[Informatica Knowledge Base](http://informaticaknowledgebase.blogspot.com/)

# Informatica Repository Schema Queries

**PowerCenter query returns all PowerCenter mappings with update strategy transformations using repository views:**

SELECT DISTINCT(PARENT\_MAPPING\_NAME) AS MAPPING\_NAME, SUBJECT\_AREA AS FOLDER\_NAME FROM REP\_ALL\_MAPPINGS A, REP\_WIDGET\_INST B WHERE A.PARENT\_MAPPING\_ID = B.MAPPING\_ID AND A.PARENT\_SUBJECT\_ID=B.SUBJECT\_ID AND B.WIDGET\_TYPE=4;

**PowerCenter repository query that lists all sources (whether they are used in a mapping or not) and any mappings the source is used in a repository:**

SELECT REP\_ALL\_SOURCES.SOURCE\_NAME, REP\_SRC\_MAPPING.MAPPING\_NAME, REP\_ALL\_SOURCES.SOURCE\_DATABASE\_NAME FROM REP\_SRC\_MAPPING, REP\_ALL\_SOURCES WHERE REP\_SRC\_MAPPING.SOURCE\_ID(+) = REP\_ALL\_SOURCES.SOURCE\_ID and REP\_SRC\_MAPPING.SOURCE\_NAME(+) = REP\_ALL\_SOURCES.PARENT\_SOURCE\_NAME and REP\_SRC\_MAPPING.SUBJECT\_ID(+) = REP\_ALL\_SOURCES.SUBJECT\_ID

**The Sequence Generator stores values in the repository.**

**Following is a query for current value of all Sequence Generator within a repository:**

SELECT a.attr\_value AS current\_value , b.WIDGET\_NAME AS Transformation\_name

FROM rep\_widget\_attr a , REP\_ALL\_TRANSFORMS b

WHERE a.attr\_id=4 AND a.widget\_id IN (SELECT widget\_id FROM REP\_ALL\_TRANSFORMS WHERE widget\_type\_name LIKE 'Sequence') AND a.widget\_id=b.widget\_id

**For a specific current value attribute of Sequence Generator, replace ' sequence\_generator\_transformation\_name ' in the following query with transformation name:**

SELECT a.attr\_value AS current\_value , b.WIDGET\_NAME AS Transformation\_name FROM rep\_widget\_attr a , REP\_ALL\_TRANSFORMS b WHERE a.attr\_id=4 AND a.widget\_id IN (SELECT widget\_id FROM REP\_ALL\_TRANSFORMS WHERE widget\_type\_name LIKE 'Sequence') AND a.widget\_id=b.widget\_id AND b.widget\_name LIKE ' sequence\_generator\_transformation\_name '

**For all or another attribute (other than current\_value ) of Sequence Generator transformation, run the following query:**

SELECT a.\* , b.WIDGET\_NAME AS Transformation\_name

FROM rep\_widget\_attr a , REP\_ALL\_TRANSFORMS b

WHERE a.attr\_id=4 AND a.widget\_id IN (SELECT widget\_id FROM REP\_ALL\_TRANSFORMS WHERE widget\_type\_name LIKE 'Sequence') AND a.widget\_id=b.widget\_id

**To include the version number:**

SELECT a.attr\_value AS current\_value , b.WIDGET\_NAME AS Transformation\_name, a.VERSION\_NUMBER

FROM rep\_widget\_attr a , REP\_ALL\_TRANSFORMS b

WHERE a.attr\_id=4 AND a.widget\_id IN (SELECT widget\_id FROM REP\_ALL\_TRANSFORMS WHERE widget\_type\_name LIKE 'Sequence') AND a.widget\_id=b.widget\_id

Order by a.VERSION\_NUMBER DESC

**PowerCenter query returns lookup connection information using repository views:**

SELECT f.SUBJECT\_AREA AS "Folder", m.MAPPING\_NAME AS "Mapping", s.INSTANCE\_NAME AS "Transformation", a.ATTR\_VALUE AS "connection"

