pending\_chdoc - user and pending request information

This view identifies users and their pending requests. The following fields are returned:

### ch\_uid

The UID for a request pending for the user given by user\_name.

### user\_name

The login identity of a user for whom the request identified by ch\_uid is pending (uppercase).

### role

The role applicable to the pending request identified by ch\_uid for the user given by user\_name.

### next\_status

The next lifecycle state to which the request may be actioned (uppercase).

### pending\_date

The date and time at which the request was actioned to the user given by user\_name.

### pending

Indicates whether the request is pending (value of Y) or not (value of N).

### actionable

A numeric field indicating whether the user can update and/or action the request:

Value	Description
1	User may enter an action description, but may not action the request (this must be done by a user with a leader role).
2	User may update attributes (if allowed for the attribute), enter an action description, and action the request to a new state.
3	User may update attributes (if allowed for the attribute), enter an action description, action the request to a new state, and edit all currently waiting action descriptions (which may have been entered by another user). This is applicable to a user who holds the relevant role and is marked as the leader.

### capability

The capacity in which the request has been actioned to the user. This may be P (for primary), S (for secondary) or L (for leader).

## **pcms\_user\_roles - roles allocated to users**

This view gives details on the allocation of roles to users. The following fields are returned:

### **user\_name**

The name of the user to whom a role is allocated (uppercase).

### **role**

The role allocated to the user (uppercase).

### **product\_id**

The name of the product for which the role is allocated (uppercase).

### **part\_id**

The identity of the design part at which the role is allocated (uppercase).

### **part\_variant**

The design part variant at which the role is allocated (uppercase). A null value indicates that the role is allocated for all variants of the design part.

### **leader**

The capability associated with the role allocation. This may be 'L' for leader, 'P' for primary or 'S' for secondary.

### **workset\_id**

The project (belonging to the given product) for which the role is allocated (uppercase). This may be null, indicating all projects on the product.

### **assign\_type**

This indicates the type of role assignment: 'R' means that this is an actual role assignment, while 'C' means that the user has been assigned the role as a candidate for possible delegations.

## **pcms\_users\_profile - registered users**

This view returns details on the users registered for the current base database. The following fields are returned:

### **user\_uid**

The UID for the user concerned.

### **user\_name**

The login identity of a registered user.

### **pcms\_db**

The database account name by which the user accesses the database. This is '\*' for a user registered as a proxy user.

### **base\_db**

The database account name for the base database.

### **group\_id**

The group identity for the user.

### **site**

The user's site.

### **dept**

The user's department.

### **full\_user\_name**

The user's full name.

### **phone**

The user's telephone number.

### **email\_addr**

The user's e-mail address.

### **privilege\_level**

A number indicating the type of database user. This is 0 for a normal user and 1 for a proxy user. A value of -1 indicates a former user, who has been made inoperative by the DUSR function.

### **default\_workset\_uid**

The uid of the default project.

## **pcms\_valid\_set\_view - valid sets**

This view returns the identity and contents of valid sets. The following fields are returned:

**vs\_uid**

The UID for a valid set.

**product\_id**

The name of the product to which the valid set belongs (uppercase).

**vs\_name**

The name of the valid set.

**no\_cols**

The number of columns in the valid set.

**validation\_error\_msg**

The error message returned on a validation failure.

**description**

A description of the valid set.

**display\_order**

A number indicating the order or rows within a valid set.

**col1**

**col2**

**col3**

**col4**

**col5**

**col6**

**col7**

**col8**

The values stored in the corresponding columns of the valid set.

## **pcms\_workset\_attributes - project attributes**

This view returns the identity and contents attributes defined for projects. The following fields are returned:

### **workset\_uid**

The UID of the project.

### **product\_id**

The name of the product to which the project belongs.

### **workset\_spec\_uid**

The UID of the object specification.

### **type\_uid**

The UID corresponding to the project type.

### **workset\_id**

The name of the project.

### **workset\_type**

The name of the project type. This is the name associated with `type_uid`.

### **seq**

A number indicating the ordering of multi-valued attributes.

### **Attributes**

One attribute field corresponding to each possible attribute as defined through Object Type Definitions in the process model. The field name in each case is the same as the `variable_name`, and the retrieved length of the field is either the maximum length as defined in the Process Model, or 1978, whichever is the shorter.

