HP Project and Portfolio Management Center

Software Version: 7.1

Reporting Meta Layer Guide and Reference

Document Release Date: April 2007
Software Release Date: March 2007

Legal Notices

This document, and the accompanying software and other documentation, is protected by U.S. and international copyright laws, and may be used only in accordance with the accompanying license agreement. Features of the software, and of other products and services of Mercury Interactive Corporation, may be covered by one or more of the following patents: United States: 5,511,185; 5,657,438; 5,701,139; 5,870,559; 5,958,008; 5,974,572; 6,137,782; 6,138,157; 6,144,962; 6,205,122; 6,237,006; 6,341,310; 6,360,332, 6,449,739; 6,470,383; 6,477,483; 6,549,944; 6,560,564; 6,564,342; 6,587,969; 6,631,408; 6,631,411; 6,633,912; 6,694,288; 6,738,813; 6,738,933; 6,754,701; 6,792,460 and 6,810,494. Australia: 763468 and 762554. Other patents pending. All rights reserved.

U.S. GOVERNMENT RESTRICTED RIGHTS. This Software Documentation is a “commercial item” as defined at 48 C.F.R. 2.101 (October 1995). In accordance with 48 C.F.R. 12.212 (October 1995), 48 C.F.R. 27.401 through 27.404 and 52.227-14 (June 1987, as amended) and 48 C.F.R. 227.7201 through 227.7204 (June 1995), and any similar provisions in the supplements to Title 48 of the C.F.R. (the “Federal Acquisition Regulation”) of other entities of the U.S. Government, as applicable, all U.S. Government users acquire and may use this Documentation only in accordance with the restricted rights set forth in the license agreement applicable to the Computer Software to which this Documentation relates.

Mercury, Mercury Interactive, the Mercury logo, the Mercury Interactive logo, LoadRunner, WinRunner, SiteScope and TestDirector are trademarks of Mercury Interactive Corporation and may be registered in certain jurisdictions. The absence of a trademark from this list does not constitute a waiver of Mercury's intellectual property rights concerning that trademark.

All other company, brand and product names may be trademarks or registered trademarks of their respective holders. Mercury disclaims any responsibility for specifying which marks are owned by which companies or which organizations.

Intel®, Intel® Itanium®, Intel® Xeon™, and Pentium® are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Java™ is a U.S. trademark of Sun Microsystems, Inc.

Microsoft®, Windows®, and Windows® XP are U.S. registered trademarks of Microsoft Corporation.

Oracle® is a registered U.S. trademark of Oracle Corporation, Redwood City, California.

UNIX® is a registered trademark of The Open Group.

Mercury provides links to external third-party Web sites to help you find supplemental information. Site content and availability may change without notice. Mercury makes no representations or warranties whatsoever as to site content or availability.

Documentation Updates

This manual's title page contains the following identifying information:

- Software version number, which indicates the software version
- Document release date, which changes each time the document is updated
- Software release date, which indicates the release date of this version of the software

To check for recent updates, or to verify that you are using the most recent edition of a document, go to: http://ovweb.external.hp.com/lpe/doc_serv/.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HP sales representative for details.

Support

Mercury Product Support

You can obtain support information for products formerly produced by Mercury as follows:

- If you work with an HP Software Services Integrator (SVI) partner (www.hp.com/managementsoftware/svi_partner_list), contact your SVI agent.
- If you have an active HP Software support contract, visit the HP Software Support site and use the Self-Solve Knowledge Search to find answers to technical questions.
- For the latest information about support processes and tools available for products formerly produced by Mercury, we encourage you to visit the HP-Mercury Software Support web site at: support.mercury.com.
- Contact your HP Sales Representative if you have additional questions.

HP Software Support

You can visit the HP Software Support web site at www.hp.com/managementsoftware/services.

HP Software online support provides an efficient way to access interactive technical support tools. As a valued support customer, you can benefit by using the support site to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HP support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract. To find more information about access levels, go to: www.hp.com/managementsoftware/access_level.

To register for an HP Passport ID, go to: www.managementsoftware.hp.com/passport-registration.html.

Table of Contents

List of Tables

1 Getting Started with the Reporting Meta Layer

In This Chapter:

Introduction to the Reporting Meta Layer

The Reporting Meta Layer (RML) for HP Project and Portfolio Management Center (PPM Center) allows customers to use third-party reporting software to define custom reports. Any third-party reporting tool capable of running SQL queries on an Oracle® database can work with PPM Center reporting capabilities by:

- Using the RML schema in the PPM Center database as its data source
- Building reports using the standard capabilities of the PPM Center reporting system

Examples of third-party software being used with PPM Center include Actuate, Brio, Cognos, Crystal Reports, and Oracle Reports.

Target users of the RML are report designers and administrators responsible for creating business reports about HP Demand Management and HP Deployment Management application usage. HP assumes that these users have a basic understanding of relational database concepts, Oracle technologies, and PPM Center applications. However, the RML makes it possible for these users to report on HP Demand Management and HP Deployment Management data without understanding the technical complexities of the underlying data model.

The RML Schema

The RML is a schema in the PPM Center Oracle database that has privileges to view tables in the database schema. However, the RML:

- Resides in a separate layer from the standard PPM Center database schema
- Has read-only access to PPM Center data, so that third-party reporting tools using RML capabilities cannot alter or corrupt the PPM Center database

In order to prevent a third-party report from exposing information to people who lack the proper authorization, security views (which can be referenced by any other view) are included in the Reporting Meta Layer.

RML database views, which are created through templates, read and interpret data from the PPM Center database. Views are created through compilation, in which a view template is read, custom information to be included is calculated, and the final view that resides in the Reporting Meta Layer is generated.

The RML stays up to date with the current state of HP Demand Management and HP Deployment Management data through synchronization. During the synchronization procedure each RML view template is parsed and used as a basis for generating an updated view or set of views in the RML schema. User-specified options control which views are compiled during synchronization.

➤ HP does not recommend changing or dropping RML templates.

The RML Views

RML views are representations of logical PPM Center business or functional entities for HP Demand Management and HP Deployment Management. RML views are presented as Oracle views.

‼ In prior versions of PPM Center, the RML views supported additional applications such as HP Resource Management and HP Project Management. With version 7.1, RML support is limited to HP Demand Management and HP Deployment Management.

Reporting Meta Layer Views

The RML views provide visibility into HP Demand Management and HP Deployment Management application areas of the PPM Center product as described in the following sections. Detailed information can be found in the appendix as referenced in each section.

Cross-Product Views

Cross-product Views on page 26 relate information across HP Demand Management and HP Deployment Management application areas. For example, MWFL_STEP_ACTIVITIES shows statistics about workflow step completion across applications.

Deployment Management Views

Deployment Management Views on page 29 provide information specific to HP Deployment Management. For example, MPKGL_OBJ_TYPE_DEPLOYMENT_D provides summary information for package deployment activity, broken down by object type and calendar day. MPKGL_PACKAGE_LINES provides information about package lines including global package line user data fields.

Demand Management Views

Demand Management Views on page 77 provide information specific to HP Demand Management. For example, MREQ_OPENED_CLOSED_BY_TYPE_D provides summary information for request submission and completion activity, broken down by request type and by calendar day. MREQ-REFERENCES provide information about references related to HP Demand Management requests.

Other Views

Other Views on page 115 provide information about HP Demand Management and HP Deployment Management entities like workflows and security groups. For example, MWFL_STEP_SECURITY_USERS lists all users with authority to act on a given workflow step through static security group or user linkage, as defined in the workflow step window in the Workflow workbench.

These views are useful to report designers. KRML_CALENDAR_DAYS is a utility table that contains daily date records.

For reporting needs not met by the view in the preceding categories, the RML provides entity-specific views that map to the data shown in the user interface. For example, each request type in HP Demand Management has a unique view in the RML that presents both request detail fields and user data fields. This allows report writers to devise reports that implement specific customer-oriented business logic contained in customer-defined fields.

2 Working with the Reporting Meta Layer

In This Chapter:

Setting Up the Reporting Meta Layer

The following sections describe the basic structure of the RML, as well as the behavior and maintenance of its views.

RML views are essentially SQL statements that return specific, useful data from the PPM Center database, providing direct mapping to the business entities defined in PPM Center applications.

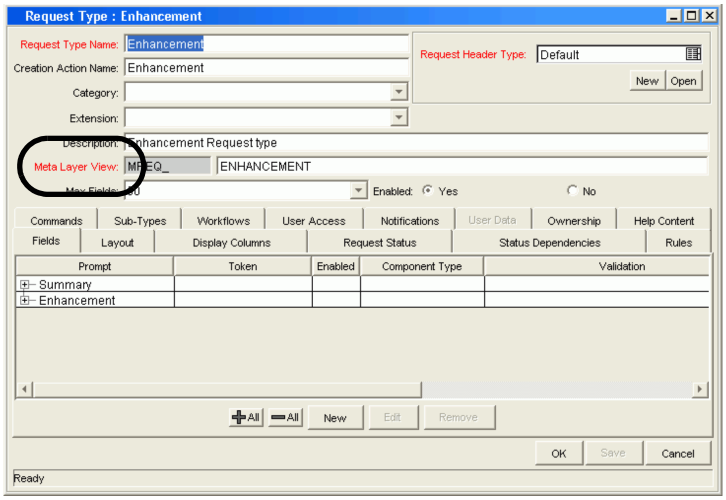
Any third-party reporting software capable of connecting to an Oracle database and running query statements in SQL can use the Reporting Meta Layer. RML views are used by including them in query statements.

Synchronizing the Reporting Meta Layer

PPM Center transactional entities like request types and object types have their own RML views. Each view is defined by a view template that dictates the view's construction. For example, templates contain markers for entities containing custom fields. When custom fields are encountered during view compilation, the template puts them into the view, using their tokens as column names.

Every time a new entity like a request type is created, it must be given a corresponding Meta Layer view name (see *Figure 2-1*). Each view must have a unique name that cannot be duplicated in the system.

Figure 2-1. Recording Meta Layer view names in entity definitions



Request Type : Enhancement

Request Type Name: Enhancement

Creation Action Name: Enhancement

Category:

Extension:

Request Header Type: Default

Description: Enhancement Request type

Meta Layer View: MREQ_ ENHANCEMENT

Max Fields: 50

Enabled: ● Yes ○ No

Prompt	Token	Enabled	Component Type	Validation
⊞ Summary				
⊞ Enhancement				

Ready

➤ Since RML views are essentially Oracle database views, they are named according to Oracle convention. Consequently RML view names cannot be more than 20 characters in length and must be composed of A–Z or 0–9 only, with _ for spaces.

Do not use Oracle–reserved words for field tokens, since they are being used as view columns.

User data fields are also incorporated into many RML views. The types of user data that could be present in one or more RML views includes the following:

- **Package user data:**
  - Contact user data
  - Package line user data
  - Project environment data
  - Request type user data
  - Security group user data
  - Workflow step user data
  - Workflow user data
- **Global and context-sensitive user data:** Global user data only

As part of routine PPM Center configuration, users can update custom fields, entity names, and other configuration information at any time. Every change has the potential to render existing RML views obsolete, invalidating reports based on these obsolete views. For configuration changes to be reflected in the Reporting Meta Layer, it must be synchronized to keep RML views current with PPM Center configurations.

The Synchronization Procedure

To synchronize the Reporting Meta Layer:

1. In the PPM Center standard interface, select **Reports > Submit New Report.**

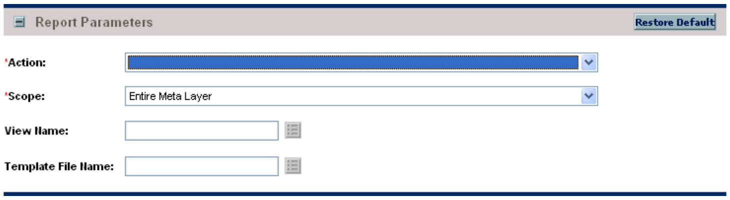
   The Submit New Report page opens.

2. In the bottom section (Select Report by Category), select the Administrative category.

3. From the list of Administrative reports, select Synchronize Meta Layer.

   The Submit Report: Synchronize Meta Layer window opens, as shown in *Figure 2-2*.

Figure 2-2. Synchronizing the Meta Layer



*Action:

*Scope: Entire Meta Layer

View Name:

Template File Name:

4. Select the **Action** you want:

   - To simulate a synchronization, select **Assess.**

     This generates a synchronization report listing the updates that would be made to the Reporting Meta Layer if the synchronization were implemented, allowing the impact of any changes to be assessed.

   - To perform the actual synchronization, select **Synchronize.**

     This compiles all views from existing view templates and generates a report of the updates made, subject to the scope specified (see step 5).

   - To remove views in the Reporting Meta Layer (including the entire RML itself) that are no longer needed, select **Drop.**

5. Select the **Scope** you want:

    - To perform the selected action on the entire Reporting Meta Layer, select **Entire Meta Layer.**
    - To activate the View Name auto-complete list, select **Specific View.**

      You must select a view to perform the selected action.

    - To activate the Template File Name auto-complete list, select **Specific Template.**

      You must select a view template to perform the selected action.

6. To run the report, click **Submit.**

   The result depends on your choices. If you chose **Synchronize** and **Entire Meta Layer,** running the reports synchronizes the entire meta layer, and the RML view is current with PPM Center configurations made since the last synchronization.

Synchronization Messages

Table 2-1 lists and describes the possible RML synchronization informational messages, user error messages, and internal error messages.

Table 2-1. RML synchronization messages (page 1 of 7)

Message ID	Display Text	Cause	Action/Solution
Informational Messages			
KNTA-10504	No description available for this view. View description is not provided in template <*TemplateName*>.	The template file for the specified RML view does not contain a description. (For HP-supplied templates, this should never occur.)	RML view templates are stored on the PPM Server file system in the $knta_home/rml/templates directory. Edit the template file with a text editor and add the text "[VIEW_DESCRIPTION =]" to the template header.
KNTA-10512	Parameter set context obsolete: ID = <*ContextID*>.	Either a request type, object type, or context-sensitive user data field set has been deleted.	The corresponding view will be dropped from the RML.
KNTA-10513	New template <*TemplateName*>.	The specified new RML view template was found in the $knta_home/rml/templates directory on the PPM Server.	New RML view(s) will be generated based on this template.
KNTA-10514	Custom fields updated	Modifications to the custom field definitions for the corresponding request type, object type, or user data field set were made since the last RML synchronization event.	The associated RML view(s) will be updated with the current field information.
KNTA-10515	View name changed from <*OldViewName*> to <*NewViewName*>.	The RML view name was changed from <*OldViewName*> to <*NewViewName*> since the last synchronization.	The corresponding view will be recreated with the new name.

Table 2-1. RML synchronization messages (page 2 of 7)

Message ID	Display Text	Cause	Action/Solution
KNTA-10516	New parameter set context (ID = <*ContextID*>) for the template.	Since the last synchronization, a new request type, object type, or context-sensitive user data field set was created.	A corresponding view will be generated in the RML.
KNTA-10517	Drop view <*ViewName*> request initiated by user	User has requested to drop the specified view from the RML.	View will be dropped.
KNTA-10518	Dependent template <*TemplateName*> dropped by user.	User has requested to drop the template from the RML.	All RML views generated from the template will be dropped.
KNTA-10550	Drop operation complete.	The operation requested by the user to drop specific RML views has been completed.	View(s) will be dropped.
KNTA-10551	Dependent context sets changed for template <*TemplateName*>.	Several types of configuration events since the last RML synchronization could lead to this message: a new request type, object type, or context-sensitive user data field set was created; a field definition was added, modified, or removed in an existing request type, object type, or user data field set; a request type, object type, or context-sensitive user data field set was deleted.	Impacted views will be dropped and re-created, as applicable.
KNTA-10562	New view for static context template <*TemplateName*>.	The RML view for the specified template did not exist prior to this synchronization.	The view will be created during the RML synchronization.

Table 2-1. RML synchronization messages (page 3 of 7)

Message ID	Display Text	Cause	Action/Solution
KNTA-10591	Existing view name is restored: <*ExistingView Name*>	The operation requested by the user to restore specific RML view has been completed.	View will be restored.
KNTA-10620	Another assessment or synchronization process is already running. There can be only one assessment or synchronization process running on a given database.	Another RML assessment or synchronization process is running at the same time this one was submitted.	Try again later.
KNTA-10622	No Change Detected.	The view or template specified for synchronization was not updated, since it is already up to date.	No further action is needed.
KNTA-10698	No views will be generated as no driving contexts were resolved for template:	The specified template will generate a view for each configuration instance (request type, object type, or user data field set), but no configurations were found for the template.	Verify that the driving contexts were correct. If they were, no further action is needed. Otherwise, provide the revised driving contexts.
User Error Messages			
KNTA-10505	Could not determine context type for Meta Layer view template file <*TemplateName*>.	None of [VIEW_NAME_PREFIX] and [STATIC_VIEW_NAME] is specified in the template.	Specify [STATIC_VIEW_NAME] for a template with static context or specify [VIEW_NAME_PREFIX] for a template with dynamic context.
KNTA-10506	Could not determine Meta Layer view name. <*ViewName*>	Either the system could not find the corresponding parameter set or the Meta Layer view name root is not specified.	Make sure the “Meta Layer View” field is completed for the corresponding Request Type, Object Type, or User Data.

Table 2-1. RML synchronization messages (page 4 of 7)

Message ID	Display Text	Cause	Action/Solution
KNTA-10507	Fail to create comment for “{0}”. {1} DDL statement: {2}.	The system grant is granted to the base PPM Center schema at the time of installation and upgrade.	Be sure to store it if it has been accidentally dropped.
KNTA-10510	Create or replace view...” statement failed	The SQL statement to create an RML view failed.	Examine the Oracle error. There may be several reasons why the SQL statement would fail. Perhaps the original SQL specified in the RML template has syntax errors, or some parameters in the SQL were either missing or resulted in an invalid SQL statement.
KNTA-10511	Oracle error when dropping view <*ViewName*>.	The specified view could not be dropped.	Examine the Oracle error to determine why the view could not be dropped, and consult the DBA for the PPM Center database to help fix the problem.
KNTA-10543	The Meta Layer view name <*ViewName*> you have specified for <*EntityType1*> <*EntityID1*> conflicts with the name of the existing Meta Layer view for <*EntityType2*> <*EntityID2*>. Please choose another name for the view.	The same RML view name has been specified for multiple request types, object types, or context-sensitive user data field sets.	Update one of the entities with the duplicate RML view name and change it to a unique name.
KNTA-10545	Cannot change the view name for built-in static context template <*TemplateName*> Please restore the original view name <*ViewName*>	The static view names defined in RML view templates that come with PPM Center cannot be changed.	Restore the original static view name for the name template, as instructed in the message.