FROM REP\_ALL\_MAPPINGS m, REP\_WIDGET\_INST s, REP\_WIDGET\_ATTR a, REP\_SUBJECT f

WHERE f.SUBJECT\_ID = m.SUBJECT\_ID AND m.MAPPING\_ID = s.MAPPING\_ID AND s.WIDGET\_TYPE = 11 AND s.WIDGET\_ID = a.WIDGET\_ID AND a.ATTR\_ID = 6

**The following query returns all the post session commands used inside each session of the repository:**

SELECT a.task\_name,c.subj\_name AS Folder\_name, b.PM\_value AS command\_value

FROM OPB\_TASK a,OPB\_TASK\_VAL\_LIST b, OPB\_SUBJECT c

WHERE a.task\_type=58 AND a.task\_name='post\_session\_command'

AND b.task\_id=a.task\_id AND b.subject\_id=c.subj\_id

**The following query returns all the post session success commands used inside each session of the repository:**

SELECT a.task\_name,c.subj\_name AS Folder\_name, b.PM\_value AS command\_value

FROM OPB\_TASK a,OPB\_TASK\_VAL\_LIST b, OPB\_SUBJECT c

WHERE a.task\_type=58 AND a.task\_name='post\_session\_success\_command'

AND b.task\_id=a.task\_id AND b.subject\_id=c.subj\_id

**The following query returns all the post session failure commands used inside each session of the repository:**

SELECT a.task\_name,c.subj\_name AS Folder\_name, b.PM\_value AS command\_value

FROM OPB\_TASK a,OPB\_TASK\_VAL\_LIST b, OPB\_SUBJECT c

WHERE a.task\_type=58 AND a.task\_name='post\_session\_failure\_command'

AND b.task\_id=a.task\_id AND b.subject\_id=c.subj\_id

**PowerCenter query can be used to find all version comments in a PowerCenter repository**

select s.subject\_area, s.subject\_id, r.comments, r.object\_name, r.version\_number, r.object\_type, r.user\_id, r.saved\_from

from rep\_version\_props r, rep\_subject s

where s.SUBJECT\_ID = r.subject\_id

AND r.comments is not null;

• OBJECT\_TYPE - Type of the versioned object.

• VERSION\_NUMBER - version number of the object.

• SUBJECT\_ID - subject Id of the object.

• USER\_ID - User Id who checked in this version of the object.

• OBJECT\_NAME - Name of the object.

• COMMENTS - checkin comments.

• SAVED\_FROM - saved from which client.

• Subject\_area - folder name

**PowerCenter Repository Query to fetch Source/Target Connection Information:**

SELECT

DISTINCT ALL\_SESSIONS.SUBJECT\_AREA FOLDER\_NAME,ALL\_SESSIONS.MAPPING\_NAME MAPPING\_NAME, ALL\_SESSIONS.SESSION\_NAME SESSION\_NAME,

SESSION\_ALL\_CNXS.READER\_WRITER\_TYPE, SESSION\_ALL\_CNXS.INSTANCE\_NAME, SESSION\_ALL\_CNXS.CNX\_NAME,

CASE

WHEN SESSION\_ALL\_CNXS.WIDGET\_TYPE = 2 THEN 'TARGET CONNECTION'

ELSE

CASE

WHEN SESSION\_ALL\_CNXS.WIDGET\_TYPE IN (1,3,56,45,55,84) THEN 'SOURCE CONNECTION'