## **pcms\_workset\_branches - project version branch names**

This view lists the valid version branch names for a project.

### **workset\_uid**

The UID of the project.

### **branch\_uid**

A unique integer identifying the branch.

### **branch\_name**

The name of the version branch.

## **pcms\_workset\_components - project components**

This view provides information on subprojects of subbaselines related to a project. The following fields are returned:

### **workset\_uid**

The UID of the project.

### **component\_uid**

The UID of the component.

### **obj\_type**

The type of object.

### **user\_name**

The name of the user who created the component.

### **date\_time**

The date and time the component was created.



## **pcms\_workset\_history - project history**

This view gives update history recorded for a project. The following fields are returned:

**workset\_uid**

The UID of the project.

**user\_name**

The name of the user who performed the update.

**date\_time**

The date and time the update was made.

**remark**

The comment for the update or type of operation performed.

**current\_status**

The current status of the project.

**old\_status**

The previous status of the project.

**history\_type**

The type of history record.

**action\_no**

The action number for this operation.

## **pcms\_workset\_history\_with\_attr - project update history including attributes**

This view gives update history including attribute updates for a project. The following fields are returned:

### **workset\_uid**

The UID for the project concerned.

### **action\_no**

The action number for this operation.

### **date\_time**

The date and time the update was made.

### **status**

The status of the baseline.

### **user\_name**

The name of the user who performed the update.

### **remark**

The comment for the update.

### **history\_type**

The type of history record.

### **attr\_no**

The attribute number affected by the operation (optional) - an integer between 1 and 220.

### **seq**

The (optional) attribute sequence number.

### **old\_attr\_val**

The old value of the attribute (if attribute tracking is enabled).

### **new\_attr\_val**

The new value of the attribute (if attribute tracking is enabled).

## **pcms\_workset\_info - project information**

This view gives information about projects. The following fields are returned:

### **workset\_uid**

The UID for the project concerned.

### **product\_id**

The name of the product to which the project belongs (uppercase).

### **workset\_name**

The name of the project (uppercase).

### **locked**

This is 'Y' or 'N' to indicate whether the project is locked or not.

### **description**

The description of the project as recorded in the database.

### **create\_date**

The date and time of creation for the project.

### **status**

The current lifecycle status of the project.

### **lifecycle\_id**

The name of the lifecycle defined for the project type.

### **lc\_seq**

The sequence number of the current lifecycle status within the normal lifecycle.

### **path\_control**

This is 'Y' or 'N' to indicate whether a request is required when refactoring changes are made to the project.

### **deployment\_method**

The method of deployment specified for this project. This can be either 'REQUEST' or 'BASELINE'. **stream**

This is 'Y' or 'N' to indicate if this project is a stream or a normal project.

## **pcms\_workset\_items - items in projects**

This view gives product items and the projects to which they belong. The following fields are returned:

### **item\_uid**

The UID of the product item concerned.

### **workset\_uid**

The UID of a project containing the product item.

### **filename**

The project filename for the product item concerned within the project identified by the workset\_uid field.

### **dir\_fullpath**

The full directory specification for the product item within its project, as held in the Dimensions database. It comprises zero-to-many path components with a single forward slash separating each component from the next.

### **unix\_directory**

The full directory specification for the product item within its project, in a UNIX-compatible format. Each component is separated from the next by a single '/' character, but there is no '/' character at the beginning or at the end.

### **item\_spec\_uid**

The uid for the product item as a whole (all revisions).

### **vms\_directory**

The operating specific translated version of DIR\_FULLPATH.

### **win\_directory**

The operating specific translated version of DIR\_FULLPATH.

## **pcms\_workset\_norm\_lc - normal lifecycle for projects**

This view provides information on the normal lifecycle for types of projects. The following fields are returned:

### **product\_id**

The name of the product to which the project belongs.

### **workset\_type**

The type of project concerned.

### **type\_uid**

The UID for the project type as defined in the relevant product.

### **state\_seq\_no**

The sequence number of the lifecycle state within the normal lifecycle.

### **status**

The name of a state on the normal lifecycle for projects of the type and product concerned.

### **lifecycle\_id**

The ID of the lifecycle.