Table 2-1. RML synchronization messages (page 5 of 7)

Message ID	Display Text	Cause	Action/Solution
KNTA-10546	Driving context changed for template <*TemplateName*>. It is not allowed to change driving context set for PPM Center built-in template. Please restore original template. Original driving context set is <*ContextID*>.	Some unsupported customization has been done for the name template.	Be sure to undo those changes.
KNTA-10547	Cannot change view name prefix to <*NewViewPrefix*> for built-in template <*TemplateName*>. Please restore original [VIEW_NAME_PREFIX=<*OldViewPrefix*>].	The Token [VIEW_NAME_PREFIX=…] must be defined for specific entity-based RML view templates. The definition is not allowed to change for those view templates that comes with PPM Center.	To resolve this problem, restore it to the original as instructed in the message.
KNTA-10548	Driving context changed from <*OldContext*> to <*NewContext*> for template <*TemplateName*>. Cannot continue assessment. To change driving context set for the template, you must drop the template first and then re-assess the template.	For specific entity-based RML view templates, [DRIVING_PARAMETER_SET = parameter_set_id] is specified.	Perform a drop operation on the template and synchronize again to make the change take effect.
KNTA-10549	Cannot find view <*ViewName*> to drop; no action taken.	The RML view to drop does not exist.	No action was taken during RML synchronization; the view was previously dropped.

Table 2-1. RML synchronization messages (page 6 of 7)

Message ID	Display Text	Cause	Action/Solution
KNTA-10552	Dependent context sets changed for template <*TemplateName*>. Could not continue process. To make this kind of change, the template must be dropped and re-assessed.	All RML templates that have custom data fields in them have associated contexts. Any changes to the dependent contexts invalidate all views and the templates that have previously been assessed/ synchronized.	Perform a drop operation on the name template and synchronize the template again.
KNTA-10629	Meta Layer View not defined for <*EntityType*> <*EntityID*>.	All configurations that drive a dynamic RML view (request types, object types, and context-sensitive user data field sets) must have an RML view name specified. This one did not.	Specify an RML view name for the corresponding configuration.
KNTA-10675	Value <*ViewName*> is too long for [VIEW_ NAME] in template <*TemplateName*>. Value for [VIEW_ NAME] must be no more than 30 characters.	The length of the supplied RML view name was too long.	Shorten the value of the RML view name to less than 30 characters.
KNTA-10678	Value <*ViewName*> is too long for [VIEW_ NAME_PREFIX] in template <*TemplateName*>. Value for [VIEW_ NAME_PREFIX] must be no more than 10 characters.	The view name prefix cannot be more than 10 characters long.	Change the length of the view name prefix.
KNTA-10680	Could not resolve driving parameter set context id in template <*TemplateName*>. Please verify the template.	The RML system could not resolve the specific entity on which the name template is based.	Contact HP-Mercury support (support.mercury.com) to report the problem.

Table 2-1. RML synchronization messages (page 7 of 7)

Message ID	Display Text	Cause	Action/Solution
KNTA-10681	The Meta Layer view name <*ViewName*> you have specified for <*EntityType*> <*EntityID*> conflicts with the name of another Meta Layer view that either failed a previous drop attempt, or is currently pending to be dropped. Please make sure the existing view <*ViewName*> has been successfully dropped by a Meta Layer Synchronization report before trying to create the new one.	The RML synchronization is attempting to create a new view with the same name as an existing view that should have been dropped.	Please make sure the existing view <*ViewName*> has been successfully dropped before trying to create the new one.
Internal Error Messages			
KNTA-10509	View ID is not returned after assessment	An unexpected error occurred.	Contact HP-Mercury support (support.mercury.com) to report the problem.

A Reporting Meta Layer

In This Appendix:

- *MWFL_WORKFLOW_STEPS*
- *KCRT_PARTICIPANT_CHECK_V*
- *KDLV_PARTICIPANT_CHECK_V*
- *KRML_CALENDAR_DAYS and KRML_CALENDAR_MONTHS*

Cross-product Views

Cross-product views relate information across PPM Center products. Each view is described in the sections that follow. After each description is a table that includes the column names and information for that view.

For example, MWFL_STEP_ACTIVITIES shows statistics about workflow step completion across applications.

MWFL_STEP_ACTIVITIES

This view contains activity statistics for all workflow steps, including subworkflows.

Usage

For any given workflow or workflow step, MWFL_STEP_ACTIVITIES can be used to get a quick snapshot of aggregate system activity. It is provided as a general reference for gathering data that is not covered by other product-specific statistical views. The internal ID columns for workflow and workflow step (WORKFLOW_ID and WORKFLOW_STEP_ID, respectively) can be used to join this view to other product action or workflow-related views to gather additional information about the records contained therein.

This view can also be used to flag step duration bottlenecks by looking at step completion times (AVG_TIME_TO_COMPLETE and AVG_TIME_OPEN), or other exceptions like spikes in the number of cancelled workflow steps for a point in time.

Sample Query

Suppose a report needs to contain summary information for the number of errors for step 2 in the FIN dev-test-prod workflow, broken down by month. The calendar table KRML_CALENDAR_MONTHS can be used to provide the month-by-month breakdown to join with the ACTIVITY_DATE column in this view:

```
SELECT m.calendar_month MONTH,
       sum(sa.error)     NUM_ERRORS
FROM   krml_calendar_months m,
       mwfl_step_activities sa
WHERE  sa.workflow = 'FIN dev-test-prod'
AND    sa.workflow_step_number = 2
AND    sa.activity_date >= m.start_date
AND    sa.activity_date < m.end_date
GROUP BY m.calendar_month
ORDER BY 1;
```

Results

```
    MONTH    NUM_ERRORS
---------                                          -------------
01-APR-01            16
01-MAY-01             4
01-JUN-01             0
01-AUG-01             0
01-SEP-01             1
```

Table A-1. MWFL_STEP_ACTIVITIES view column descriptions (page 1 of 2)

Column Name	Data Type	Description
WORKFLOW	VARCHAR2(80)	Name of workflow being analyzed.
WORKFLOW_STEP	VARCHAR2(80)	Name of workflow step to analyze.
WORKFLOW_STEP_NUMBER	NUMBER	Workflow step number.
ACTIVITY_DATE	DATE	Date of activity.
AVG_TIME_TO_COMPLETE	NUMBER	Average number of days to complete a step activity.
AVG_TIME_OPEN	NUMBER	Average number of days steps have remained open.
ELIGIBLE	NUMBER	Number of eligible steps.
COMPLETE	NUMBER	Number of completed steps.
ERROR	NUMBER	Number of steps having errors.
IN_PROGRESS	NUMBER	Number of in-progress steps.
CANCELLED	NUMBER	Number of cancelled steps.

Table A-1. MWFL_STEP_ACTIVITIES view column descriptions (page 2 of 2)

Column Name	Data Type	Description
PENDING	NUMBER	Number of steps pending some other event.
SCHEDULED	NUMBER	Number of scheduled execution steps.
WORKFLOW_ID	NUMBER	Internal ID for the workflow.
WORKFLOW_STEP_ID	NUMBER	Internal ID for the workflow step.

Deployment Management Views

Deployment Management views provide information specific to HP Deployment Management. For example, MPKGL_OBJ_TYPE_DEPLOYMENT_D provides summary information for package deployment activity, broken down by object type and calendar day. MPKGL_PACKAGE_LINES provides information about package lines including global package line user data fields.

MPKG_DEPLOYMENT_DETAILS

Provides information on the details of object deployments to environments. MPKG_DEPLOYMENT_DETAILS has a record for each deployment.

Usage

This view is based on object deployment history stored in the environment contents tables. As a result, it includes accurate records for deployments even when the destination environment specified on the migration workflow step was overridden during object type command processing.

Sample Query

The following example reports on all objects deployed to the MFG Prod environment in the last day:

```
SELECT package_number  package,
       line_number     line,
       object_type     object,
       object_name     name,
       object_revision version
FROM   mpkg_deployment_details
WHERE  destination_environment = 'MFG Prod'
AND    deployment_date > sysdate - 1;
```

Results

```
package  line object             name               version
-------- ---- ------------------ ------------------ -------
30023       3 Migrate SQL file   add_user.sql       3.12
30023       5 Migrate SQL file   create_links.sql   8
30121       1 File Migration     runProcess.sh      2.7
30122       1 FIle Migration     runProcess.sh      2.9
…
```

Table A-2. MPKG_DEPLOYMENT_DETAILS view column descriptions (page 1 of 2)

Column Name	Data Type	Description
PACKAGE_NUMBER	VARCHAR2(40)	Package number.
PACKAGE_DESCRIPTION	VARCHAR2(240)	Package description.
LINE_NUMBER	NUMBER	Line number of the deployed package.
OBJECT_TYPE	VARCHAR2(80)	Object type used by this package line.
OBJECT_NAME	VARCHAR2(300)	Value of the Object Name field.
OBJECT_REVISION	VARCHAR2(300)	Value of the Object Revision field.
APPLICATION_CODE	VARCHAR2(30)	Application code of the package line.
SOURCE_ENVIRONMENT	VARCHAR2(80)	Source of deployed object.
DESTINATION_ENVIRONMENT	VARCHAR2(80)	Destination of deployed object.
WORKFLOW	VARCHAR2(80)	Name of the package workflow.
WORKFLOW_STEP	VARCHAR2(80)	Deployment step.
DEPLOYMENT_DATE	DATE	Date package line was deployed.
DEPLOYED_BY_USERNAME	VARCHAR2(200)	Username of PPM Center user that deployed the package line object.
DEPLOYED_BY_FULL_NAME	VARCHAR2(61)	Full name of PPM Center user that deployed the package line object.
PACKAGE_ID	NUMBER	Internal ID of this package.
PACKAGE_LINE_ID	NUMBER	Internal ID of this package line.
WORKFLOW_ID	NUMBER	Internal ID of the package workflow.

Table A-2. MPKG_DEPLOYMENT_DETAILS view column descriptions (page 2 of 2)

Column Name	Data Type	Description
WORKFLOW_STEP_ID	NUMBER	Internal ID of the workflow step.
SOURCE_ ENVIRONMENT_ID	NUMBER	Internal ID of the source environment.
DEST_ENVIRONMENT_ ID	NUMBER	Internal ID of the destination environment.

MPKG_NOTES

Provides access to the notes for all packages in Deployment Management.

Usage

Notes are stored in an Oracle LONG database column; to prevent an overload of information this is presented in a separate Meta Layer view, making it less likely to design a report that inadvertently returns too much data.

To query package notes, join this view with the MPKG_ALL_PACKAGES view.

Sample Query

To retrieve a list of the notes for all open packages being processed through the FIN dev -> prod workflow, and that have Critical priority, use the following logic in an SQL statement:

```
SELECT p.package_number PKG_NUM,
       n.NOTE_DATA NOTES
FROM   mpkg_packages p,
       mpkg_notes n
WHERE  p.priority = 'Critical'
AND    p.workflow = 'FIN dev -> prod'
AND    p.package_id = n.package_id;
```

Table A-3. MPKG_NOTES view column descriptions (page 1 of 2)

Column Name	Data Type	Description
PACKAGE_ID	NUMBER	Package internal identifier.
PACKAGE_NUMBER	VARCHAR2(40)	Package number.
AUTHOR_ID	NUMBER	Author of the note (a PPM Center user).
AUTHOR_USERNAME	VARCHAR2(200)	Author of the note (a PPM Center user).
AUTHOR_FULL_NAME	VARCHAR2(80)	Author of the note (a PPM Center user).
AUTHORED DATE	DATE(7)	Date the note was authored.
NOTE_CONTEXT_ VALUE	VARCHAR2(30)	Any context information for the note entry.
NOTE_CONTEXT_ VISIBLE_VALUE	VARCHAR2(80)	Any visible context information for the note entry.
NOTE_TYPE_CODE	VARCHAR2(17)	User-entered or system-generated note.

Table A-3. MPKG_NOTES view column descriptions (page 2 of 2)

Column Name	Data Type	Description
USER_NOTE_DATA	CLOB(4000)	User-entered note.
NOTE_DATA	VARCHAR2(4000)	Note contents.
OLD_COLUMN_VALUE	VARCHAR2(1800)	Old value for a field-change note entry.
OLD_VISIBLE_ COLUMN_VALUE	VARCHAR2(1800)	Old visible value for a field-change note entry.
NEW_COLUMN_VALUE	VARCHAR2(1800)	New value for a field-change note entry.
NEW_VISIBLE_ COLUMN_VALUE	VARCHAR2(1800)	New visible value for a field-change note entry.
COLUMN_PROMPT	VARCHAR2(80)	Column prompt.

MPKG_PACKAGES

The most general view into package transaction data. A blind query (that is, `SELECT * FROM mpkg_packages`) returns one row for each package present in the system, including closed packages.

Usage

The view columns map to package header fields like Priority, Package Group, and Assigned-to User. There are also columns for the package status and the dates on which it was submitted, closed, and/or cancelled. Since global package user data fields are present on all packages, there is also a view column for each global package user data field that is defined.

The column name for each global package user data field is the same as the token name for that field.

Context-sensitive package user data sets have their own views. See MPKG_UD_<*Context Value*>.)

Use this view when writing a report to present general package header information. For information about individual package lines, use the other views that provide line detail.

The MPKGL_PACKAGE_LINES view can be used to query general package line data, including package line user data fields. If it is necessary to report on the activity of specific object types, the set of object type-specific views is more appropriate. See MPKGL_<*Object Type Name*>.

Sample Query 1

For example, to get a picture of the number of open packages in the system and to whom they are assigned:

```
SELECT assigned_to_username ASSIGNED_USER,
       COUNT(*) NUM_OPEN
FROM   mpkg_packages
WHERE  close_date IS NULL
AND    cancel_date IS NULL
AND    submission_date IS NOT NULL
GROUP BY assigned_to_username
ORDER BY 1;
```

Results 1

```
ASSIGNED_USER                  NUM_OPEN
------------------------------                        ---------
…
rfrazier                                     13
rjeffries                                     1
rjones                                       28
rnelson                                       9
rsmith                                        3
…
```

Sample Query 2

For another example, consider the case where a global package user data field has been defined to capture the username of a backup user responsible for each package. The token name for this field is BACKUP_USERNAME. In this view there would be a column named BACKUP_USERNAME:

```
SQL> desc mpkg_packages
```

Results 2

```
Name                            Null?    Type
------------------------------- -------- ----
PACKAGE_NUMBER                  NOT NULL VARCHAR2(30)
PACKAGE_DESCRIPTION                      VARCHAR2(240)
…
PACKAGE_TYPE_CODE               NOT NULL DATE
BACKUP_USERNAME                          VARCHAR2(200)
PARENT_REQUEST_ID                        NUMBER
CREATED_BY                      NOT NULL VARCHAR2(30)
CREATION_DATE                   NOT NULL DATE
…
```

This new column can be used to drive a report, if necessary. For example, to report on packages that have been open for more than five days and assigned to a particular backup user:

```
SELECT backup_username BACKUP_USER,
       assigned_to_username ASSIGNED_USER,
       COUNT(*) NUM_OLD_REQS
FROM   mpkg_packages
WHERE  backup_username = '<ValidUsername>'
AND    close_date IS NULL
AND    cancel_date IS NULL
AND    submission_date IS NOT NULL
AND    (sysdate - submission_date) > 5
GROUP BY backup_username, assigned_to_username
ORDER BY 1, 2;
```

This query also displays the original user to whom the package was assigned.

Table A-4. MPKG_PACKAGES view column descriptions (page 1 of 2)

Column Name	Data Type	Description
PACKAGE_NUMBER	VARCHAR2(40)	Package number.
PACKAGE_ DESCRIPTION	VARCHAR2(240)	Description of this package.
WORKFLOW	VARCHAR2(80)	Workflow used by this package.
NUMBER_OF_LINES	NUMBER	Number of package Lines in this package.
ASSIGNED_TO_ USERNAME	VARCHAR2(200)	Username of PPM Center user to which this package is assigned.
ASSIGNED_TO_GROUP	VARCHAR2(80)	Name of security group to which this package is assigned.
SUBMISSION_DATE	DATE	Date this package was submitted.
PRIORITY	VARCHAR2(80)	Package priority.
PRIORITY_SEQ	NUMBER	Package priority sequence number.
PACKAGE_STATUS	VARCHAR2(80)	Current package status.
PACKAGE_GROUP	VARCHAR2(80)	Package group this package belongs to.
PACKAGE_TYPE	VARCHAR2(80)	Type of package.
Package Global User Data fields	VARCHAR2(200)	One column for each Package Global User Data field. Column name is the User Data field token name.
PARENT_REQUEST_ID	NUMBER	ID of request in Demand Management that spawned this package (if applicable).
CREATED_BY_ USERNAME	VARCHAR2(200)	Username of PPM Center user that created this package.
CREATION_DATE	DATE	Creation date of this package.
LAST_UPDATE_DATE	DATE	Last update date of this package.

Table A-4. MPKG_PACKAGES view column descriptions (page 2 of 2)

Column Name	Data Type	Description
PARENT_STEP_ TRANSACTION_ID	NUMBER	Internal identifier for workflow step transaction record corresponding to the workflow step of the request that spawned this package (if applicable).
WORKFLOW_ID	NUMBER	Internal identifier for package workflow.
PACKAGE_ID	NUMBER	Internal identifier for the package.

MPKG_PENDING_PACKAGES

Used to create a report that shows the volume of open packages for any given workflow in Deployment Management.

Usage

It can be used to get a quick snapshot of ongoing package processing work. It shows a summary of packages currently open for a specific Deployment Management workflow (for example, total number or average age), as well as information showing how many packages have been opened and closed in the current week and current month.

MPKG_PENDING_PACKAGES is aggregated across all packages.

In addition to overall totals of open packages, this view breaks down the information by priority (using the Priority header field). This is done because priority is usually the most important breakdown of load information. Data is grouped into three priority groupings: P1, P2, and P3. These groupings map to the three highest–priority levels defined.

Sample Query

Suppose a project manager has deployments running through three separate workflows in a current project. The manager needs a report that will show current work volume in each of these workflows, to help prioritize work and identify bottlenecks. If the three workflows are named MFG prod deployment, FIN prod deployment, and prod backup, the following SQL query can be used as a basis for a report to display the desired information:

```
SELECT workflow              Workflow,
       open_packages         Open_Pkgs,
       avg_age_open_packages Avg Age,
       p1_open_packages      P1 Open Pkgs,
       p2_open_packages      P2 Open Pkgs
FROM   mpkg_pending_packages
WHERE  workflow IN
       ('MFG prod deployment',
       'FIN prod deployment',
       'prod backup');
```

Results

```
                                    P1    P2
                       Open  Avg  Open  Open
WORKFLOW               Pkgs  Age  Pkgs  Pkgs
--------------------- ----- ---- ----- -----
MFG prod deployment      11    9     3     8
FIN prod deployment      39   16    14    25
prod backup               6   54     5     1
```

This view ignores packages that have not been submitted.