ELSE

NULL

END

END,

SESSION\_USERS.USER\_NAME

FROM

REP\_VERSION\_PROPS SESSION\_VERSION\_PROPS,

REP\_USERS SESSION\_USERS, REP\_LOAD\_SESSIONS ALL\_SESSIONS,

REP\_REPOSIT\_INFO SESSION\_REPOSIT\_INFO, REP\_SESS\_WIDGET\_CNXS SESSION\_ALL\_CNXS

WHERE

(SESSION\_VERSION\_PROPS.USER\_ID = SESSION\_USERS.USER\_ID AND

ALL\_SESSIONS.SESSION\_ID <> SESSION\_REPOSIT\_INFO.REPOSITORY\_ID AND

ALL\_SESSIONS.SESSION\_ID = SESSION\_VERSION\_PROPS.OBJECT\_ID AND

ALL\_SESSIONS.SUBJECT\_ID = SESSION\_VERSION\_PROPS.SUBJECT\_ID AND

SESSION\_VERSION\_PROPS.OBJECT\_TYPE = 68 AND

ALL\_SESSIONS.SESSION\_ID = SESSION\_ALL\_CNXS.SESSION\_ID AND

ALL\_SESSIONS.SESSION\_VERSION\_NUMBER = SESSION\_ALL\_CNXS.SESSION\_VERSION\_NUMBER)

**PowerCenter query created using repository view that extracts all Lookup transformations in a folder and their associated lookup tables:**

SELECT

f.SUBJECT\_AREA AS "Folder Name", m.MAPPING\_NAME AS "Mapping Name", s.INSTANCE\_NAME AS "Transformation Name", a.ATTR\_VALUE AS "Query"

FROM

REP\_ALL\_MAPPINGS m, REP\_SUBJECT f, REP\_WIDGET\_INST s, REP\_WIDGET\_ATTR a

WHERE

f.SUBJECT\_ID = m.SUBJECT\_ID and m.MAPPING\_ID = s.MAPPING\_ID and s.WIDGET\_TYPE = 11 and s.WIDGET\_ID = a.WIDGET\_ID and a.ATTR\_ID = 2 and f.SUBJECT\_AREA = 'Folder\_name'

**PowerCenter repository query that returns the the number of sessions that ran on a specific day for each PowerCenter Integration Service:**

SELECT COUNT(\*), SERVER\_NAME

FROM REP\_TASK\_INST\_RUN WHERE TO\_CHAR(START\_TIME,'mm/dd/yyyy') = '12/10/2007' GROUP BY SERVER\_NAME

**PowerCenter query will return objects, whose status is not enabled:**

SELECT \*

FROM rep\_all\_tasks a, rep\_workflows b

WHERE a.subject\_id = b.subject\_id

AND a.task\_id = b.workflow\_id

AND b.workflow\_version\_number = (SELECT max(c.workflow\_version\_number)

FROM rep\_workflows c

WHERE c.workflow\_id = b.workflow\_id)

AND a.is\_enabled = 0;

This query returns the most recent version of the objects (for versioned repositories).

**PowerCenter repository query used in PowerCenter to find all the emails with attachment either used as part of email task or post-session commands:**

SELECT DISTINCT D.SUBJ\_NAME FOLDER\_NAME, C.WORKFLOW\_NAME, A.TASK\_NAME TASK\_NAME, B.ATTR\_VALUE FROM OPB\_TASK A, OPB\_TASK\_ATTR B, REP\_TASK\_INST\_RUN C, OPB\_SUBJECT D WHERE A.TASK\_ID = B.TASK\_ID AND A.TASK\_TYPE = B.TASK\_TYPE AND C.SUBJECT\_ID = A.SUBJECT\_ID AND A.SUBJECT\_ID = d.SUBJ\_ID AND A.TASK\_TYPE = 65 AND B.ATTR\_ID in (2,3) AND (B.ATTR\_VALUE like '%\%a%' ESCAPE '\' OR B.ATTR\_VALUE like '%\%g%' ESCAPE '\');

**PowerCenter Query to get the user\_name that last updated a workflow/mapping**

**The repository table REP\_VERSION\_PROPS, provides information about the history of user\_id, object\_name, last\_saved, saved\_from etc of the user who last modified this version of the object. You can query this table for the object and get the user\_id to find the user\_name from the REP\_USERS table.**

select a.user\_name,b.last\_saved,b.object\_name from rep\_users a,rep\_version\_props b where b.object\_name='objectname'and a.user\_id=b.user\_id

This way you can find the user who last modified this version of the object.