## **pcms\_workset\_related\_chdocs - project request relationships**

This view provides information about requests that are related to a project. The following fields are returned:

### **from\_workset\_uid**

The UID of the project.

### **to\_ch\_uid**

The UID of the related request.

### **reltype\_name**

The type of relationship between the change document and the project. This may be **Affected**, **In Response To**, **Dependent**, or **Info**.

### **parent\_class\_name**

The name of the parent request class.

## pcms\_workset\_structure\_history - project refactoring changes

This view provides details on refactoring changes that have occurred only against a request in a Dimensions CM project or stream. The following fields are returned:

### change\_uid

The sequence number of a refactoring change. The sequence numbers grow in chronological order of the changes made in a project or stream.

### related\_change\_uid

The sequence number of the change that rolled this change back.

### request\_id

The ID of the request associated with this change.

### project\_spec

The project or stream in which this change was made.

### change\_type

The type of change being recorded. This may be ADDITION, CREATE, MODIFY, REMOVAL, MOVE, or RENAME.



**NOTE** For MODIFY, the item check in or check out must be done specifying a request for the modification to show up in the view.

### is\_dir

This returns Y if the changes relates to a directory, and N if not.

### item\_spec

The item specification for changes relating to items, or NULL for changes relating to directories.

### user\_name

The user who made the change.

### date\_time

The date and time when the structure change took place.

### old\_path

The old path to the directory or item.

### new\_path

The new path to the directory or item.

## **pcms\_full\_wset\_struct\_history - project refactoring changes**

This view provides details on refactoring changes that have occurred in a Dimensions CM project or stream (regardless of whether a request was used or not). The following fields are returned:

### **change\_uid**

The sequence number of a refactoring change. The sequence numbers grow in chronological order of the changes made in a project or stream.

### **project\_uid**

The unique identifier of the CM project or stream.

### **project\_spec**

The project or stream in which this change was made.

### **change\_type**

The type of change being recorded. This may be ADDITION, CREATE, MODIFY, REMOVAL, MOVE, or RENAME.

### **is\_dir**

This returns Y if the change relates to a directory, and N if not.

### **obj\_spec\_uid**

The item specification UID for changes relating to items, or -1 for changes relating to directories. This is of type NUMBER.

### **item\_spec**

The item specification for changes relating to items, or NULL for changes relating to directories.

### **user\_name**

The user who made the change.

### **date\_time**

The date and time when the change took place.

### **old\_path**

The old path to the directory or item.

### **new\_path**

The new path to the directory or item.



## Use of the Views

These views are designed to support reporting functions, and full versatility may be attained by judicious joins between views and suitable 'where' clauses. The following examples show suitable SQL statements for various reports.

### List of Open Requests

The following query returns all open requests of types CR and PR on the product PAYPHONE, giving a few basic fields. Only one attribute field, the title, is retrieved, and this is a single-valued attribute. No multi-valued attributes are required.

```
select ch_doc_type, doc_seq, ch_doc_id, status, create_date,
       originator, title from pcms_chdoc_data
```

where:

```
product_id = 'PAYPHONE'
  and ch_doc_type in ('CR', 'PR')
  and seq = 1
  order by ch_doc_type, doc_seq;
```

The clause "and seq = 1" ensures that only the first row of attributes is returned: this is appropriate when no multi-valued attributes are required.

The "order by" clause places the requests in numerical order, rather than a lexical order as happens when ordering on the field ch\_doc\_id.

### Request Selected on Multi-Valued Attributes

The following query returns all open requests of type PR on the product PAYPHONE, giving a few basic fields. One single-valued attribute field, the title, is retrieved, and two multi-valued attributes, affected\_release and site\_involved. These are to have "4.2" and "Swindon" respectively among their values. The title is to refer to "Dialling".

```
select ch_doc_type, doc_seq, ch_doc_id, status, title,
       affected_release, site_involved
  from pcms_chdoc_data
 where product_id = 'PAYPHONE'
       and ch_doc_type = 'PR'
       and ch_uid in
           (select ch_uid from pcms_chdoc_data
            where product_id = 'PAYPHONE'
              and ch_doc_type = 'PR'
              and upper(title) like %DIALLING%
            intersect
            select ch_uid from pcms_chdoc_data
            where product_id = 'PAYPHONE'
              and ch_doc_type = 'PR'
              and affected_release = '4.2'
            intersect
            select ch_uid from pcms_chdoc_data
            where product_id = 'PAYPHONE'
              and ch_doc_type = 'PR'
              and upper(site_involved) = 'SWINDON')
 order by 1, 2;
```

## Requests Pending for a User and Related Parts

There is often a requirement for reports of the type that list all requests pending for a given user, with their related design parts. This can be done by means of nested queries, as shown in the next example. In some cases, it is possible to use a single query, as outlined below.