Table A-5. MPKG_PENDING_PACKAGES view column descriptions

Column Name	Data Type	Description
WORKFLOW	VARCHAR2(80)	Name of workflow.
WORKFLOW_ DESCRIPTION	VARCHAR2(240)	Workflow description.
OPEN_PACKAGES	NUMBER	Number of open packages for this workflow.
AVG_AGE_OPEN_ PACKAGES	NUMBER	Average age of open packages.
MAX_AGE_OPEN_ PACKAGES	NUMBER	Age of oldest open package.
P1_OPEN_PACKAGES	NUMBER	Number of open P1 packages.
P2_OPEN_PACKAGES	NUMBER	Number of open P2 packages.
P3_OPEN_PACKAGES	NUMBER	Number of open P3 packages.
P1_AVG_AGE_OPEN_ PACKAGES	NUMBER	Average age of open P1 packages.
P2_AVG_AGE_OPEN_ PACKAGES	NUMBER	Average age of open P2 packages.
P3_AVG_AGE_OPEN_ PACKAGES	NUMBER	Average age of open P3 packages.
P1_MAX_AGE_OPEN_ PACKAGES	NUMBER	Age of oldest open P1 package.
P2_MAX_AGE_OPEN_ PACKAGES	NUMBER	Age of oldest open P2 package.
P3_MAX_AGE_OPEN_ PACKAGES	NUMBER	Age of oldest open P3 package.
PKGS_OPENED_THIS_ MONTH	NUMBER	Number of packages opened this month.
PKGS_OPENED_THIS_ WEEK	NUMBER	Number of packages opened this week.
PKGS_CLOSED_THIS_ MONTH	NUMBER	Number of packages closed this month.
PKGS_CLOSED_THIS_ WEEK	NUMBER	Number of packages closed this week.
WORKFLOW_ID	NUMBER	Internal ID of workflow.

MPKG_REFERENCES

References are used throughout PPM Center to relate transaction entities together. The MPKG_REFERENCES view can be used to view the references of packages in Deployment Management.

Usage

There are several types of references for packages. If a package is part of a release, then there will be a reference for that release. If a package was spawned by a request, then there will be a reference for that request. Packages can be related to other packages through the use of references. References are also used to attach documents to a package.

The RELATIONSHIP column in MPKG_REFERENCES describes the relationship of the referenced item to the package that references it. This view also has columns for each of the entities that can be referenced to a package—other packages, projects, tasks, requests, releases, attachments, and URLs. For each record in MPKG_REFERENCES, only one of these columns will have a value and the others will be NULL.

Sample Query

The following SQL statement can be used to retrieve a list of all references to a particular package:

```
SELECT referenced_package_id PKG,
       referenced_project_id PROJ,
       referenced_request_id REQ,
       referenced_release_id REL,
       referenced_task_id    TASK,
       attachment_name       ATTACHMENT,
       document_url          URL,
       relationship          RELATIONSHIP
FROM   mpkg_references
WHERE  package_number = '30121';
```

Results

```
  PKG  PROJ   REQ   REL  TASK ATTACHMENT         URL               RELATIONSHIP
----- ----- ----- ----- ----- ------------------ ----------------- ----------------------
                  30012                                            Contains this Package
30332                                                              Run after this Package
30043                                                              Run before this Package
30044                                                              Run before this Package
30046                                                              Run before this Package
                              design32_3.doc
```

Table A-6. MPKG_REFERENCES view column descriptions

Column Name	Data Type	Description
PACKAGE_NUMBER	VARCHAR2(40)	Package number for which to show references.
RELATIONSHIP	VARCHAR2(30)	Relationship of reference to this package.
REFERENCED_ PACKAGE_ID	NUMBER	ID of referenced package.
REFERENCED_ PROJECT_ID	NUMBER	ID of referenced project.
REFERENCED_ REQUEST_ID	NUMBER	ID of referenced request.
REFERENCED_ RELEASE_ID	NUMBER	ID of referenced release.
REFERENCED_TASK_ ID	NUMBER	ID of referenced task.
OVERRIDE_FLAG	VARCHAR2(1)	Code to manually override the dependency behavior of the reference.
ATTACHMENT_NAME	VARCHAR2(200)	Name of attached document.
DOCUMENT_URL	VARCHAR2(200)	URL of referenced document on the Web.
PACKAGE_ID	NUMBER	Internal ID of this package.

MPKG_UD_<*Context Value*>

Set of views containing context-sensitive package user data information.

Usage

When the Reporting Meta Layer is synchronized, a view is created for every set of context-sensitive package user data fields defined in the system. The name of each view is defined in the User Data window in the Meta Layer View field. It defaults to a prefix MPKG_UD_ and a suffix that defaults to the first 20 alphanumeric characters of the corresponding context value.

For example, if there are two sets of context-sensitive package user data defined in PPM Center, with a Workflow context field and context values FIN dev -> prod and MFG dev -> prod, then two corresponding Meta Layer views would exist: MPKG_UD_FIN_DEV_PROD and MPKG_UD_MFG_DEV_PROD.

If no context-sensitive package user data has been defined in the User Data window, then no views of this type will exist in the Meta Layer. Global package user data fields are incorporated directly into the package view MPKG_PACKAGES and therefore do not require a separate unique view.

If context-sensitive package user data has been defined, only new packages with this user data and existing packages that have been edited will appear in the views.

Sample Query1

Continuing with the example started above, suppose there are two package user data fields defined for the FIN dev -> prod workflow context, with tokens named VERSION_CTL_PROJECT and VERSION_CTL_ENV.

In the corresponding view MPKG_UD_FIN_DEV_PROD, two columns named the same as the token names would be present:

```
SQL> desc mpkg_ud_fin_dev_prod;
```

Results 1

```
 Name                               Null?    Type
 ------------------------------- -------- ----
 PACKAGE_NUMBER                     NOT NULL VARCHAR2(30)
 PACKAGE_TYPE                       NOT NULL VARCHAR2(80)
 CONTEXT_FIELD                               VARCHAR2(80)
 CONTEXT_VALUE                               VARCHAR2(200)
 CONTEXT_CODE                                VARCHAR2(200)
 VERSION_CTL_PROJECT                         VARCHAR2(200)
 VERSION_CTL_ENV                             VARCHAR2(200)
 CREATION_DATE                      NOT NULL DATE
 CREATED_BY_USERNAME                NOT NULL VARCHAR2(30)
 LAST_UPDATE_DATE                   NOT NULL DATE
 PACKAGE_ID                         NOT NULL NUMBER
```

Sample Query 2

Suppose that a report is needed that shows the number of open packages that are being processed through the FIN dev -> prod workflow, broken down by VERSION_CTL_PROJECT and priority:

```
SELECT f.version_ctl_project PROJECT,
       p.priority PRIORITY,
       COUNT(*) NUM_OPEN_PKGS
FROM   mpkg_ud_fin_dev_prod f,
       mpkg_packages p
WHERE  p.close_date IS NULL
AND    p.cancel_date IS NULL
AND    p.submission_date IS NOT NULL
AND    p.package_id = f.package_id
GROUP BY f.version_ctl_project, p.priority
ORDER BY 1, 2;
```

Results 2

```
PROJECT                           PRIORITY        NUM_OPEN_PKGS
--------------------------------- --------------- -------------
Rel 3.0                           High                        2
                                  Normal                     12
                                  Low                        32

Rel 2.1.2                         Critical                    1
                                  High                        1
                                  Normal                      8
                                  Low                         3

Rel 2.1                           Low                        23
…
```

Table A-7. MPKG_UD_<*Context Value*> view column descriptions

Column Name	Data Type	Description
PACKAGE_NUMBER	VARCHAR2(40)	Package number.
PACKAGE_TYPE	VARCHAR2(80)	Type of package.
CONTEXT_FIELD	VARCHAR2(30)	Field whose value drives this context-sensitive user data.
CONTEXT_VALUE	VARCHAR2(200)	Displayed value of the CONTEXT_FIELD on which this context-sensitive user data is based.
CONTEXT_CODE	VARCHAR2(200)	Hidden code of the CONTEXT_FIELD on which this context-sensitive user data is based.
Package User Data fields for Context Value	VARCHAR2(200)	One column for each context-sensitive user data field for the driving context of this view - column name is the User Data field token name.
CREATION_DATE	DATE	Creation date of this package.
CREATED_BY_ USERNAME	VARCHAR2(200)	Username of PPM Center user that created this package record.
LAST_UPDATE_DATE	DATE	Last update date of this package.
PACKAGE_ID	NUMBER	Internal identifier for this package.

MPKGL_<*Object Type Name*>

Set of views containing object type-specific package line information.

Usage

When the Reporting Meta Layer is synchronized, a view is created for every object type defined in the system. The name of each view is defined on the object type screen in the Meta Layer View field. It defaults to a prefix MPKGL_ and a suffix that defaults to the first 20 alphanumeric characters of the corresponding object type name.

If there are three object types defined in Deployment Management named Java File Migration, SQL Script Migration, and Forms 4.5 Migration, then three corresponding Meta Layer views would exist: MPKGL_JAVA_FILE_MIGRATION, MPKGL_SQL_SCRIPT_MIGRATION, and MPGKL_FORMS_45_MIGRATION.

The view columns are identical to those of the general MPKGL_PACKAGE_LINES view (including the package line user data fields), and they include additional columns for each custom field for the object type. This allows a report designer to create a report that implements business logic that drives off of customer-defined object type fields.

For example, consider the Java File Migration object type mentioned above. This object type might have custom fields with tokens such as FILE_NAME, FILE_LOCATION, and SUB_PATH. The corresponding view MPKGL_JAVA_FILE_MIGRATION would contain columns with these names.

Sample Query 1

```
SQL> desc mpkgl_java_file_migration;
```

Results 1

```
 Name                            Null?    Type
 ------------------------------- -------- ----
 PACKAGE_NUMBER                  NOT NULL VARCHAR2(40)
 LINE_NUMBER                     NOT NULL NUMBER
…
 CANCEL_DATE                              DATE
 FILE_NAME                                VARCHAR2(200)
 SUB_PATH                                 VARCHAR2(200)
 FILE_LOCATION                            VARCHAR2(200)
 CREATION_DATE                   NOT NULL DATE
 CREATED_BY_USERNAME             NOT NULL VARCHAR2(30)
…
```

Sample Query 2

To continue the example, suppose a report is needed that will list the PPM Center user who is assigned to open packages containing one (or more) package lines that are Java File Migration objects, and that are eligible for migration.

A SQL query such as the following might handle this:

```
SELECT p.workflow,
       p.assigned_to_username ASSIGNED_USER,
       COUNT(UNIQUE(p.package_id)) NUM_ELIGIBLE
FROM   mpkg_packages p,
       mpkgl_package_line_actions pla,
       mpkgl_java_file_migration j
WHERE  j.close_date IS NULL
AND    j.cancelled_flag = 'N'
AND    j.submission_date IS NOT NULL
AND    j.package_line_id = pla.package_line_id
AND    pla.status_type = 'ELIGIBLE'
AND    j.package_id = p.package_id
GROUP BY p.workflow, p.assigned_to_username
ORDER BY 1, 2;
```

Table A-8. MPKGL_<*Object Type Name*> view column descriptions (page 1 of 2)

Column Name	Data Type	Description
PACKAGE_NUMBER	VARCHAR2(40)	Package number.
LINE_NUMBER	NUMBER	Sequence number of this package line.
SUBMISSION_DATE	DATE	Date package line was submitted.
OBJECT_TYPE	VARCHAR2(80)	Name of object type of this package line.
OBJECT_NAME	VARCHAR2(300)	Name of this package line.
OBJECT_REVISION	VARCHAR2(300)	Object revision of this package line.
APPLICATION_CODE	VARCHAR2(30)	Application context of this package line.
LINE_STATUS	VARCHAR2(80)	Current status of this package line.
CLOSE_DATE	DATE	If this package line is closed, this is the date on which it was closed.

Table A-8. MPKGL_<*Object Type Name*> view column descriptions (page 2 of 2)

Column Name	Data Type	Description
CANCELLED_FLAG	VARCHAR2(1)	Was this package line cancelled? (Y/N).
CANCEL_DATE	DATE	If this package line was cancelled, this is the date on which it was cancelled.
Package Line Global User Data fields	VARCHAR2(200)	One column for each package line global user data field. Column name is the user data field token name.
Package Line Fields	VARCHAR2(200)	One column for each package line field. Column name is the token name of the field.
CREATION_DATE	DATE	Creation date of this package line.
CREATED_BY_ USERNAME	VARCHAR2(200)	Username of PPM Center user that created this package line.
LAST_UPDATE_DATE	DATE	Last update date of this package line.
LAST_UPDATED_BY_ USERNAME	VARCHAR2(200)	Username of PPM Center user that last updated this package line.
WORKFLOW_ID	NUMBER	Internal identifier for package workflow.
PACKAGE_ID	NUMBER	Internal identifier for the package.
OBJECT_TYPE_ID	NUMBER	Internal identifier for the object type.
PACKAGE_LINE_ID	NUMBER	Internal identifier for the package line.

MPKGL_APP_DEPLOYMENT_D/M

Summary information for package deployment activity, broken down by application, environment, and calendar day (month) for MPKGL_APP_DEPLOYMENT_D (MPKGL_APP_DEPLOYMENT_M).

Usage

This information can be used to quickly assess regular package throughput for each application managed by the IT department, and can help indicate trends in package processing over time for a specified application. An application corresponds to an app code designated in environment definitions.

Besides just the number or packages which were deployed on a given day (month), these views also contain columns to show the number of packages and package lines that were involved in listed deployments, and the number of different object types that were used.

Results from a query of one of these views contain records only for days (months) on which deployments occurred for each application.

Sample Query

The following SQL query can be used as a basis for a report that summarizes all package deployment activity, per day, for a specified application, over a range of dates:

```
SELECT app_code           Application,
       environment        Dest_Env,
       deployment_date    Date,
       total_deployments Total_Deployed,
       unique_obj_types   Num_Obj_Types
FROM   mpkgl_app_deployment_d
WHERE  deployment_date BETWEEN '01-APR-01' AND '05-APR-01'
AND    app_code = 'FINAPP02'
ORDER BY deployment_date;
```

To get a breakdown by month, replace `deployment_date` with `deployment_month` and `mpkgl_app_deployment_d` with `mpkgl_app_deployment_m` in the query above.

Results

```
                                                         Num
                                                Total    Obj
Application  Dest_Env         Date      Deployed   Types
------------ ---------------- --------- -------- -------
FINAPP02     FIN Test 1       01-APR-01       42       4
FINAPP02     FIN Test 2       01-APR-01       12       2
FINAPP02     FIN Prod         01-APR-01        2       1
FINAPP02     FIN Test 1       02-APR-01        3       1
FINAPP02     FIN Test 2       02-APR-01       55       3
FINAPP02     FIN Prod         02-APR-01       39       3
FINAPP02     FIN Test 1       03-APR-01       18       4
FINAPP02     FIN Test 2       03-APR-01       22       3
FINAPP02     FIN Prod         03-APR-01       11       2
…
```

Table A-9. MPKGL_APP_DEPLOYMENT_D view column descriptions

Column Name	Data Type	Description
APP_CODE	VARCHAR2(30)	Application code.
ENVIRONMENT	VARCHAR2(80)	PPM Center environment name.
ENVIRONMENT_ID	NUMBER	Internal ID of environment.
ENVIRONMENT_DESCRIPTION	VARCHAR2(240)	Description of environment.
DEPLOYMENT_DATE	DATE	Day on which deployment occurred.
TOTAL_DEPLOYMENTS	NUMBER	Number of deployments on DEPLOYMENT_DATE.
UNIQUE_PKGS	NUMBER	Number of packages with deployed lines.
UNIQUE_PKG_LINES	NUMBER	Number of package lines deployed.
UNIQUE_OBJ_TYPES	NUMBER	Number of distinct deployed object types.

Table A-10. MPKGL_APP_DEPLOYMENT_M view column descriptions

Column Name	Data Type	Description
APP_CODE	VARCHAR2(30)	Application code.
ENVIRONMENT	VARCHAR2(80)	PPM Center environment name.
ENVIRONMENT_ID	NUMBER	Internal ID of environment.
ENVIRONMENT_ DESCRIPTION	VARCHAR2(240)	Description of environment.
DEPLOYMENT_MONTH	DATE	Month in which deployment occurred.
TOTAL_DEPLOYMENTS	NUMBER	Number of deployments in DEPLOYMENT_MONTH.
UNIQUE_PKGS	NUMBER	Number of packages with deployed lines.
UNIQUE_PKG_LINES	NUMBER	Number of PKG lines deployed.
UNIQUE_OBJ_TYPES	NUMBER	Number of distinct deployed object types.

MPKGL_ENV_DEPLOYMENT_D/M

The Reporting Meta Layer views MPKGL_ENV_DEPLOYMENT_D and MPKGL_ENV_DEPLOYMENT_M give summary information for package deployment activity, broken down by environment and calendar day (month).

Usage

These views can be used to assess regular package throughput for each environment managed by the IT department, and can help indicate trends in package processing over time for a specified environment.

Besides just the number or packages which were deployed on a given day (month), these views also contain columns to show the number of packages and package lines that were involved in listed deployments, and the number of different object types that were used.

Results from a query of one of these views contain records only for days (months) on which deployments occurred for each environment.

Sample Query

The following SQL query can be used as a basis for a report that summarizes all package deployment activity, per day, for a specified environment, over a range of dates:

```
SELECT environment       Dest_Env,
       deployment_date   Date,
       total_deployments Total_Deployed,
       unique_obj_types  Num_Obj_Types
FROM   mpkgl_env_deployment_d
WHERE  deployment_date BETWEEN '01-APR-01' AND '10-APR-01'
AND    environment = 'FIN Test 2'
ORDER BY deployment_date;
```

To get a breakdown by month, replace `deployment_date` with `deployment_month` and `mpkgl_env_deployment_d` with `mpkgl_env_deployment_m` in the query above.

Results

```
                                        Num
                              Total     Obj
Dest_Env         Date      Deployed   Types
---------------- --------- -------- -------
FIN Test 2       01-APR-01       12       2
Fin Test 2       02-APR-01       55       3
FIN Test 2       03-APR-01       22       3
FIN Test 2       04-APR-01        3       1
FIN Test 2       05-APR-01       18       4
FIN Test 2       06-APR-01       39       3
FIN Test 2       07-APR-01       18       4
FIN Test 2       09-APR-01       22       3
FIN Test 2       10-APR-01        3       1
```

Table A-11. MPKGL_ENV_DEPLOYMENT_D view column descriptions

Column Name	Data Type	Description
ENVIRONMENT	VARCHAR2(80)	PPM Center environment name.
ENVIRONMENT_ID	NUMBER	Internal ID of environment.
ENVIRONMENT_ DESCRIPTION	VARCHAR2(240)	Description of environment.
DEPLOYMENT_DATE	DATE	Day on which deployment occurred.
TOTAL_DEPLOYMENTS	NUMBER	Number of deployments on DEPLOYMENT_DATE.
UNIQUE_PKGS	NUMBER	Number of packages with deployed lines.
UNIQUE_PKG_LINES	NUMBER	Number of PKG lines deployed.
UNIQUE_OBJ_TYPES	NUMBER	Number of distinct deployed object types.

Table A-12. MPKGL_ENV_DEPLOYMENT_M view column descriptions

Column Name	Data Type	Description
ENVIRONMENT	VARCHAR2(80)	PPM Center environment name.
ENVIRONMENT_ID	NUMBER	Internal ID of environment.
ENVIRONMENT_ DESCRIPTION	VARCHAR2(240)	Description of environment.
DEPLOYMENT_MONTH	DATE	Month in which deployment occurred.
TOTAL_DEPLOYMENTS	NUMBER	Number of deployments in DEPLOYMENT_MONTH.
UNIQUE_PKGS	NUMBER	Number of packages with deployed lines.
UNIQUE_PKG_LINES	NUMBER	Number of PKG lines deployed.
UNIQUE_OBJ_TYPES	NUMBER	Number of distinct deployed object types.

MPKGL_OBJ_TYPE_DEPLOYMENT_D/M

The views MPKGL_OBJ_TYPE_DEPLOYMENT_D and MPKGL_OBJ_TYPE_DEPLOYMENT_M give summary information for package deployment activity, broken down by object type and calendar day (month).

Usage

These views can be used to assess regular package throughput for each object type used by the IT department, and can help indicate trends in package processing over time for a specified object type.

Besides the number or packages that were deployed on a given day (month), these views also contain columns to show the number of packages and package lines that were involved in listed deployments, and the number of different environments they were deployed to.

Results from a query of one of these views contain records only for days (months) on which deployments occurred for each object type.

Sample Query

The following SQL query can be used as a basis for a report that summarizes all package deployment activity, per month, for a specified object type, over a range of dates:

```
SELECT object_type         Object_Type,
       deployment_month    Month,
       total_deployments   Total_Deployed,
       unique_environments Num_Envs
FROM   mpkgl_obj_type_deployment_m
WHERE  deployment_month BETWEEN '01-MAR-01' AND '01-AUG-01'
AND    object_type = 'File Migration'
ORDER BY deployment_date;
```

To get a breakdown by day, replace `deployment_month` with `deployment_day` and `mpkgl_obj_type_deployment_m` with `mpkgl_obj_type_deployment_d` in the query above.

Results

```
                                      Total      Num
Object_Type       Date       Deployed     Envs
----------------- ---------- --------- -------
File Migration    01-MAR-01       122       12
File Migration    01-APR-01       104       12
File Migration    01-MAY-01        87       15
File Migration    01-JUN-01       156       16
File Migration    01-JUL-01       263       22
File Migration    01-AUG-01       290       23
```

Table A-13. MPKGL_OBJ_TYPE_DEPLOYMENT_D view column descriptions

Column Name	Data Type	Description
OBJECT_TYPE	VARCHAR2(80)	Object type name.
OBJECT_TYPE_ID	NUMBER	Internal ID of object type.
OBJECT_TYPE_ DESCRIPTION	VARCHAR2(240)	Description of object type.
OBJECT_TYPE_ CATEGORY	VARCHAR2(200)	Category of object type.
DEPLOYMENT_DATE	DATE	Day on which deployment occurred.
TOTAL_DEPLOYMENTS	NUMBER	Number of deployments on DEPLOYMENT_DATE.
UNIQUE_PKGS	NUMBER	Number of packages with deployed lines.
UNIQUE_PKG_LINES	NUMBER	Number of PKG lines deployed.
UNIQUE_ ENVIRONMENTS	NUMBER	Number of distinct environments deployed to.

Table A-14. MPKGL_OBJ_TYPE_DEPLOYMENT_M view column descriptions

Column Name	Data Type	Description
OBJECT_TYPE	VARCHAR2(80)	Object type name.
OBJECT_TYPE_ID	NUMBER	Internal ID of object type.
OBJECT_TYPE_ DESCRIPTION	VARCHAR2(240)	Description of object type.
OBJECT_TYPE_ CATEGORY	VARCHAR2(200)	Category of object type.
DEPLOYMENT_MONTH	DATE	Month in which deployment occurred.
TOTAL_DEPLOYMENTS	NUMBER	Number of deployments in DEPLOYMENT_MONTH.
UNIQUE_PKGS	NUMBER	Number of packages with deployed lines.
UNIQUE_PKG_LINES	NUMBER	Number of PKG lines deployed.
UNIQUE_ ENVIRONMENTS	NUMBER	Number of distinct environments deployed to.

MPKGL_PACKAGE_LINES

General view into package line transaction data. A blind query (`SELECT * FROM mpkgl_package_lines;`) will return one row for each package line present in the system, including closed lines.

The view columns map to common package line fields like Sequence, Object Type Name, Object Revision, and App Code. There are also columns for the dates on which it was submitted, closed, and/or cancelled, and for each package line user data field that is defined.

The column name for each package line user data field is the same as the token name for that field.

Usage

This view does not contain an indication of workflow status. Since workflows may be branched and multiple steps might be active at one time, the workflow status is not necessarily a single piece of information that can be represented in a view column. Instead, the report designer must also reference the MPKGL_ PACKAGE_LINE_ACTIONS view for workflow step statuses.

Sample Query

The package line ID is provided as a key column on which to join MPKGL_ PACKAGE_LINE_ACTIONS with MPKGL_PACKAGE_LINES. For example, to list all workflow steps that a particular PPM Center user is eligible to act on:

```
SELECT p.package_number PKG_NUM,
       pl.line_number LINE_NUM,
       pl.object_name OBJECT,
       pla.workflow_step_number STEP_NUM
FROM   mpkg_packages p,
       mpkgl_package_lines pl,
       mwfl_step_security_users ssu,
       mpkgl_package_line_actions pla
WHERE  pla.status_type = 'ELIGIBLE'
AND    ssu.workflow_step_id = pla.workflow_step_id
AND    ssu.username = 'FJOHNSON'
AND    pla.package_line_id = pl.package_line_id
AND    pla.package_id = p.package_id
ORDER BY 1,2,4;
```

The view column PACKAGE_LINE_ID was used to join MPKGL_ PACKAGE_LINES with MPKGL_PACKAGE_LINE_ACTIONS.

The column PACKAGE_ID was used to join MPKGL_PACKAGE_LINES with MPKG_PACKAGES.

Note also the use of the Meta Layer view MWFL_STEP_SECURITY_ USERS, which is used to determine if a specified user is authorized for a specified workflow step.

Table A-15. MPKGL_PACKAGE_LINES view column descriptions (page 1 of 2)

Column Name	Data Type	Description
PACKAGE_NUMBER	VARCHAR2(40)	Package number.
LINE_NUMBER	NUMBER	Sequence number of this package line.
SUBMISSION_DATE	DATE	Date package line was submitted.
OBJECT_TYPE	VARCHAR2(80)	Name of object type of this package line.
OBJECT_NAME	VARCHAR2(300)	Name of this package line.
OBJECT_REVISION	VARCHAR2(300)	Object revision of this package line.
APPLICATION_CODE	VARCHAR2(30)	Application context of this package line.
LINE_STATUS	VARCHAR2(80)	Current status of this package line.
CLOSE_DATE	DATE	If this package line is closed, this is the date on which it was closed.
CANCELLED_FLAG	VARCHAR2(1)	Was this package line cancelled? (Y/N).
CANCEL_DATE	DATE	If this package line was cancelled, this is the date on which it was cancelled.
Package Line Global User Data fields	VARCHAR2(200)	One column for each package line global user data field. Column name is the user data field token name.
CREATION_DATE	DATE	Creation date of this package line.
CREATED_BY_ USERNAME	VARCHAR2(200)	Username of PPM Center user that created this package line.
LAST_UPDATE_DATE	DATE	Last update date of this package line.

Table A-15. MPKGL_PACKAGE_LINES view column descriptions (page 2 of 2)

Column Name	Data Type	Description
LAST_UPDATED_BY_ USERNAME	VARCHAR2(200)	Username of PPM Center user that last updated this package line.
WORKFLOW_ID	NUMBER	Internal identifier for package workflow.
PACKAGE_ID	NUMBER	Internal identifier for the package.
OBJECT_TYPE_ID	NUMBER	Internal identifier for the object type.
PACKAGE_LINE_ID	NUMBER	Internal identifier for the package line.

MPKGL_PACKAGE_LINE_ACTIONS

Used to gather transaction details for any given package line in Deployment Management. Contains columns to display the current status of a step, how long that step has been in the current status, whether the step is complete or resulted in an error, details about the step (source and destination environment), and other relevant details.

Usage

To relate information from this view with detail information from related packages or package lines, the report designer can use the package and package line identifiers (PACKAGE_ID and PACKAGE_LINE_ID columns) to join with other standard views such as MPKG_PACKAGES and MPKGL_PACKAGE_LINES.

Sample Query

Suppose a report is needed that shows the number of package lines that have had certain actions taken for each calendar week in the last month, broken down by object type, for a customer’s Dev-Test-Prod workflow:

```
SELECT  trunc(eligible_date,'WW')                      Week,
        line_object_type                               Object_Type,
        sum(decode(action_name,'Open',1,0))            Opened,
        sum(decode(action_name,'Migrate to Test',1,0)) Into_Test,
        sum(decode(action_name,'Migrate to Prod',1,0)) Into_Prod,
        sum(decode(action_name,'Close',1,0))           Closed
FROM    mpkgl_package_line_actions
WHERE   package_workflow = 'Dev - Test - Prod'
AND     eligible_date > sysdate - 30
GROUP BY trunc(eligible_date,'WW'),
         line_object_type;
```

The column STATUS is the status name that is displayed for lines in the status tab of packages in the Deployment Management application. The internal code STATUS_TYPE is provided to group these status names into logical groupings.

For example, there may be many different statuses that all represent a COMPLETE status type (for example, the result value of any workflow step, such as Approved, Succeeded, Rejected, Failed QA Test).

While STATUS may have many different possible values, STATUS_TYPE has any of the following possible values:

- SUBMITTED
- IN_PROGRESS
- CLOSED_SUCCESS

- ELIGIBLE
- ERROR
- CLOSED_FAILURE
- PENDING
- COMPLETE
- CANCELLED

The internal code STEP_TRANSACTION_ID is provided in this view for use with the Meta Layer view MWFL_TRANSITIONS, which can be used to get detailed information about previous or subsequent process steps.

Table A-16. MPKGL_PACKAGE_LINE_ACTIONS view column descriptions (page 1 of 3)

Column Name	Data Type	Description
PACKAGE_NUMBER	VARCHAR2(40)	Package number.
PACKAGE_ DESCRIPTION	VARCHAR2(240)	Description of the package.
LINE_NUMBER	NUMBER	Number of this line in the package.
LINE_OBJECT_TYPE	VARCHAR2(80)	Object type of this line.
LINE_OBJECT_NAME	VARCHAR2(300)	Value of line's Object Name field.
LINE_OBJECT_ REVISION	VARCHAR2(300)	Value of line's Object Revision field.
LINE_APPLICATION_ CODE	VARCHAR2(30)	App code of this line.
PACKAGE_WORKFLOW	VARCHAR2(80)	Top-level workflow used by this package.
LINE_WORKFLOW_ STEP_LABEL	VARCHAR2(2000)	Visible label of this step on this line in package status tab.
ACTION_NAME	VARCHAR2(80)	Name of workflow step action.
WORKFLOW	VARCHAR2(80)	Name of workflow that contains this step.
WORKFLOW_STEP_ NUMBER	NUMBER	Workflow sequence number of this step.

Table A-16. MPKGL_PACKAGE_LINE_ACTIONS view column descriptions (page 2 of 3)

Column Name	Data Type	Description
STATUS	VARCHAR2(200)	Visible status of this package line.
STATUS_TYPE	VARCHAR2(30)	Internal code for STATUS.
ELIGIBLE_DATE	DATE	Date this step became eligible.
ACTION_DATE	DATE	Date action was taken on this step.
ACTION_RESULT	VARCHAR2(200)	Result of the action.
ERROR_MESSAGE	VARCHAR2(240)	(If STATUS_TYPE = 'ERROR') error message.
DURATION	NUMBER	Number of days at this status, or until completed (if STATUS_TYPE = 'COMPLETE').
SOURCE_ ENVIRONMENT	VARCHAR2(80)	Source environment (if applicable).
SOURCE_ ENVIRONMENT_ GROUP	VARCHAR2(100)	Source environment group (if applicable).
DEST_ENVIRONMENT	VARCHAR2(80)	Destination environment (if applicable).
DEST_ENVIRONMENT_ GROUP	VARCHAR2(100)	Destination environment group (if applicable).
USER_COMMENT	VARCHAR2(240)	User comment entered when taking action on this step.
CHILD_REQUEST_ID	NUMBER	ID of child request.
CHILD_PACKAGE_ID	NUMBER	Internal ID of child package.
STEP_TRANSACTION_ ID	NUMBER	Internal ID of this transaction.
PACKAGE_ID	NUMBER	Internal ID of the package.
PACKAGE_LINE_ID	NUMBER	Internal ID of this line in the package.

Table A-16. MPKGL_PACKAGE_LINE_ACTIONS view column descriptions (page 3 of 3)

Column Name	Data Type	Description
PACKAGE_ WORKFLOW_ID	NUMBER	Internal ID for top-level workflow used by this package.
WORKFLOW_ID	NUMBER	Internal ID for the workflow that contains this workflow step.
WORKFLOW_STEP_ID	NUMBER	Internal ID of this workflow step.
ACTION_RESULT_ CODE	VARCHAR2(200)	Internal code for ACTION_ RESULT.
SOURCE_ ENVIRONMENT_ID	NUMBER	Internal ID of source environment.
SOURCE_ ENVIRONMENT_ GROUP_ID	NUMBER	Internal ID of source environment group.
DEST_ENVIRONMENT_ ID	NUMBER	Internal ID of destination environment.
DEST_ENVIRONMENT_ GROUP_ID	NUMBER	Internal ID of destination environment group.

MPKGL_PENDING_DEPLOYMNT_BY_ENV/APP/OT

Summarizes the number of open packages and package lines that are currently pending deployment into environments.

The deployment information is broken down into a different category for each view:

- To see the distribution of the number of objects pending deployment across environments, use the view MPKGL_PENDING_DEPLOYMNT_BY_ENV.
- To see the same information distributed across applications, use MPKGL_PENDING_DEPLOYMNT_BY_APP.
- To see the same deployment information distributed across object types, use MPKGL_PENDING_DEPLOYMENT_BY_OT.

Sample Query

To obtain a quick look at the volume of deployments queued up at each environment defined in the system (for those with one or more pending deployments):

```
SELECT environment,
       total_count,
       unique_pkgs,
       unique_pkg_lines,
       unique_obj_types
FROM   mpkgl_pending_deploymnt_by_env;
```

The internal ID columns for Environments and Object Types (ENVIRONMENT_ID and OBJECT_TYPE_ID) can be used to link this view with other relevant views (for example, MPKGL_PACKAGE_LINES) to provide additional information in a report built from these views.

This view will not capture processes in which the package line is waiting at an approval step which will fire an immediate execution step.

Table A-17. MKPGL_PENDING_DEPLOYMNT_BY_ENV view column descriptions (page 1 of 2)

Column Name	Data Type	Description
ENVIRONMENT	VARCHAR2(80)	Environment name.
ENVIRONMENT_ID	NUMBER	Environment name.
ENVIRONMENT_DESCRIPTION	VARCHAR2(240)	Internal ID of environment.

Table A-17. MKPGL_PENDING_DEPLOYMNT_BY_ENV view column descriptions (page 2 of 2)

Column Name	Data Type	Description
TOTAL_COUNT	NUMBER	Total number of pending objects.
UNIQUE_PKGS	NUMBER	Number of packages with pending objects.
UNIQUE_PKG_LINES	NUMBER	Number of PKG lines with pending objects.
UNIQUE_OBJ_TYPES	NUMBER	Number of distinct pending object types.

Table A-18. MPKGL_PENDING_DEPLOYMNT_BY_APP view column descriptions

Column Name	Data Type	Description
APP_CODE	VARCHAR2(30)	Application code.
ENVIRONMENT	VARCHAR2(80)	Environment name.
ENVIRONMENT_ID	NUMBER	Internal ID of environment.
ENVIRONMENT_ DESCRIPTION	VARCHAR2(240)	Description of environment.
TOTAL_COUNT	NUMBER	Total number of pending objects.
UNIQUE_PKGS	NUMBER	Number of packages with pending objects.
UNIQUE_PKG_LINES	NUMBER	Number of PKG lines with pending objects.
UNIQUE_OBJ_TYPES	NUMBER	Number of distinct pending object types.

Table A-19. MPKGL_PENDING_DEPLOYMENT_BY_OT view column descriptions

Column Name	Data Type	Description
OBJECT_TYPE	VARCHAR2(80)	Object type name.
OBJECT_TYPE_ID	NUMBER	Internal ID of object type.
OBJECT_TYPE_DESCRIPTION	VARCHAR2(240)	Description of object type.
OBJECT_TYPE_CATEGORY	VARCHAR2(80)	Category of object type.
TOTAL_COUNT	NUMBER	Total number of pending objects.
UNIQUE_PKGS	NUMBER	Number of packages with pending objects.
UNIQUE_PKG_LINES	NUMBER	Number of PKG lines with pending objects.
UNIQUE_ENVIRONMENTS	NUMBER	Number of distinct pending environments.

MREL_DISTRIBUTIONS

Used to gather information about distributions of releases in Deployment Management. Contains columns to display the workflow used by a distribution, a distribution's status, whether a distribution has provided a feedback value to contained packages, and other information.

Usage

To relate information from this view with information from related views, the report designer can use the release identifier RELEASE_ID and distribution identifier DISTRIBUTION_ID to join with other views like MREL_ RELEASES and MREL_DISTRIBUTION_ACTIONS.

Also provided is the DIST_WORKFLOW_ID, which can be useful in joining to workflow views such as MWFL_WORKFLOWS to include information about the workflows being used by a distribution.

Table A-20. MREL_DISTRIBUTIONS view column descriptions (page 1 of 2)

Column Name	Data Type	Description
RELEASE_NAME	VARCHAR2(200)	Name of release for this distribution.
DISTRIBUTION_NAME	VARCHAR2(80)	Name of distribution.
DIST_DESCRIPTION	VARCHAR2(240)	Distribution description.
DIST_WORKFLOW	VARCHAR2(80)	Workflow used by this distribution.
DIST_STATUS	VARCHAR2(80)	Current status of the distribution.
PACKAGE_FEEDBACK_ FLAG	VARCHAR2(1)	Has package feedback occurred? (Y/N).
PACKAGE_FEEDBACK	VARCHAR2(80)	Result value fed back to packages.
CREATION_DATE	DATE	Date this distribution was created.
CREATED_BY_ USERNAME	VARCHAR2(200)	Username of PPM Center user that created this distribution.
LAST_UPDATE_DATE	DATE	Date this distribution was last updated.

Table A-20. MREL_DISTRIBUTIONS view column descriptions (page 2 of 2)

Column Name	Data Type	Description
RELEASE_ID	NUMBER	Internal ID of the release.
DISTRIBUTION_ID	NUMBER	Internal ID of this distribution.
DIST_WORKFLOW_ID	NUMBER	Internal ID of the distribution workflow.

MREL_DISTRIBUTION_ACTIONS

Used to gather information about current workflow steps for any given release distribution in Deployment Management. Contains columns to display the current status of a step, how long that step has been in the current status, whether the step is complete or resulted in an error, details about the step (source and destination Environment), and other relevant details.

Usage

To relate information from this view with information from related releases or release distributions, the report designer can use the release and distribution identifiers (RELEASE_ID and DISTRIBUTION_ID columns) to join with other standard views like MREL_RELEASES and MREL_DISTRIBUTIONS.