This query returns the user's identity, a few details including the title for each pending request, and the specification of related design parts.

```
select distinct a.user_name, role, next_status,
               b.product_id, ch_doc_type, doc_seq, ch_doc_id,
               b.status, title,
               reltype_name, parent_class_name,
               d.product_id, part_id, variant, pcs
from pcms_user_pending_chdoc a, pcms_chdoc_data b,
     pcms_chdoc_related_parts c, pcms_part_data d
where a.ch_uid = b.ch_uid
     and a.ch_uid = c.from_ch_uid
     and c.to_part_uid = d.part_uid
     and b.seq = 1
     and a.user_name = 'SMITH'
```

This query returns one row per pending request and related part. It is not suitable for use in connection with multi-valued attributes, since the rows returned for any pending request will be the Cartesian product of the attribute with most values and the number of related parts.

## Nested Queries

This example is a report listing all requests of type PR on the product PAYPHONE, and for each request listing all related design parts and all related requests. This is done by three queries: one outer query enclosing two inner queries.

The outer query is very similar to the first example:

```
select ch_uid, ch_doc_type, doc_seq, ch_doc_id, status,
       create_date, originator, title
from pcms_chdoc_data
where product_id = 'PAYPHONE'
     and ch_doc_type = 'PR'
     and seq = 1
order by 2, 3;
```

This query is arranged to return ch\_uid into a host variable :outer\_ch\_uid (the syntax varies between report writers).

For each request returned by the outer query, two inner queries are performed. The first of these gets the related design parts:

```
select product_id, part_id, variant, pcs, reltype_name,
       parent_class_name
from pcms_chdoc_related_parts a, pcms_part_data b
where a.from_ch_uid = :outer_ch_uid
     and a.to_part_uid = b.part_uid
order by 1, 2, 3, 4;
```

The second inner query returns the related requests:

```

select product_id, ch_doc_type, doc_seq, ch_doc_id, status, title
  from pcms_chdoc_related_chdoc a, pcms_chdoc_data b
 where a.from_ch_uid = :outer_ch_uid
        and a.to_ch_uid = b.ch_uid
        and seq = 1
 order by 1, 2, 3;

```

## Hierarchical Queries

Reports of the "family tree" type require the use of the CONNECT BY construct of the SQL language. Queries incorporating this construct are known as hierarchical queries, and (for technical reasons) can only be done using certain views. These views are identified in their preambles in this document.

This kind of report is often suited to the use of nested queries. However, an example of a hierarchical report performed by a single query is to list all the design parts related by breakdown to a given design part and its descendants. Each line of the report gives the parent and child design parts for one relationship in the tree. The query is:

```

select substr (p.product_id, 1, 10), substr (p.part_id, 1, 25),
       substr (p.variant, 1, 4),
       substr (p.pcs, 1, 3), substr (c.product_id, 1, 10),
       substr (c.part_id, 1, 25),
       substr (c.variant, 1, 4), substr (c.pcs, 1, 3)
  from pcms_part_data p, pcms_part_data c
 where (p.part_uid, c.part_uid) in
        (select parent_part_uid, child_part_uid
         from pcms_parts_structure
         where type_of_rel = 1
         connect by parent_part_uid = prior child_part_uid
         start with parent_part_uid in
         (select part_uid from pcms_part_data
          where product_id = 'PAYPHONE'
          and part_id = 'RINGER'
          and variant = 'AAAA'
          and pcs = '1'));

```

The SUBSTR function is used to limit the displayed length of the various fields. The database allows much longer data in some of these columns, and users should have regard both to the lengths of their data, and to report formatting considerations.