Sample Query

Suppose a report is needed that takes a release name input from the user running the report, and shows the details of all open distributions of the release:

```
SELECT release_name                                    RELEASE_NAME,
       distribution_name                             DISTRIBUTION_NAME,
       dist_workflow_step_label || ': '|| action_name
                                                       ELIGIBLE_STEP,
       duration                                        DAYS_ELIGIBLE
FROM   mrel_distribution_actions
WHERE  status_type = 'ELIGIBLE'
GROUP BY release_name,
         distribution_name,
         dist_workflow_step_label || ': '|| action_name,
         duration
ORDER BY 1,2;
```

The column STATUS is the status name that is displayed in the status tab of distributions in the Deployment Management application.

The internal code STATUS_TYPE is provided to group these status names into logical groupings. For example, there may be many different statuses that all represent a COMPLETE status type (for example, the result value of any workflow step like Approved, Succeeded, Rejected, Failed QA Test).

While STATUS may have many different possible values, STATUS_TYPE has only the following possible values:

- SUBMITTED
- IN_PROGRESS
- CLOSED_SUCCESS
- ELIGIBLE
- ERROR
- CLOSED_FAILURE
- PENDING
- COMPLETE
- CANCELLED

The internal code STEP_TRANSACTION_ID is provided in this view for use with the Meta Layer view MWFL_TRANSITIONS, which can be used to get detailed information about previous or subsequent process steps.

Table A-21. MREL_DISTRIBUTION_ACTIONS view column descriptions (page 1 of 3)

Column Name	Data Type	Description
DISTRIBUTION_NAME	VARCHAR2(80)	Release distribution name.
DIST_DESCRIPTION	VARCHAR2(240)	Description of the distribution.
RELEASE_NAME	VARCHAR2(200)	Name of the parent release.
DIST_WORKFLOW	VARCHAR2(80)	Top-level distribution workflow.
DIST_WORKFLOW_STEP_LABEL	VARCHAR2(2000)	Visible label of this step on this line in package status tab.
ACTION_NAME	VARCHAR2(80)	Name of workflow step action.
WORKFLOW	VARCHAR2(80)	Name of workflow that contains this step.
WORKFLOW_STEP_NUMBER	NUMBER	Workflow sequence number of this step.
STATUS	VARCHAR2(200)	Visible status of this distribution.
STATUS_TYPE	VARCHAR2(30)	Internal code for STATUS.

Table A-21. MREL_DISTRIBUTION_ACTIONS view column descriptions (page 2 of 3)

Column Name	Data Type	Description
ELIGIBLE_DATE	DATE	Date this step became eligible.
ACTION_DATE	DATE	Date action was taken on this step.
ACTION_RESULT	VARCHAR2(200)	Result of the action.
ERROR_MESSAGE	VARCHAR2(240)	(If STATUS_TYPE = ‘ERROR’) error message.
DURATION	NUMBER	Number of days at this status, or until completed (if STATUS_TYPE = 'COMPLETE').
SOURCE_ ENVIRONMENT	VARCHAR2(80)	Source environment (if applicable).
SOURCE_ ENVIRONMENT_ GROUP	VARCHAR2(100)	Source environment group (if applicable).
DEST_ENVIRONMENT	VARCHAR2(80)	Destination environment (if applicable).
DEST_ENVIRONMENT_ GROUP	VARCHAR2(100)	Destination environment group (if applicable).
USER_COMMENT	VARCHAR2(240)	User comment entered when taking action on this step.
CHILD_REQUEST_ID	NUMBER	ID of child request.
CHILD_PACKAGE_ID	NUMBER	Internal ID of child package.
STEP_TRANSACTION_ ID	NUMBER	Internal ID of this transaction.
RELEASE_ID	NUMBER	Internal ID of the release.
DISTRIBUTION_ID	NUMBER	Internal ID of this distribution.
DIST_WORKFLOW_ID	NUMBER	Internal ID for top-level workflow used by this distribution.
WORKFLOW_ID	NUMBER	Internal ID for the workflow that contains this Workflow step.

Table A-21. MREL_DISTRIBUTION_ACTIONS view column descriptions (page 3 of 3)

Column Name	Data Type	Description
WORKFLOW_STEP_ID	NUMBER	Internal ID of this workflow step.
ACTION_RESULT_ CODE	VARCHAR2(200)	Internal code for ACTION_ RESULT.
SOURCE_ ENVIRONMENT_ID	NUMBER	Internal ID of source environment.
SOURCE_ ENVIRONMENT_ GROUP_ID	NUMBER	Internal ID of source environment group.
DEST_ENVIRONMENT_ ID	NUMBER	Internal ID of destination environment.
DEST_ENVIRONMENT_ GROUP_ID	NUMBER	Internal ID of destination environment group.

MREL_REFERENCES

Used to view the references of releases in Deployment Management.

Usage

There are several types of references for releases. If a package is part of a release, then there will be a reference for that package. Similarly, if a request is part of a release, then there will be a reference for that request. Releases can be designated as children or parents of other releases through the use of references. References are also used to attach documents to a release.

The RELATIONSHIP column in MREL_REFERENCES describes the relationship of the referenced item to the release that references it. This view also has columns for each of the entities that can be referenced to a release: other releases, requests, packages, attachments, and URLs.

For each record in MREL_REFERENCES, only one of these columns will have a value and the others will be NULL.

Sample Query

To retrieve a list of all references to a particular release:

```
SELECT referenced_release_id REL,
       referenced_package_id PKG,
       referenced_request_id REQ,
       attachment_name       ATTACHMENT,
       document_url          URL,
       relationship          RELATIONSHIP
FROM   mrel_references
WHERE  release_name = 'FIN Apps Prod Release';
```

Results

```
REL   PKG   REQ ATTACHMENT           URL                 RELATIONSHIP
----- ----- ----- -------------------- ------------------- ----------------------
30012                                                      Parent Release
                  finAppsRelease.doc
      42764                                                Contained in this Release
      42765                                                Contained in this Release
      42772                                                Contained in this Release
      42773                                                Contained in this Release
      42774                                                Contained in this Release
      42778                                                Contained in this Release
…
```

Table A-22. MREL_REFERENCES view column descriptions

Column Name	Data Type	Description
RELEASE_NAME	VARCHAR2(200)	Release for which to show references.
RELATIONSHIP	VARCHAR2(30)	Relationship of reference to this release.
REFERENCED_ PACKAGE_ID	NUMBER	ID of referenced package.
REFERENCED_ PROJECT_ID	NUMBER	ID of referenced project.
REFERENCED_ REQUEST_ID	NUMBER	ID of referenced request.
REFERENCED_ RELEASE_ID	NUMBER	ID of referenced release.
REFERENCED_TASK_ ID	NUMBER	ID of referenced task.
OVERRIDE_FLAG	VARCHAR2(1)	Code to manually override the dependency behavior of the reference.
ATTACHMENT_NAME	VARCHAR2(200)	Name of attached document.
DOCUMENT_URL	VARCHAR2(200)	URL of referenced document on the Web.
RELEASE_ID	NUMBER	Internal ID of this release.

MREL_RELEASES

Used to gather information about releases in Deployment Management. Contains columns to display the current status of a release, the number of distributions that have been deployed for a release, the manager, team, and group of a release, and other information.

Usage

To relate information from this view with information from related distributions, the report designer can use the release identifier RELEASE_ID to join with other views such as MREL_DISTRIBUTIONS or MREL_REFERENCES.

Sample Query

Suppose a report is needed that shows details about releases that are part of the release team FIN Apps Prod Release. To show all packages that are included in relevant releases, and their statuses:

```
SELECT r.release_name    RELEASE,
       r.release_status  REL_STATUS,
       p.package_number  PKG_NUMBER,
       p.package_status  PKG_STATUS
FROM   mpkg_packages p,
       mrel_references ref,
       mrel_releases r
WHERE  r.release_team = 'FIN Apps Prod Release'
AND    r.release_id = ref.release_id
AND    p.package_id = ref.referenced_package_id
ORDER BY r.release_name, p.package_number;
```

Results

```
RELEASE        REL_STATUS         PKG_NUMBER      PKG_STATUS
-------------- ------------------ --------------- ------------------
Apply to Test  Code Freeze        43002           Ready for Release
Apply to Test  Code Freeze        43004           Ready for Release
Apply to Test  Code Freeze        43005           In Progress
Apply to Test  Code Freeze        43007           Ready for Release
…
```

The column RELEASE_STATUS in MREL_RELEASES is the status displayed in the releases screen in the Deployment Management application. The RELEASE_STATUS column has the following possible values:

- New
- Code freeze
- Open
- Closed

Table A-23. MREL_RELEASES view column descriptions

Column Name	Data Type	Description
RELEASE_NAME	VARCHAR2(200)	Name of release.
RELEASE_DESCRIPTION	VARCHAR2(200)	Release description.
RELEASE_STATUS	VARCHAR2(200)	Current status of release.
RELEASE_MANAGER_USERNAME	VARCHAR2(200)	Username of PPM Center user designated as release manager.
RELEASE_TEAM	VARCHAR2(200)	Name of security group designated as release team.
RELEASE_GROUP	VARCHAR2(200)	Release group, if this release is categorized as part of a group.
NUMBER_OF_DISTRIBUTIONS	NUMBER	Number of distributions created to date.
CREATION_DATE	DATE	Date this release was created.
CREATED_BY_USERNAME	VARCHAR2(200)	Username of PPM Center user that created this release.
LAST_UPDATE_DATE	DATE	Date this release was last updated.
RELEASE_ID	NUMBER	Internal ID of this release.
RELEASE_MANAGER_USER_ID	NUMBER	Internal ID of release manager.
RELEASE_TEAM_ID	VARCHAR2(200)	Internal ID of release team security group.

Demand Management Views

Demand Management views provide information specific to HP Demand Management. For example, MREQ_OPENED_CLOSED_BY_TYPE_D provides summary information for request submission and completion activity, broken down by request type and by calendar day. MREQ-REFERENCES provide information about references related to HP Demand Management requests.

MREQ_<*Request Type Name*>

Contains request type-specific information.

Usage

When the Reporting Meta Layer is synchronized, a view is created for every request type defined in the system. The name of each view is defined on the request type screen in the Meta Layer View field. It defaults to a prefix MREQ_ and a suffix that defaults to the first 20 alphanumeric characters of the corresponding request type name. For example, if there are three request types defined in Demand Management named Support Ticket, Bug, and Work Order, then three corresponding Meta Layer views would exist: MREQ_SUPPORT_TICKET, MREQ_BUG, and MREQ_WORK_ORDER.

The view columns are identical to those of the general MREQ_ALL_REQUESTS view (including the global request user data fields), and they also include additional columns for each custom request detail field for the request type. This allows a report designer to create a report that implements business logic that drives off of customer-defined request detail fields.

Sample Query 1

For example, consider the Work Order request type mentioned above. This request type might have custom detail fields with tokens like CUSTOMER, TIME_ESTIMATE, and ACTUAL_TIME. The corresponding view MREQ_WORK_ORDER would contain columns with these names:

```
SQL> desc mreq_work_order
```

Results 1

```
 Name                            Null?    Type
 ------------------------------- -------- ----
 REQUEST_ID                      NOT NULL NUMBER
 REQUEST_DESCRIPTION             NOT NULL VARCHAR2(240)
 SUBMISSION_DATE                 NOT NULL DATE
 REQUEST_STATUS                  NOT NULL VARCHAR2(80)
…
 CANCEL_DATE                     NOT NULL DATE
 BACKUP_USERNAME                          VARCHAR2(200)
 CUSTOMER                                 VARCHAR2(200)
 TIME_ESTIMATE                            VARCHAR2(200)
 ACTUAL_TIME                              VARCHAR2(200)
 REQUEST_TYPE_NAME                        VARCHAR2(80)
 REQUEST_SUBTYPE_NAME                     VARCHAR2(80)
…
```

Sample Query 2

Suppose a report is needed that will list information about work order requests in which the actual time was more than one day longer than the estimated time.

An SQL query such as the following would handle this:

```
SELECT request_number REQUEST_NUM,
       status_name CURRENT_STATUS,
       customer CUSTOMER,
       (actual_time - time_estimate) EXTRA_DAYS_WORKED
FROM   mreq_work_order
WHERE  time_estimate IS NOT NULL
AND    actual_time IS NOT NULL
AND    (actual_time - time_estimate) > 1
ORDER BY request_number;
```

Table A-24. MREQ_<*Request Type Name*> view column descriptions (page 1 of 3)

Column Name	Data Type	Description
REQUEST_ID	NUMBER	Name of the request record.
REQUEST_ DESCRIPTION	VARCHAR2(240)	Request description.
SUBMISSION_DATE	DATE	Date request was submitted.
REQUEST_STATUS	VARCHAR2(80)	Current status of the request.
WORKFLOW	VARCHAR2(80)	Name of workflow used by this request.
DEPARTMENT	VARCHAR2(80)	Visible value of request department.
DEPARTMENT_CODE	VARCHAR2(30)	Internal code for department.

Table A-24. MREQ_<*Request Type Name*> view column descriptions (page 2 of 3)

Column Name	Data Type	Description
PRIORITY	VARCHAR2(80)	Visible value of request priority.
PRIORITY_CODE	VARCHAR2(30)	Internal code for priority.
APPLICATION	VARCHAR2(80)	Visible value of request application.
APPLICATION_CODE	VARCHAR2(30)	Internal code for application.
REQUEST_GROUP	VARCHAR2(80)	Visible value of request group.
REQUEST_GROUP_CODE	VARCHAR2(30)	Internal code for request group.
ASSIGNED_TO_USERNAME	VARCHAR2(200)	Username of PPM Center user to which this request is assigned.
ASSIGNED_TO_GROUP	VARCHAR2(80)	Name of security group to which this request is assigned.
CONTACT_FULL_NAME	VARCHAR2(80)	Full name of request contact.
CONTACT_LAST_NAME	VARCHAR2(30)	Last name of request contact.
CONTACT_FIRST_NAME	VARCHAR2(30)	First name of request contact.
CONTACT_PHONE	VARCHAR2(30)	Phone number of request contact.
CONTACT_EMAIL	VARCHAR2(80)	Email address of request contact.
COMPANY	VARHCAR2(80)	Company of request contact.
CLOSE_RESULT	VARCHAR2(80)	If this request is closed, this is the visible result value.
CLOSE_DATE	DATE	If this request is closed, this is the date on which it was closed.
CANCEL_DATE	DATE	If this request is cancelled, this is the date on which it was cancelled.

Table A-24. MREQ_<*Request Type Name*> view column descriptions (page 3 of 3)

Column Name	Data Type	Description
Request Global User Data	VARCHAR2(200)	One column for each request global user data field. Column name is the user data field token name.
Request Detail Fields	VARCHAR2(200)	One column for each custom field defined on the request type (including custom header fields, custom detail fields, and field group fields).
REQUEST_TYPE	VARCHAR2(80)	Name of the request type.
REQUEST_SUBTYPE	VARCHAR2(80)	Name of the request subtype.
REQUEST_HEADER_ TYPE	VARCHAR2(80)	Name of the request header type.
CREATION_DATE	DATE	Date the request record was created.
CREATED_BY_ USERNAME	VARCHAR2(200)	Username of PPM Center user who created the request.
LAST_UPDATE_DATE	DATE	Date the request record was last updated.
CONTACT_ID	NUMBER	Internal identifier for contact.
REQUEST_TYPE_ID	NUMBER	Internal identifier for request type.
WORKFLOW_ID	NUMBER	Internal identifier for workflow.

MREQ_CONTACTS

Contains all fields for contacts defined in Demand Management. Contains all relevant pieces of information about a contact, including a denormalized username (if present) and a column for each Contact User Data field defined in the system. The column name for each Contact User Data field is the same as the token name for that field.

A subset of the information provided here is also present in the request views MREQ_REQUESTS and MREQ_<*Request Type Name*>.

Sample Query 1

```
SELECT full_name NAME,
       phone_number PHONE_NUMBER,
       email_address EMAIL
FROM   mreq_contacts
WHERE  enabled_flag = 'Y';
```

Sample Query 2

If there are Contact User Data fields defined, the token for each field will appear as a separate column in MREQ_CONTACTS.

For example, consider a scenario in which two Contact User Data fields have been defined to track additional contact information, with tokens PAGER_ NUMBER and HOME_PHONE_NUMBER. Two columns with the same names would be present in MREQ_CONTACTS:

```
SQL> desc mreq_contacts
```

Results 2

```
 Name                            Null?    Type
 ------------------------------- -------- ----
 LAST_NAME                       NOT NULL VARCHAR2(30)
 FIRST_NAME                      NOT NULL VARCHAR2(30)
…
 PAGER_NUMBER                             VARCHAR2(200)
 HOME_PHONE_NUMBER                        VARCHAR2(200)
 ENABLED_FLAG                    NOT NULL VARCHAR2(1)
 CREATION_DATE                   NOT NULL DATE
…
```

Sample Query 3

For a slightly more complex example (building on the previous example from the MREQ_REQUESTS section, in which a global Request User Data field called BACKUP_USERNAME has been defined), consider designing a report to print the full name and pager, work, and home phone numbers of all users who are assigned as backup users on requests that have been open for more than 5 days.

An SQL statement to achieve this type of information might look as follows:

```
SELECT r.backup_username USERNAME,
       c.full_name NAME,
       c.pager_number PAGER_NUMBER,
       c.phone_number WORK_NUMBER,
       c.home_phone_number HOME_NUMBER
FROM   mreq_contacts c,
       mreq_requests r
WHERE  c.enabled_flag = 'Y'
AND    r.backup_username = c.username (+)
AND    r.close_date IS NULL
AND    r.cancel_date IS NULL
AND    r.submission_date IS NOT NULL
AND    (sysdate - r.submission_date) > 5;
```

Table A-25. MREQ_CONTACTS view column descriptions (page 1 of 2)

Column Name	Data Type	Description
LAST_NAME	VARCHAR2(30)	Last name of contact.
FIRST_NAME	VARCHAR2(30)	First name of contact.
FULL_NAME	VARCHAR2(80)	Full name of contact.
USERNAME	VARCHAR2(200)	PPM Center username of contact, if applicable.
PHONE_NUMBER	VARCHAR2(30)	Phone number of contact.
EMAIL_ADDRESS	VARCHAR2(80)	Email address of contact.
COMPANY	VARCHAR2(80)	Company of contact.
Contact Global User Data	VARCHAR2(200)	One column for each Contact Global User Data field. Column name is the user data field token name.
ENABLED_FLAG	VARCHAR2(1)	Is this contact enabled? (Y/N).
CREATION_DATE	DATE	Date contact record was created.

Table A-25. MREQ_CONTACTS view column descriptions (page 2 of 2)

Column Name	Data Type	Description
CREATED_BY_ USERNAME	VARCHAR2(200)	Username of PPM Center user who created this contact record.
LAST_UPDATE_DATE	DATE	Date contact record was last updated.
CONTACT_ID	NUMBER	Internal identifier for this contact.

MREQ_CHANGES

When a field is being audited, a record is stored in the PPM Center database every time the value in that field changes on any open Request. This audit history can be important to business decision-making.

MREQ_CHANGES allows a report to display and drive off of changes to request fields. This view exposes the audit trail for the request header and detail fields. It contains columns for the old and new values, and the field prompts and tokens.

Sample Query

To report on the frequency at which the request priority is changed from any value to Critical, an SQL statement such as the following can be used:

```
SELECT m.calendar_month MONTH,
       c.old_field_value OLD_VALUE,
       count(*) NUM_CHANGED
FROM   mreq_changes c,
       krml_calendar_months m
WHERE  c.field_prompt = 'Priority'
AND    c.new_field_code = 'C'
AND    c.change_date >= m.start_date
AND    c.change_date < m.end_date
GROUP BY m.calendar_month, c.old_field_value
ORDER BY 1, 2;
```

In the WHERE clause of this statement that we are testing, the NEW_FIELD_CODE is used instead of the NEW_FIELD_VALUE. Either would work.

C is the code for the Critical priority; this statement could also have been written `WHERE c.new_field_value = 'Critical'`.

The validation for the request priority field contains the hidden and visible values for this field. Consult this validation in the Validations window for verification of these values.

Consider a slight extension to the previous SQL statement. If it was necessary to limit this information to a specific request type, an additional AND condition could be used: AND `c.request_type = '<`*`Name`*`>'`.

Table A-26. MREQ_CHANGES view column descriptions

Column Name	Data Type	Description
REQUEST_ID	NUMBER	ID of the parent request for the field.
CHANGE_DATE	DATE	Date change occurred.
CHANGED_BY_ USERNAME	VARCHAR2(200)	Username of PPM Center user who made change.
REQUEST_TYPE	VARCHAR2(80)	Request type of the parent request for the field.
COLUMN_NAME	VARCHAR2(30)	Name of field column whose value changed.
FIELD_PROMPT	VARCHAR2(80)	Prompt of field whose value changed.
FIELD_TOKEN	VARCHAR2(30)	Token for field whose value changed.
OLD_FIELD_VALUE	VARCHAR2(1800)	Field value before change.
NEW_FIELD_VALUE	VARCHAR2(1800)	Field value after change.
OLD_FIELD_CODE	VARCHAR2(1800)	Field hidden code before change.
NEW_FIELD_CODE	VARCHAR2(1800)	Field hidden code after change.

MREQ_NOTES

Provides access to the notes for all requests in Demand Management.

Usage

Notes are stored in an Oracle LONG database column; to prevent an overload of information this was presented in a separate Meta Layer view, making it less likely to design a report that inadvertently returns too much data.

To query request notes, join this view with a request view (MREQ_ REQUESTS, or a request type-specific view MREQ_<*Request Type Name*>).

Sample Query

To retrieve a list of the notes for all open requests of the Bug request type, that have Critical priority, use the following logic in an SQL statement:

```
SELECT r.request_number REQ_NUM,
       n.notes NOTES
FROM   mreq_bug r,
       mreq_notes n
WHERE  r.priority = 'Critical'
AND    r.request_id = n.request_id;
```

Table A-27. MREQ_NOTES view column descriptions (page 1 of 2)

Column Name	Data Type	Description
REQUEST_ID	NUMBER	Request internal identifier.
REQUEST_NUMBER	VARCHAR2(30)	Request number.
AUTHOR_ID	NUMBER	Author of the note (a PPM Center user).
AUTHOR_USERNAME	VARCHAR2(200)	Author of the note (a PPM Center user).
AUTHOR_FULL_NAME	VARCHAR2(80)	Author of the note (a PPM Center user).
AUTHORED DATE	DATE(7)	Date the note was authored.
NOTE_CONTEXT_ VALUE	VARCHAR2(30)	Any context information for the note entry.
NOTE_CONTEXT_ VISIBLE_VALUE	VARCHAR2(80)	Any visible context information for the note entry.
NOTE_TYPE_CODE	VARCHAR2(17)	User-entered or system-generated note.
NOTE_DATA	VARCHAR2(4000)	Note entry contents.

Table A-27. MREQ_NOTES view column descriptions (page 2 of 2)

Column Name	Data Type	Description
USER_NOTE_DATA	CLOB(4000)	User-entered note.
OLD_COLUMN_VALUE	VARCHAR2(1800)	Old value for a field-change note entry.
OLD_VISIBLE_ COLUMN_VALUE	VARCHAR2(1800)	Old visible value for a field-change note entry.
NEW_COLUMN_VALUE	VARCHAR2(1800)	New value for a field-change note entry.
NEW_VISIBLE_ COLUMN_VALUE	VARCHAR2(1800)	New visible value for a field-change note entry.
COLUMN_PROMPT	VARCHAR2(80)	Column prompt.

MREQ_OPENED_CLOSED_BY_DETAIL_D/M

These views provide information for request submission and completion activity, broken down by day (month) and by combinations of request type, application, department, priority, and assigned-to user.

Usage

Use the Meta Layer view to assess daily request throughput. It can also help indicate trends in open requests over time. Use this view instead of the simpler view MREQ_OPENED_CLOSED_BY_TYPE_D (MREQ_OPENED_CLOSED_BY_TYPE_M) to report on request throughput for a specific application, department, priority, or assigned user, allowing access to more granular summary information.

Results from a query of this view contain records only for days (months) on which there were requests opened and/or closed.

Sample Query

Consider using this view to create a report to examine throughput of all work order request types for the IT development department:

```
SELECT activity_date,
       application,
       priority,
       total_opened,
       total_closed,
       num_still_open,
       avg_comp_time_opened,
       avg_comp_time_closed
FROM   mreq_opened_closed_by_detail_d
WHERE  activity_date BETWEEN '01-APR-01' AND '05-APR-01'
AND    request_type_name = 'Work Order'
AND    department = 'Development'
ORDER BY activity_date;
```

To get a breakdown by month, replace `activity_date` with `activity_month` and `mreq_opened_closed_by_detail_d` with `mreq_opened_closed_by_detail_m` in the query above.

Results

```
                                                         Avg     Avg
                                                         Num    Comp    Comp
                                         Total   Total Still    Time    Time
Date                Application Priority   Open  Closed  Open    Open  Closed
--------- -------------------- -------- ------ ------- ----- ------- -------
01-APR-01 Manufacturing        Normal        0       2     0           26.06
01-APR-01 Financials           Normal        0       2     0           31.07
01-APR-01 Work-in-process      Normal        0       2     0           22.74
02-APR-01 Documentation        Normal        0       1     0           21.78
03-APR-01 Bill-of-materials    Low           0       1     0           41.01
03-APR-01 Bill-of-materials    Normal        0       1     0           26.09
04-APR-01 Bill-of-materials    Low           0       1     0           47.35
04-APR-01 Bill-of-materials    Normal        0       2     0           20.60
04-APR-01 Configuration        Normal        0       1     0           63.18
04-APR-01 Workflow             Low           0       2     0           20.70
05-APR-01 Manufacturing        Low           0       2     0           36.90
05-APR-01 Work-in-process      Normal        0       1     0            4.35
```

Table A-28. MREQ_OPEN_CLOSED_BY_DETAIL_D view column descriptions

Column Name	Data Type	Description
REQUEST_TYPE_NAME	VARCHAR2(80)	Name of request type.
DEPARTMENT	VARCHAR2(80)	Value of request Department field.
APPLICATION	VARCHAR2(80)	Value of request Application field.
PRIORITY	VARCHAR2(80)	Value of request Priority field.
ASSIGNED_TO	VARCHAR2(80)	User assigned to the request.
ACTIVITY_DATE	DATE	Date of activity.
TOTAL_OPENED	NUMBER	Number of requests opened on ACTIVITY_DATE.
TOTAL_CLOSED	NUMBER	Number of requests closed on ACTIVITY_DATE.
NET_CHANGE	NUMBER	TOTAL_OPENED - TOTAL_CLOSED.
NUM_STILL_OPEN	NUMBER	Number of requests opened on ACTIVITY_DATE that are still open.
AVG_COMP_TIME_OPENED	NUMBER	Average time to complete requests that were opened on ACTIVITY_DATE.
AVG_COMP_TIME_CLOSED	NUMBER	Average time to complete requests that were closed on ACTIVITY_DATE.

Table A-29. MREQ_OPEN_CLOSED_BY_DETAIL_M view column descriptions

Column Name	Data Type	Description
REQUEST_TYPE_NAME	VARCHAR2(80)	Name of request type.
DEPARTMENT	VARCHAR2(80)	Value of request Department field.
APPLICATION	VARCHAR2(80)	Value of request Application field.
PRIORITY	VARCHAR2(80)	Value of request Priority field.
ACTIVITY_MONTH	DATE	Month of activity.
TOTAL_OPENED	NUMBER	Number of requests opened during ACTIVITY_MONTH.
TOTAL_CLOSED	NUMBER	Number of requests closed during ACTIVITY_MONTH.
NET_CHANGE	NUMBER	TOTAL_OPENED - TOTAL_ CLOSED.
NUM_STILL_OPEN	NUMBER	Number of requests opened during ACTIVITY_MONTH that are still open.
AVG_COMP_TIME_ OPENED	NUMBER	Average time to complete requests that were opened during ACTIVITY_MONTH.
AVG_COMP_TIME_ CLOSED	NUMBER	Average time to complete requests that were closed during ACTIVITY_MONTH.

MREQ_OPENED_CLOSED_BY_TYPE_D/M

The Reporting Meta Layer views MREQ_OPENED_CLOSED_BY_TYPE_D (MREQ_OPENED_CLOSED_BY_TYPE M) provide summary information for request submission and completion activity, broken down by request type and by calendar day (month).

Usage

They can be used to assess daily (monthly) request throughput, and can help indicate trends in request processing over time.

Besides just the number or requests which were opened or closed in a given day (month), these views also contains columns like the net change in number of open requests during that day (month), the number of requests still open at the end of the day (month), and the average time to completion, in days (months), for requests opened in that day (month) and which have already been closed.

Results from a query of this view contain records only for days (months) in which there were requests opened and/or closed.

Sample Query

The following SQL query can be used as a basis for a report that summarizes all request submission and completion activity, per month, over a range of dates:

```
SELECT *
FROM   mreq_opened_closed_by_type_m
WHERE  activity_month BETWEEN '01-MAR-01' AND '01-APR-01'
ORDER BY activity_month;
```

To get a breakdown by day, replace `activity_month` and `mreq_opened_closed_by_type_m` with `activity_date` and `mreq_opened_closed_by_type_d` (respectively) in the query above.

Results

```
                                                                         Avg     Avg
                                                                 Num    Comp    Comp
                                           Tot   Total     Net Still    Time    Time
REQUEST_TYPE_NAME              Month      Open  Closed  Change  Open    Open  Closed
------------------------------ --------- ----- ------- ------- ----- ------- -------
Customer Access                01-MAR-01    53      52       1     0     .07     .01
HR Job Requisition             01-MAR-01    16      17      -1     6   38.84   48.93
HR New Hire Process            01-MAR-01    13      10       3     1   40.35   25.61
Product Bug                    01-MAR-01    83     232    -149    60    7.64  299.71
Product Patch                  01-MAR-01     8       0       8     8
Purchase Request               01-MAR-01    18      24      -6     0   13.72    9.13
Services Work Order            01-MAR-01    17       3      14    17     .00   33.59
Training Approval Request      01-MAR-01   336     369     -33     9    8.46   19.59
Vacation Request               01-MAR-01   115      72      43    33   25.87   27.84
Customer Access                01-APR-01    15       6       9    12   11.63  150.55
HR Job Requisition             01-APR-01     5       6      -1     0     .81   36.94
HR New Hire Process            01-APR-01    27       6      21    27    9.87  255.96
Product Bug                    01-APR-01    36      35       1     2     .21     .29
…
```

For more detailed request information filtered by common request header fields like Application, Department, Priority, and Assigned-to User, use the detail summary views MREQ_OPENED_CLOSED_BY_DETAIL_D and MREQ_OPENED_CLOSED_BY_DETAIL_M.

Table A-30. MREQ_OPEN_CLOSED_BY_TYPE_D view column descriptions

Column Name	Data Type	Description
REQUEST_TYPE_NAME	VARCHAR2(80)	Name of request type.
ACTIVITY_DATE	DATE	Date of activity.
TOTAL_OPENED	NUMBER	Number of requests opened on ACTIVITY_DATE.
TOTAL_CLOSED	NUMBER	Number of requests closed on ACTIVITY_DATE.
NET_CHANGE	NUMBER	TOTAL_OPENED - TOTAL_ CLOSED.
NUM_STILL_OPEN	NUMBER	Number of requests opened on ACTIVITY_DATE that are still open.
AVG_COMP_TIME_ OPENED	NUMBER	Average time to complete requests that were opened on ACTIVITY_DATE.
AVG_COMP_TIME_ CLOSED	NUMBER	Average time to complete requests that were closed on ACTIVITY_DATE.

Table A-31. MREQ_OPEN_CLOSED_BY_TYPE_M view column descriptions

Column Name	Data Type	Description
REQUEST_TYPE_NAME	VARCHAR2(80)	Name of request type.
ACTIVITY_MONTH	DATE	Month of activity.
TOTAL_OPENED	NUMBER	Number of requests opened during ACTIVITY_MONTH.
TOTAL_CLOSED	NUMBER	Number of requests closed during ACTIVITY_MONTH.
NET_CHANGE	NUMBER	TOTAL_OPENED - TOTAL_CLOSED.
NUM_STILL_OPEN	NUMBER	Number of requests opened during ACTIVITY_MONTH that are still open.
AVG_COMP_TIME_OPENED	NUMBER	Average time to complete requests that were opened during ACTIVITY_MONTH.
AVG_COMP_TIME_CLOSED	NUMBER	Average time to complete requests that were closed during ACTIVITY_MONTH.

MREQ_PENDING_REQUESTS

Used to create a report that shows the volume of open requests for any given request type in Demand Management.

Usage

This report can be used to get information about ongoing request processing work. It shows a summary of requests currently open for a specific Demand Management request type (for example, total number or average age), as well as information showing how many requests have been opened and closed in the current week and current month.

MREQ_PENDING_REQUESTS is aggregated across all requests.

In addition to overall totals of open requests, this view breaks down the information by priority (using the Priority header field). This is done because priority is usually the most important breakdown of load information. Data is grouped into three priority groupings—P1, P2, and P3, which map to the three highest-priority levels defined.

Sample Query

Suppose a QA manager has three types of requests to handle, running through three separate processes. The manager needs a report that will show current work volume for each of these request types, to help prioritize work and identify bottlenecks.

If the three request types are named MFG bug report, FIN bug report, and APPS enhancement request, the following SQL query can be used as a basis for a report to display the desired information:

```
SELECT request_type           Request_Type,
       open_requests          Open_Reqs,
       avg_age_open_requests  Avg Age,
       p1_open_requests       P1 Open Reqs,
       p2_open_requests       P2 Open Reqs
FROM   mreq_pending_requests
WHERE  process_name IN
       ('MFG bug report',
       'FIN bug report',
       'APPS enhancement request');
```

Results

```
                                              P1    P2
                           Open  Avg   Open  Open
REQUEST_TYPE               Reqs  Age   Reqs  Reqs
-------------------------  ----- ----  ----- -----
MFG bug report                98    3     21    77
FIN bug report                39    4     14    25
APPS enhancement request     140   12      8   132
```

This view ignores requests that have not been submitted.

Table A-32. MREQ_PENDING_REQUESTS view column descriptions (page 1 of 2)

Column Name	Data Type	Description
REQUEST_TYPE	VARCHAR2(80)	Target request type.
REQUEST_TYPE_ID	NUMBER	Internal ID of the request type.
REQUEST_TYPE_ DESCRIPTION	VARCHAR2(240)	Description of the request type.
OPEN_REQUESTS	NUMBER	Number of open requests for this request type.
AVG_AGE_OPEN_ REQUESTS	NUMBER	Average age of open requests.
MAX_AGE_OPEN_ REQUESTS	NUMBER	Age of oldest open request.
P1_OPEN_REQUESTS	NUMBER	Number of open P1 requests.
P2_OPEN_REQUESTS	NUMBER	Number of open P2 requests.
P3_OPEN_REQUESTS	NUMBER	Number of open P3 requests.
P1_AVG_AGE_OPEN_ REQUESTS	NUMBER	Average age of open P1 requests.
P2_AVG_AGE_OPEN_ REQUESTS	NUMBER	Average age of open P2 requests.
P3_AVG_AGE_OPEN_ REQUESTS	NUMBER	Average age of open P3 requests.
P1_MAX_AGE_OPEN_ REQUESTS	NUMBER	Age of oldest open P1 request.
P2_MAX_AGE_OPEN_ REQUESTS	NUMBER	Age of oldest open P2 request.
P3_MAX_AGE_OPEN_ REQUESTS	NUMBER	Age of oldest open P3 request.

Table A-32. MREQ_PENDING_REQUESTS view column descriptions (page 2 of 2)

Column Name	Data Type	Description
REQS_OPENED_THIS_ MONTH	NUMBER	Number of requests opened this month.
REQS_OPENED_THIS_ WEEK	NUMBER	Number of requests opened this week.
REQS_CLOSED_THIS_ MONTH	NUMBER	Number of requests closed this month.
REQS_CLOSED_THIS_ WEEK	NUMBER	Number of requests closed this week.

MREQ_REQUESTS

The most general view into request transaction data.

A blind query (`SELECT * FROM mreq_requests;`) will return one row for each request present in the system, including closed request.

The view columns map to the request fields that are common to all request types (for example, priority, department, application, assigned-to user, and contact information). There are also columns for the status of a request and the dates on which it was submitted, closed, and/or cancelled.

Since global request user data fields are present on all requests, there is also a view column for each global request user data field that is defined. The column name for each global request user data field is the same as the token name for that field.

Context-sensitive request user data sets have their own views—see MREQ_UD_<*Context Value*>.

Usage

Use this view when writing a report to present general request information without respect to a particular request type.

To build reports that make use of custom detail fields of a particular request type, the request type-specific views are more appropriate—see MREQ_<*Request Type Name*>.

Sample Query 1

To get information about the number of open requests in the system and to whom they are assigned:

```
SELECT assigned_to_username ASSIGNED_USER,
       COUNT(*) NUM_OPEN
FROM   mreq_requests
WHERE  close_date IS NULL
AND    cancel_date IS NULL
AND    submission_date IS NOT NULL
GROUP BY assigned_to_username
ORDER BY 1;
```

Results 1

```
ASSIGNED_USER                  NUM_OPEN
----------------------------------------
…
rfrazier                              13
rjeffries                              1
rjones                                28
rnelson                                9
rsmith                                 3
…
```

Sample Query 2

Or consider a similar query with the results grouped by the request type, to see how many requests of each type are open:

```
SELECT request_type_name REQUEST_TYPE,
       COUNT(*) NUM_OPEN
FROM   mreq_requests
WHERE  close_date IS NULL
AND    cancel_date IS NULL
AND    submission_date IS NOT NULL
GROUP BY request_type_name
ORDER BY 1;
```

Results 2

```
REQUEST_TYPE                       NUM_OPEN
--------------------------------------------
HR Job Requisition                        37
HR New Hire Process                       11
Press Release                              3
Product Patch                             33
Purchase Request                          11
Services Work Order                       81
Training Approval Request                115
Vacation Request                          56
```

Sample Query 3

Consider the case where a global request user data field has been defined to capture the username of a backup user responsible for each request.

The token name for this field is BACKUP_USERNAME. Therefore, in this view there would be a column named BACKUP_USERNAME:

```
SQL> desc mreq_requests
```

Results 3

```
 Name                              Null?    Type
 -------------------------------- -------- ----
 REQUEST_ID                        NOT NULL NUMBER
 REQUEST_DESCRIPTION               NOT NULL VARCHAR2(240)
 SUBMISSION_DATE                   NOT NULL DATE
 REQUEST_STATUS                    NOT NULL VARCHAR2(80)
…
 CANCEL_DATE                       NOT NULL DATE
 BACKUP_USERNAME                            VARCHAR2(200)
 REQUEST_TYPE_NAME                          VARCHAR2(80)
 REQUEST_SUBTYPE_NAME                       VARCHAR2(80)
…
```

Sample Query 4

The new column can be used to drive a report, if necessary. For example, to report on requests that have been open for more than five days and assigned to a particular backup user:

```
SELECT backup_username BACKUP_USER,
       assigned_to_username ASSIGNED_USER,
       COUNT(*) NUM_OLD_REQS
FROM   mreq_requests
WHERE  backup_username = '<ValidUsername>'
AND    close_date IS NULL
AND    cancel_date IS NULL
AND    submission_date IS NOT NULL
AND    (sysdate - submission_date) > 5
GROUP BY backup_username, assigned_to_username
ORDER BY 1, 2;
```

This query also displays the original user to which the request was assigned.

Table A-33. MREQ_REQUESTS view column descriptions (page 1 of 3)

Column Name	Data Type	Description
REQUEST_ID	NUMBER	Name of the request record.
REQUEST_ DESCRIPTION	VARCHAR2(240)	Request description.
SUBMISSION_DATE	DATE	Date request was submitted.
REQUEST_STATUS	VARCHAR2(80)	Current status of the request.
WORKFLOW	VARCHAR2(80)	Name of workflow used by this request.
DEPARTMENT	VARCHAR2(80)	Visible value of request department.
DEPARTMENT_CODE	VARCHAR2(30)	Internal code for department.

Table A-33. MREQ_REQUESTS view column descriptions (page 2 of 3)

Column Name	Data Type	Description
PRIORITY	VARCHAR2(80)	Visible value of request priority.
PRIORITY_CODE	VARCHAR2(30)	Internal code for priority.
APPLICATION	VARCHAR2(80)	Visible value of request application.
APPLICATION_CODE	VARCHAR2(30)	Internal code for application.
REQUEST_GROUP	VARCHAR2(80)	Visible value of request group.
REQUEST_GROUP_ CODE	VARCHAR2(30)	Internal code for request group.
ASSIGNED_TO_ USERNAME	VARCHAR2(200)	Username of PPM Center user to which this request is assigned.
ASSIGNED_TO_GROUP	VARCHAR2(80)	Name of security group to which this request is assigned.
CONTACT_FULL_NAME	VARCHAR2(80)	Full name of request contact.
CONTACT_LAST_NAME	VARCHAR2(30)	Last name of request contact.
CONTACT_FIRST_ NAME	VARCHAR2(30)	First name of request contact.
CONTACT_PHONE	VARCHAR2(30)	Phone number of request contact.
CONTACT_EMAIL	VARCHAR2(80)	Email address of request contact.
COMPANY	VARHCAR2(80)	Company of request contact.
CLOSE_RESULT	VARCHAR2(80)	If this request is closed, this is the visible result value.
CLOSE_DATE	DATE	If this request is closed, this is the date on which it was closed.
CANCEL_DATE	DATE	If this request is cancelled, this is the date on which it was cancelled.
REQUEST_TYPE	VARCHAR2(80)	Name of the request type.
REQUEST_SUBTYPE	VARCHAR2(80)	Name of the request subtype.

Table A-33. MREQ_REQUESTS view column descriptions (page 3 of 3)

Column Name	Data Type	Description
REQUEST_HEADER_ TYPE	VARCHAR2(80)	Name of the request header type.
CREATION_DATE	DATE	Date the request record was created.
CREATED_BY_ USERNAME	VARCHAR2(200)	Username of PPM Center user who created the request.
LAST_UPDATE_DATE	DATE	Date the request record was last updated.
CONTACT_ID	NUMBER	Internal identifier for contact.
REQUEST_TYPE_ID	NUMBER	Internal identifier for request type.
WORKFLOW_ID	NUMBER	Internal identifier for Workflow.
PERCENT_COMPLETE	NUMBER	Value for percent complete defined in the request’s workflow.

MREQ_REQUEST_ACTIONS

Used to gather information about all workflow actions for any given request in Demand Management.

Contains columns to display the result status of each step, how long it took to complete, details about the step (for example, source and destination environment), and other relevant details. It also adds the submission (Process Open) and completion (Process Close) of a request as pseudo workflow step actions, displaying the entire life cycle of the request in a single view.

Usage

This view can be used directly to view the full transaction history of a request, or it can be used as a basis for more complex reports showing, for example, throughput at specific request steps.

To relate information from this view with information from relevant requests, the report designer can use the request identifier REQUEST_ID to join with other standard views (for example, MREQ_REQUESTS, or a view for requests of a specific request type, MREQ_*<Request Type Name>*.

Sample Query 1

Consider a report that takes a request ID as input from the person running the report, and shows all transactions for that request. The report designer would probably want to include the name of the step, the date an action was taken, the result, and how long the step stayed active before the action was taken.

An SQL statement such as the following can accomplish this with MREQ_ REQUEST_ACTIONS:

```
SELECT action_name,
       action_date,
       action_result,
       duration
FROM   mreq_request_actions
WHERE  request_id = <DesiredID>
ORDER BY action_date;
```

Results 1

```
                               Action    Action
Process Step                   Date      Result            Duration
------------------------------ --------- ----------------- --------
Open                           26-APR-01 Released               .00
Check Priority                 26-APR-01 Normal                 .00
SA - Check Prodcut             26-APR-01 NULL result            .00
CL - Check issue assignment    26-APR-01 aaslani                .00
Work In Progress               15-MAY-01 Resolved             18.72
Feedback                       20-MAY-01 Timeout               5.00
Close                          20-MAY-01 Closed [Success]       .00
Request resolved               20-MAY-01 Succeeded              .00
```

Sample Query 2

Consider a work order request type that has a Customer field with token CUSTOMER.

The name of the corresponding request view will be MREQ_WORK_ORDER (based on the general view MREQ_<*Request Type Name*>). Suppose a report is needed to show all work order requests that are eligible for a particular PPM Center user (in this case, a user with username fjohnson) to act on, broken down by customer:

```
SELECT wo.customer                CUSTOMER,
       wo.request_id              REQ_NUM,
       ra.request_workflow_step_label || ': '|| ra.action_name
                                  ELIGIBLE_STEP,
       ra.duration                DAYS_ELIGIBLE
FROM   mreq_work_order wo,
       mwfl_step_security_users ssu,
       mreq_request_actions ra
WHERE  ra.status_type = 'ELIGIBLE'
AND    ssu.workflow_step_id = ra.workflow_step_id
AND    ssu.username = 'fjohnson'
AND    ra.request_id = wo.request_id
ORDER BY 1,2,3,4;
```

In this example, MREQ_REQUEST_ACTIONS was joined to the view MREQ_WORK_ORDER with the REQUEST_ID column.

Note the format of the ELIGIBLE_STEP column being selected, which will return a value like `12.3.1: Review by Lead` .

Note also the use of the Meta Layer view MWFL_STEP_SECURITY_USERS, which is used to determine if a specified user is authorized for a specified workflow step.

Additional considerations:

- The column STATUS is the status name that is displayed in the status tab of requests in the Demand Management application.

  The internal code STATUS_TYPE is provided to group these status names into logical groupings. For example, there may be many different statuses that all represent a COMPLETE status type (the result value of any workflow step—Approved, Succeeded, Rejected, Failed QA Test). While STATUS may have many different possible values, STATUS_TYPE has only the following possible values:

  - SUBMITTED
  - IN_PROGRESS

  - CLOSED_SUCCESS
  - ELIGIBLE
  - ERROR
  - CLOSED_FAILURE
  - PENDING
  - COMPLETE
  - CANCELLED

- The internal code STEP_TRANSACTION_ID is provided in this view for use with the Meta Layer view MWFL_TRANSITIONS, which can be used to get detailed information about previous or subsequent process steps.

Table A-34. MREQ_REQUEST_ACTIONS view column descriptions (page 1 of 3)

Column Name	Data Type	Description
REQUEST_ID	NUMBER	The request ID.
REQUEST_DESCRIPTION	VARCHAR2(240)	Description of the request.
REQUEST_TYPE	VARCHAR2(80)	Request type of this request.
REQUEST_WORKFLOW	VARCHAR2(80)	Top-level workflow used by this request.
REQUEST_WORKFLOW_STEP_LABEL	VARCHAR2(2000)	Visible label of this step in request status tab.
ACTION_NAME	VARCHAR2(80)	Name of workflow step action.
WORKFLOW	VARCHAR2(80)	Name of workflow that contains this step.
WORKFLOW_STEP_NUMBER	NUMBER	Workflow sequence number of this step.
STATUS	VARCHAR2(200)	Visible status of this request.
STATUS_TYPE	VARCHAR2(30)	Internal code for STATUS.
ELIGIBLE_DATE	DATE	Date this step became eligible.
ACTION_DATE	DATE	Date action was taken on this step.

Table A-34. MREQ_REQUEST_ACTIONS view column descriptions (page 2 of 3)

Column Name	Data Type	Description
ACTION_RESULT	VARCHAR2(200)	Result of the action.
ERROR_MESSAGE	VARCHAR2(240)	(If STATUS_TYPE = 'ERROR') error message.
DURATION	NUMBER	Number of days at this status, or until completed (if STATUS_TYPE = 'COMPLETE').
SOURCE_ ENVIRONMENT	VARCHAR2(80)	Source environment (if applicable).
SOURCE_ ENVIRONMENT_ GROUP	VARCHAR2(100)	Source environment group (if applicable).
DEST_ENVIRONMENT	VARCHAR2(80)	Destination environment (if applicable).
DEST_ENVIRONMENT_ GROUP	VARCHAR2(100)	Destination environment group (if applicable).
USER_COMMENT	VARCHAR2(240)	User comment entered when taking action on this step.
CHILD_REQUEST_ID	NUMBER	ID of child request.
CHILD_PACKAGE_ID	NUMBER	Internal ID of child package.
STEP_TRANSACTION_ ID	NUMBER	Internal ID of this transaction.
REQUEST_ WORKFLOW_ID	NUMBER	internal ID for top-level workflow used by this request.
WORKFLOW_ID	NUMBER	Internal ID for the workflow that contains this Workflow step.
WORKFLOW_STEP_ID	NUMBER	Internal ID of this workflow step.
ACTION_RESULT_ CODE	VARCHAR2(200)	Internal code for ACTION_ RESULT.
SOURCE_ ENVIRONMENT_ID	NUMBER	Internal ID of source environment.

Table A-34. MREQ_REQUEST_ACTIONS view column descriptions (page 3 of 3)

Column Name	Data Type	Description
SOURCE_ ENVIRONMENT_ GROUP_ID	NUMBER	Internal ID of source environment group.
DEST_ENVIRONMENT_ ID	NUMBER	Internal ID of destination environment.
DEST_ENVIRONMENT_ GROUP_ID	NUMBER	Internal ID of destination environment group.

MREQ_REFERENCES

Used to view the references of requests in Demand Management.

There are several types of references for requests. If a request is part of a release, then there will be a reference for that release. If a request is a parent or child of another request, then there will be a reference for that request. References are also used to attach documents to a request.

Usage

The RELATIONSHIP column in MREQ_REFERENCES describes the relationship of the referenced item to the request that references it. This view also has columns for each of the entities that can be referenced to a request—other requests, packages, projects, tasks, releases, attachments, and URLs.

For each record in MREQ_REFERENCES, only one of these columns will have a value and the others will be NULL.

Sample Query

The following SQL can be used to retrieve a list of all references to a particular request:

```
SELECT referenced_package_id PKG,
       referenced_project_id PROJ,
       referenced_request_id REQ,
       referenced_release_id REL,
       referenced_task_id    TASK,
       attachment_name       ATTACHMENT,
       document_url          URL,
       relationship          RELATIONSHIP
FROM   mreq_references
WHERE  request_number = '54872';
```

Results

```
  PKG  PROJ   REQ   REL  TASK ATTACHMENT         URL               RELATIONSHIP
----- ----- ----- ----- ----- ------------------ ----------------- ----------------------
                        43301                                      Contains this Request
                        43304                                      Contains this Request
30043                                                              Child of this Request
            52383                                                  Parent of this Request
                              screenShot.doc
```

Table A-35. MREQ_REFERENCES view column descriptions

Column Name	Data Type	Description
REQUEST_NUMBER	VARCHAR2(30)	Number of the target request.
RELATIONSHIP	VARCHAR2(30)	Relationship of reference to this request.
REFERENCED_ PACKAGE_ID	NUMBER	ID of referenced package.
REFERENCED_ PROJECT_ID	NUMBER	ID of referenced project.
REFERENCED_ REQUEST_ID	NUMBER	ID of referenced request.
REFERENCED_ RELEASE_ID	NUMBER	ID of referenced release.
REFERENCED_TASK_ ID	NUMBER	ID of referenced task.
OVERRIDE_CODE	NUMBER	Code to manually override the dependency behavior of the reference.
ATTACHMENT_NAME	VARCHAR2(200)	Name of attached document.
DOCUMENT_URL	VARCHAR2(200)	URL of referenced document on the Web.
REQUEST_ID	NUMBER	Internal ID of this request.

MREQ_REQUEST_HEADER_TYPES

Accesses configuration details of request header types in Demand Management.

Usage

In some cases a report designer might need to include request header type information in a report, and can join the REQUEST_HEADER_TYPE column in this view with the same column in the MREQ_REQUEST_TYPES view, and in general request views (MREQ_REQUESTS and MREQ_<*Request Type Name*>).

PPM Center supports user data on request header types. All defined request header type user data fields are represented in MREQ_REQUEST_HEADER_TYPES view; there is a column for each request header type user data field.

The column name for each request header type user data field is the same as the token name for that field.

Sample Query

Consider a scenario in which a user data field with token name OWNER is defined for request header types, perhaps to keep track of a PPM Center administrator responsible for maintaining each request header type configuration.

A corresponding view column named OWNER will be present in MREQ_REQUEST_HEADER_TYPES view:

```
SQL> desc mreq_request_header_types
```

Results

```
 Name                            Null?    Type
 ------------------------------- -------- ----
 REQUEST_HEADER_TYPE             NOT NULL VARCHAR2(80)
 REQUEST_HEADER_TYPE_DESC                 VARCHAR2(240)
…
 ACCELERATOR_NAME                NOT NULL VARCHAR2(80)
 OWNER                                    VARCHAR2(200)
 CREATION_DATE                   NOT NULL DATE
 CREATED_BY_USERNAME             NOT NULL VARCHAR2(30)
 LAST_UPDATE_DATE                NOT NULL DATE
```

Table A-36. MREQ_REQUEST_HEADER_TYPES view column descriptions

Column Name	Data Type	Description
REQUEST_HEADER_TYPE	VARCHAR2(80)	Name of request header type.
REQUEST_HEADER_TYPE_DESC	VARCHAR2(240)	Description of request header type.
REFERENCE_FLAG	VARCHAR2(1)	Is this a reference request header type? (Y/N).
ENABLED_FLAG	VARCHAR2(1)	Is this request header type enabled? (Y/N).
ACCELERATOR_NAME	VARCHAR2(80)	Name of parent extension.
Request Header Type Global User Data fields	VARCHAR2(200)	One column for each request header type global user data field. Column name is the user data field token name.
CREATION_DATE	DATE	Creation date of this request header type.
CREATED_BY_USERNAME	VARCHAR2(200)	Username of PPM Center user that created this request header type.
LAST_UPDATE_DATE	DATE	Last update date of this request header type.
REQUEST_HEADER_TYPE_ID	NUMBER	Internal identifier for the request header type record.

MREQ_REQUEST_TYPES

Accesses configuration details of request types in Demand Management.

Usage

In some cases a report designer might need to include request type information in a report, and can join the REQUEST_TYPE column in this view with the same column in the general request views (MREQ_ REQUESTS and MREQ_ <*Request Type Name*>).

PPM Center supports user data on request types. All defined request type user data fields are represented in MREQ_REQUEST_TYPES view; there is a column for each request type user data field. The column name for each request type user data field is the same as the token name for that field.

Sample Query 1

Consider a scenario in which a user data field with token name OWNER is defined for request types, perhaps to keep track of a PPM Center administrator responsible for maintaining each request type configuration.

A corresponding view column named OWNER will be present in MREQ_ REQUEST_TYPES view:

```
SQL> desc mreq_request_types
```

Results 1

```
 Name                               Null?    Type
 ---------------------------------- -------- ----
 REQUEST_TYPE                       NOT NULL VARCHAR2(30)
 REQUEST_TYPE_DESCRIPTION           NOT NULL VARCHAR2(240)
…
 INITIAL_STATUS                     NOT NULL VARCHAR2(80)
 RESTRICTION                        NOT NULL VARCHAR2(30)
 OWNER                                       VARCHAR2(200)
 CREATION_DATE                      NOT NULL DATE
 CREATED_BY_USERNAME                NOT NULL VARCHAR2(30)
…
```

Sample Query 2

A SQL query based on this view might be used to determine how many requests were created prior to a configuration change for a particular request type.

For example, suppose a request type named Work Order has undergone a significant configuration change, which might invalid open work order requests that were created before the change. Therefore, a report is needed to determine the status of open work order requests that were created before the changes, which might be based on the following SQL example:

```
SELECT wo.request_id          REQUEST_NUM,
       wo.request_status      CURRENT_STATUS,
       wo.request_description DESCRIPTION
FROM   mreq_work_order wo,
       mreq_request_types rt
WHERE  wo.creation_date < rt.last_update_date
AND    rt.request_type = 'Work Order'
ORDER BY 1;
```

Notice that we do not have to join the explicit request type name to the view MREQ_WORK_ORDER, because it is already implicit in the view definition—only work order requests are returned from that view.

Table A-37. MREQ_REQUEST_TYPES view column descriptions (page 1 of 2)

Column Name	Data Type	Description
REQUEST_TYPE	VARCHAR2(80)	Name of request type.
REQUEST_TYPE_ DESCRIPTION	VARCHAR2(240)	Description of request type.
REFERENCE_FLAG	VARCHAR2(1)	Is this a reference request type? (Y/N).
ENABLED_FLAG	VARCHAR2(1)	Is this request type enabled? (Y/N).
REQUEST_HEADER_ TYPE	VARCHAR2(80)	Name of request header type.
ACCELERATOR_NAME	VARCHAR2(80)	Name of parent extension.
CRT_WORKBENCH_ ONLY_FLAG	VARCHAR2(1)	Is this request type available only to the PPM Workbench interface? (Y/N).
INITIAL_STATUS	VARCHAR2(80)	Initial status upon submission of requests of this request type.
RESTRICTION	VARCHAR2(30)	Either PARTICIPANT or UNRESTRICTED.
Request Type Global User Data fields	VARCHAR2(200)	One column for each request type global user data field. Column name is the user data field token name.
CREATION_DATE	DATE	Creation date of this request type.

Table A-37. MREQ_REQUEST_TYPES view column descriptions (page 2 of 2)

Column Name	Data Type	Description
CREATED_BY_ USERNAME	VARCHAR2(200)	Username of PPM Center user that created this request type.
LAST_UPDATE_DATE	DATE	Last update date of this request type.
REQUEST_TYPE_ID	NUMBER	Internal identifier for the request type record.

MREQ_TABLE_COMPONENT

Contains table component data for request detail fields with validations.

Table A-38. MREQ_TABLE_COMPONENT view column descriptions

Column Name	Data Type	Description
VALIDATION_ID	NUMBER	ID of the table component validation.
VALIDATION_NAME	VARCHAR2(80)	Name of the table component validation.
DESCRIPTION	VARCHAR2(240)	Description of the table component validation.
REQUEST_ID	NUMBER	Request ID of the request where this table component is being used.
TABLE_COLUMNS		View column for each column defined on the table component.
PARAMETER_TOKEN	VARCHAR2(30)	Token defined in the request type fields defined for table components.
ROW_SEQUENCE_NUMBER	NUMBER	Control sequence used to identify the order of the current row in the table field.

Other Views

Other views provide information about PPM Center entities like workflows and security groups. For example, MWFL_STEP_SECURITY_USERS lists all users with authority to act on a given workflow step through static security group or user linkage, as defined in the workflow step window in the Workflow workbench.

MWFL_STEP_SECURITY_GROUPS and MWFL_STEP_SECURITY_USERS)

Usage

Used to get information about PPM Center users or security groups linked to Workflow steps.

The view MWFL_STEP_SECURITY_USERS lists all users with authority to act on a given workflow step through static security group or user linkage, as defined in the workflow step dialog in the Workflow Workbench.

The view MWFL_STEP_SECURITY_GROUPS lists all security groups with authority to act on a step through static security group linkage.

These views can be useful for reporting on specific key workflow steps to show more detailed information that may not be available in the more general activity management views.

Sample Query

Consider a scenario in which a report is needed to show all requests in Demand Management for which a given user is eligible for one or more approval workflow steps. The view MWFL_WORKFLOW_STEPS can be used to show which workflow steps are approval steps, and the view MREQ_REQUEST_ACTIONS will provide the request information for eligible steps:

```
SELECT ssu.username                ELIGIBLE_USER,
       ra.request_id               REQUEST_NUM,
       ra.request_workflow_step_label || ': '|| ra.action_name
                                   ELIGIBLE_STEP,
       ra.duration                 DAYS_ELIGIBLE
FROM   mwfl_step_security_users ssu,
       mwfl_workflow_steps ws,
       mreq_request_actions ra
WHERE  ra.status_type = 'ELIGIBLE'
AND    ws.step_type = 'Approval'
AND    ra.workflow_step_id = ws.workflow_step_id
AND    ssu.workflow_step_id = ra.workflow_step_id
ORDER BY 1,2,3,4;
```

In this query, the workflow step identifier WORKFLOW_STEP_ID was used to join MWFL_STEP_SECURITY_USERS with the view MREQ_ REQUEST_ACTIONS, to relate request workflow step information.

Dynamic workflow step security defined by tokens is not included in these views.

Table A-39. MWFL_STEP_SECURITY_GROUPS view column descriptions

Column Name	Data Type	Description
WORKFLOW_STEP	VARCHAR2(80)	Name of workflow step.
WORKFLOW_STEP_ NUMBER	NUMBER	Workflow step sequence number.
WORKFLOW	VARCHAR2(80)	Name of workflow containing this step.
SECURITY_GROUP_ NAME	VARCHAR2(40)	Security group authorized to act on this step.
WORKFLOW_STEP_ID	NUMBER	Internal identifier of workflow step.
SECURITY_GROUP_ID	NUMBER	Internal identifier of security group.
WORKFLOW_ID	NUMBER	Internal identifier of workflow.

Table A-40. MWFL_STEP_SECURITY_USERS view column descriptions

Column Name	Data Type	Description
WORKFLOW_STEP_ID	NUMBER	Internal identifier of workflow step.
WORKFLOW_STEP	VARCHAR2(80)	Name of workflow step.
WORKFLOW_STEP_ NUMBER	NUMBER	Workflow step sequence number.
WORKFLOW	VARCHAR2(80)	Name of workflow containing this step.
USERNAME	VARCHAR2(200)	Username of PPM Center user authorized to act on this step.
USER_ID	NUMBER	Internal identifier of user.
WORKFLOW_ID	NUMBER	Internal identifier of workflow.

MWFL_WORKFLOWS

Accesses basic configuration details of workflows.

Usage

Report designers who need to include workflow information in a report can join the WORKFLOW_ID column in this view with the same column in Workflow transaction views (for example, MREQ_REQUEST_ACTIONS). The view MWFL_WORKFLOW has columns for the main workflow definition fields present on the first tab of the workflow detail window in the PPM Workbench, and also includes a column for each workflow user data field defined in the system.

Sample Query

If the system has three workflow user data fields defined, this view will contain three columns that use the user data fields' token names as view column names.

If these three user data fields have the tokens DEPARTMENT, ADMINSTRATOR_USERNAME, and WORKFLOW_MANAGER, then the MWFL_WORKFLOWS view would contain three columns with these names:

```
SQL> desc mwfl_workflows
```

Results

```
 Name                            Null?    Type
 ------------------------------- -------- ----
 WORKFLOW                        NOT NULL VARCHAR2(80)
 WORKFLOW_DESCRIPTION                     VARCHAR2(240)
…
 SUB_WORKFLOW_FLAG                        VARCHAR2(1)
 DEPARTMENT                               VARCHAR2(200)
 ADMINISTRATOR_USERNAME                   VARCHAR2(200)
 WORKFLOW_MANAGER                         VARCHAR2(200)
 CREATED_BY_USERNAME             NOT NULL VARCHAR2(30)
 CREATION_DATE                   NOT NULL DATE
…
```

By default this view returns both reference and non-reference workflows in the system.

PPM Center provides reference copies of some workflows, which are disabled and not usable by PPM Center transactions, and as such are rarely of reporting interest. The view column REFERENCE_FLAG can be used to filter results. To only show active, non-reference workflows while using MWFL_ WORKFLOWS view, include `REFERENCE_FLAG = 'N'` in the query.

Table A-41. MWFL_WORKFLOWS view column descriptions (page 1 of 2)

Column Name	Data Type	Description
WORKFLOW	VARCHAR2(80)	Workflow name.
WORKFLOW_DESCRIPTION	VARCHAR2(240)	Workflow description.
REFERENCE_FLAG	VARCHAR2(1)	Is this a reference workflow? (Y/N).
ENABLED_FLAG	VARCHAR2(1)	Is this workflow enabled? (Y/N).
PRODUCT_SCOPE	VARCHAR2(200)	PPM Center product scope of this workflow.
RESTRICT_OBJECT_FLAG	VARCHAR2(1)	Are new objects automatically restricted from this workflow? (Y/N).
FORCE_APP_CODES_FLAG	VARCHAR2(1)	(If PRODUCT_SCOPE = ‘Deployment Management’:) is app code choice required on package lines using this workflow? (Y/N).
RESTRICT_WORKFLOWS_FLAG	VARCHAR2(1)	Are new workflows automatically restricted from this workflow? (Y/N).
SUB_WORKFLOW_FLAG	VARCHAR2(1)	Is this workflow a subworkflow? (Y/N).
Workflow Global User Data	VARCHAR2(200)	One column for each workflow global user data field. Column name is the user data field token name.
CREATED_BY_USERNAME	VARCHAR2(200)	Username of PPM Center user who created this workflow.
CREATION_DATE	DATE	Date workflow was created.
LAST_UPDATE_DATE	DATE	Date workflow was last updated.

Table A-41. MWFL_WORKFLOWS view column descriptions (page 2 of 2)

Column Name	Data Type	Description
FIRST_WORKFLOW_STEP_ID	NUMBER	Internal identifier (workflow step ID) of the first step in this workflow.
REOPEN_STEP_ID	NUMBER	Internal identifier (workflow step ID) of the step that will be eligible if this workflow is re-opened.
WORKFLOW_ID	NUMBER	Internal identifier for this workflow.

MWFL_WORKFLOW_STEPS

Provides configuration details of workflow steps.

Usage

In some cases a report designer might need to present workflow step information in a report. The report designer can join this view with other workflow views through the key values WORKFLOW_STEP_ID and WORKFLOW_ID.

This view also includes a column for each workflow step user data field defined in the system.

Sample Query 1

Consider a scenario in which a user data field has been defined for workflow steps to provide a categorization. Suppose this field is called Step Category, has a token CATEGORY. Therefore, in this view there will be a CATEGORY column:

```
SQL> desc mwfl_workflow_steps
```

Results 1

```
 Name                              Null?    Type
 -------------------------------- -------- ----
 WORKFLOW_STEP                     NOT NULL VARCHAR2(80)
 WORKFLOW_STEP_NUMBER              NOT NULL NUMBER
…
 PARENT_REQUEST_TYPE_STATUS                 VARCHAR2(30)
 CATEGORY                                   VARCHAR2(200)
 CREATED_BY_USERNAME               NOT NULL VARCHAR2(30)
 CREATION_DATE                     NOT NULL DATE
…
```

This type of information can be used to drive reports built using the Meta Layer.

Sample Query 2

To continue the example, suppose the CATEGORY user data field has possible values like Normal, Test Gate, and Prod Gate to give an indication of the nature of each step. Suppose a report is needed to show if a particular PPM Center user (in this case a user with username fjohnson) is eligible for any Deployment Management workflow steps that are critical gateways to production (that is, in the Prod Gate category), and how long they have been eligible:

```
SELECT pla.package_number         PACKAGE_NUM,
       pla.line_number            LINE_NUM,
       pla.line_workflow_step_label || ': '|| pla.action_name
                                  ELIGIBLE_STEP,
       pla.duration               TIME_ELIGIBLE,
       ws.workflow                WORKFLOW
FROM   mwfl_step_security_users ssu,
       mwfl_workflow_steps ws,
       mpkgl_package_line_actions pla
WHERE  pla.status_type = 'ELIGIBLE'
AND    ws.category = 'Prod Gate'
AND    ws.workflow_step_id = pla.workflow_step_id
AND    ws.workflow_step_id = ssu.workflow_step_id
AND    ssu.username = 'fjohnson';
```

In this example, MWFL_WORKFLOW_STEPS was joined to the view MPKGL_PACKAGE_LINE_ACTIONS with the WORKFLOW_STEP_ID column.

Note the use of the Meta Layer view MWFL_STEP_SECURITY_USERS, which is used to determine if a specified user is authorized for a specified Workflow step.

Additional information:

- By default this view returns both reference and non-reference workflow steps in the system. PPM Center provides reference copies of some workflow steps, which are disabled and not usable by PPM Center transactions, and as such are rarely of reporting interest. The view column REFERENCE_FLAG can be used to filter results. To show only active, non-reference workflow steps while using the MWFL_WORKFLOW_STEPS view, include `REFERENCE_FLAG = 'N'` in the query.
- The type of each workflow step is accessible through the column STEP_TYPE. The following types of workflow steps are available:
  - Condition
  - Decision
  - Execution
  - Workflow

Table A-42. MWFL_WORKFLOW_STEPS view column descriptions (page 1 of 2)

Column Name	Data Type	Description
WORKFLOW_STEP	VARCHAR2(80)	Name of workflow step.
WORKFLOW_STEP_NUMBER	NUMBER	Workflow step sequence number.
STEP_TYPE	VARCHAR2(200)	Type of workflow step.
REFERENCE_FLAG	VARCHAR2(1)	Is this a reference workflow step? (Y/N).
ENABLED_FLAG	VARCHAR2(1)	Is this workflow step enabled? (Y/N).
STEP_SOURCE_NAME	VARCHAR2(50)	Name of source of this workflow step.
WORKFLOW	VARCHAR2(80)	Workflow name.
PRODUCT_SCOPE	VARCHAR2(200)	PPM Center product scope of workflow step.
PARENT_REQUEST_TYPE_STATUS	VARCHAR2(80)	(If PRODUCT_SCOPE = 'Demand Management':) status to set in parent request when at this workflow step.
PARENT_ASSIGNED_TO_USERNAME	VARCHAR2(200)	PPM Center user to assign to parent.
PARENT_ASSIGNED_TO_GROUP	VARCHAR2(40)	Security group to assign to parent.
Workflow Step Global User Data	VARCHAR2(200)	One column for each workflow step global user data field. Column name is the user data field token name.
CREATED_BY_USERNAME	VARCHAR2(200)	Username of PPM Center user who created this workflow step.
CREATION_DATE	DATE	Date workflow was created.
LAST_UPDATED_BY_USERNAME	VARCHAR2(30)	Username of PPM Center user who last updated this workflow step.
LAST_UPDATE_DATE	DATE	Date workflow step was last updated.

Table A-42. MWFL_WORKFLOW_STEPS view column descriptions (page 2 of 2)

Column Name	Data Type	Description
WORKFLOW_STEP_ID	NUMBER	Internal identifier for workflow step.
WORKFLOW_ID	NUMBER	Internal identifier for this workflow.
PERCENT_COMPLETE	NUMBER	Percent complete value defined for a workflow step.
PARENT_ASSIGNED_TO_USER_ID	NUMBER	ID for parent assigned-to PPM Center user.
PARENT_ASSIGNED_TO_GROUP_ID	NUMBER	ID for parent assigned-to security group.

KCRT_PARTICIPANT_CHECK_V

Used to enforce request participant security in the data presented in reports.

Usage

A query of KCRT_PARTICIPANT_CHECK_V will return the requests in Demand Management in which a particular PPM Center user is a participant. This view can be joined into report queries to check whether the user running the report is a participant of requests that are enforcing participant-only viewing restriction.

Sample Query

Consider a report that is to return the description of open requests in Demand Management. To restrict reported information to only those requests in which the user running the report is a participant, the so that it requires a valid username as an input field.

➤ The methodology and support for this type of report input will vary between reporting systems. Consult the documentation for the reporting system you are using for specific instructions.

Assuming the input username was available as REPORT_USER, you can include the following SQL fragment in the report query:

```
…
FROM   kcrt_participant_check_v kpc
WHERE  kpc.username = REPORT_USER
AND    kpc.request_id = …
…
```

Including this fragment in the full SQL statement might look as follows (with an example username of fjohnson):

```
SELECT r.request_id,
       r.request_status,
       r.request_description
FROM   mreq_requests r,
       kcrt_participant_check_v kpc
WHERE  r.request_status not in ('Cancelled','Closed')
AND    kpc.username = 'fjohnson'
AND    kpc.request_id = r.request_id;
```

If a request type does not enforce request participant security, then all requests of this request type will be returned by KCRT_PARTICIPANT_CHECK_V as viewable.

Table A-43. KCRT_PARTICIPANT_CHECK_V view column descriptions

Column Name	Data Type	Description
USERNAME	VARCHAR2(200)	Username of PPM Center user.
REQUEST_ID	NUMBER	Internal ID of the request that this user is a participant of.
USER_ID	NUMBER	Internal ID of the PPM Center user.
USERNAME	VARCHAR2(200)	Username of the PPM Center user.
REQUEST_ID	NUMBER	Internal ID of the request that this user is a participant of.
USER_ID	NUMBER	Internal ID of the PPM Center user.
USERNAME	VARCHAR2(200)	Username of PPM Center user.
REQUEST_ID	NUMBER	Internal ID of the request that this user is a participant of.

KDLV_PARTICIPANT_CHECK_V

Used to enforce package participant security in the data presented in reports.

Usage

A query of KDLV_PARTICIPANT_CHECK_V will return the packages in Deployment Management of which a particular user is a participant. This view can be joined into report queries to check whether the user running the report is a participant of packages that are enforcing participant-only viewing restriction.

Sample Query

As a simple example, consider a report that is to return the description of open packages in Deployment Management. To restrict reported information to only those packages that the user running the report is a participant of, you must design the report so that it requires a valid username as an input field.

➤ The methodology and support for this type of report input will vary between reporting systems. Consult the documentation for the reporting system you are using for specific instructions.

Assuming the input username is available as REPORT_USER, include the following SQL fragment in the report query:

```
…
FROM   kdlv_participant_check_v kpc
WHERE  kpc.username = REPORT_USER
AND    kpc.package_id = …
…
```

Including this fragment in the full SQL statement might look as follows (with an example username of fjohnson):

```
SELECT p.package_id,
       p.package_status
       p.package_description
FROM   mpkg_packages p,
       kdlv_participant_check_v kpc
WHERE  p.package_status not like 'Closed%'
AND    kpc.username = 'fjohnson'
AND    kpc.package_id = p.package_id;
```

If a Deployment Management workflow does not enforce package participant security, then all packages using this workflow will be returned by KDLV_ PARTICIPANT_CHECK_V as viewable.

Table A-44. KDLV_PARTICIPANT_CHECK_V view column descriptions

Column Name	Data Type	Description
USERNAME	VARCHAR2(200)	Username of PPM Center user.
REQUEST_ID	NUMBER	Internal ID of the request that this user is a participant of.
USER_ID	NUMBER	Internal ID of the PPM Center user.

KRML_CALENDAR_DAYS and KRML_CALENDAR_MONTHS

These tables, included in the RML, contain sequential dates. KRML_CALENDAR_DAYS contains a record for every day from January 1, 1998, to mid-2011. KRML_CALENDAR_MONTHS contains a record for every month from January 1998 to mid-2011.

Usage

These tables can be used to provide a date for organizing and grouping the results of queries.

Sample Query

Suppose that a report needs to contain summary information for the number of errors for step 2 in the FIN dev-test-prod workflow, broken down by month. The calendar table KRML_CALENDAR_MONTHS can be used to provide the month-by-month breakdown to join with the ACTIVITY_DATE column in the view MWFL_STEP_ACTIVITIES:

```
SELECT m.calendar_month MONTH,
       sum(sa.error)    NUM_ERRORS
FROM   krml_calendar_months m,
       mwfl_step_activities sa
WHERE  sa.workflow = 'FIN dev-test-prod'
AND    sa.workflow_step_number = 2
AND    sa.activity_date >= m.start_date
AND    sa.activity_date < m.end_date
GROUP BY m.calendar_month
ORDER BY 1;
```

Note the comparison of ACTIVITY_DATE to the START_DATE and END_DATE of the calendar month. This can be very useful for grouping discrete activity dates into aggregate time buckets.

Table A-45. KRML_CALENDAR_DAYS view column descriptions

Column Name	Data Type	Description
CALENDAR_DATE	DATE	A calendar date.
CALENDAR_MONTH	DATE	Date of the first day of the month containing the calendar date.
CALENDAR_YEAR	DATE	Date of the first day of the year containing the calendar date.

Table A-46. KRML_CALENDAR_MONTHS view column descriptions

Column Name	Data Type	Description
CALENDAR_MONTH	DATE	Date of the first day of a calendar month.
START_DATE	DATE	Date the calendar month started.
END_DATE	DATE	Date the calendar month ended.

Index