**MX repository views that returns the run details for a worklet in a specific workflow for a particular time period:**

SELECT DISTINCT WORKLET\_RUN.WORKFLOW\_NAME,ALL\_WORKLETS.TASK\_NAME, WORKLET\_RUN.START\_TIME, WORKLET\_RUN.END\_TIME

FROM REP\_ALL\_TASKS ALL\_WORKLETS, REP\_TASK\_INST\_RUN WORKLET\_RUN

WHERE (ALL\_WORKLETS.TASK\_ID = WORKLET\_RUN.TASK\_ID AND

ALL\_WORKLETS.VERSION\_NUMBER = WORKLET\_RUN.TASK\_VERSION\_NUMBER ) AND

(WORKLET\_RUN.WORKFLOW\_NAME = 'workflow\_name' AND

WORKLET\_RUN.START\_TIME > TO\_DATE('03/12/2008 18:00:00','MM/DD/YYYY HH24:MI:SS') AND ALL\_WORKLETS.TASK\_NAME like 'workletname%' )

ORDER BY

WORKLET\_RUN.START\_TIME

**PowerCenter query returns a list of all PowerCenter (8.1.x and earlier) repository users belonging to the Administrator group:**

SELECT USER\_NAME,REP\_GROUPS.GROUP\_NAME

FROM REP\_USERS,REP\_GROUPS, REP\_USER\_GROUPS

WHERE REP\_GROUPS.GROUP\_ID = 2

AND REP\_USERS.USER\_ID = REP\_USER\_GROUPS.USER\_ID

AND REP\_GROUPS.GROUP\_ID = REP\_USER\_GROUPS.GROUP\_ID

**PowerCenter query to find the reusable transformations in the repository :**

SELECT DISTINCT B.MAPPING\_NAME, A.\*

FROM REP\_ALL\_TRANSFORMS A, REP\_WIDGET\_INST C, REP\_ALL\_MAPPINGS B

WHERE PARENT\_WIDGET\_IS\_REUSABLE = 1

AND A.PARENT\_WIDGET\_ID = C.WIDGET\_ID

AND B.MAPPING\_ID = C.MAPPING\_ID

**PowerCenter query lists all workflows and the name of each workflow's associated Integration Service. If no Integration Service is assigned then the SERVER\_NAME column will not have any value.**

SELECT WORKFLOW\_NAME,SERVER\_NAME FROM REP\_WORKFLOWS

**PowerCenter Query a repository to find Workflows that use a scheduler**

**If you query the REP\_WORKFLOWS view, you can retrieve the workflows that use a scheduler. Following is the query:**

select workflow\_name, scheduler\_name from rep\_workflows where scheduler\_name is not null;

**Following is the Workflows query for reusable schedulers:**

select workflow\_name, scheduler\_name from rep\_workflows where scheduler\_is\_reusable = '1';

**Following is the Workflow query for a particular name:**

select workflow\_name, scheduler\_name from rep\_workflows where scheduler\_name = '

**PowerCenter Query to list the current memory settings (buffer block size, etc.) for all sessions in a PowerCenter repository using a repository query**

This information can be obtained from the REP\_TASK\_ATTR and REP\_SESS\_CONFIG\_PARM views, and the following query returns the values set for DTM buffer size, buffer block size, and line sequential buffer length for each session:

SELECT a.SUBJECT\_AREA AS Folder\_Name, a.task\_name AS Session\_Name, b.ATTR\_VALUE AS DTM\_BUFFER\_SIZE, c.ATTR\_VALUE AS Buffer\_Block\_Size,

d.ATTR\_VALUE AS Line\_Sequential\_Buffer\_Length

FROM REP\_ALL\_TASKS a, REP\_TASK\_ATTR b, REP\_SESS\_CONFIG\_PARM c, REP\_SESS\_CONFIG\_PARM d

WHERE a.TASK\_ID = b.TASK\_ID

AND a.TASK\_ID = c.SESSION\_ID

AND a.TASK\_ID = d.SESSION\_ID

AND b.ATTR\_ID = 101

AND c.ATTR\_ID = 5

AND d.ATTR\_ID = 6

ORDER BY 1, 2

**PowerCenter query can be used to determine session connections and memory properties in the PowerCenter repository**

The following query can be used to obtain the memory information:

SELECT ATTR\_NAME, ATTR\_VALUE,SESSION\_INSTANCE\_NAME

FROM REP\_SESS\_CONFIG\_PARM,REP\_SESSION\_INSTANCES

WHERE REP\_SESS\_CONFIG\_PARM.SESSION\_ID = REP\_SESSION\_INSTANCES.SESSION\_ID

**For connection details you can use the following query:**

SELECT CNX\_NAME

FROM REP\_SESS\_WIDGET\_CNXS,REP\_SESSION\_INSTANCES

WHERE REP\_SESSION\_INSTANCES .SESSION\_INSTANCE\_NAME = REP\_SESS\_WIDGET\_CNXS.INSTANCE\_NAME

**PowerCenter Query to get the list of source and target for a particular mapping, use the following query:**

select DISTINCT

MAPP.SUBJECT\_AREA FOLDER\_NAME,

MAPP.MAPPING\_NAME MAPPING\_NAME,

DECODE(WINST.WIDGET\_TYPE, '1', 'SOURCE', '2', 'TARGET') SOURCE\_TARGET,

WINST.INSTANCE\_NAME SOURCE\_TARGET\_NAME

from REP\_ALL\_MAPPINGS MAPP,

REP\_WIDGET\_INST WINST

where MAPP.MAPPING\_ID = WINST.MAPPING\_ID and

WINST.WIDGET\_TYPE in (1,2)

**Power Center query to get the workflows that have not been run from a particular date.**

select distinct subject\_area , workflow\_name from rep\_task\_inst\_run where

workflow\_name not in (select workflow\_name from rep\_task\_inst\_run where

start\_time > sysdate - 1 )

order by 1,2

Where, the start\_time filter condition can be altered to the required date to filter as per the needs.

**PowerCenter query to display the current value of all reusable and non-reusable Sequence Generators in mappings:**

SELECT

B.SUBJECT\_AREA AS FOLDER,

M.MAPPING\_NAME,

A.ATTR\_VALUE AS CURRENT\_VALUE,

B.WIDGET\_NAME AS TRANSFORMATION\_NAME

FROM REP\_WIDGET\_ATTR A , REP\_ALL\_TRANSFORMS B, REP\_WIDGET\_INST W, REP\_ALL\_MAPPINGS M

WHERE

A.ATTR\_ID=4

AND

A.WIDGET\_ID IN (SELECT WIDGET\_ID FROM REP\_ALL\_TRANSFORMS WHERE WIDGET\_TYPE\_NAME LIKE 'Sequence')

AND

A.WIDGET\_ID = B.WIDGET\_ID

AND

W.WIDGET\_ID = A.WIDGET\_ID

AND

W.WIDGET\_TYPE = 7

AND

W.MAPPING\_ID = M.MAPPING\_ID

**PowerCenter query to display the current value of all reusable Sequence Generators in mapplets:**

SELECT

B.SUBJECT\_AREA AS FOLDER,

M.MAPPLET\_NAME, A.ATTR\_VALUE AS CURRENT\_VALUE, B.WIDGET\_NAME AS TRANSFORMATION\_NAME

FROM REP\_WIDGET\_ATTR A , REP\_ALL\_TRANSFORMS B, REP\_WIDGET\_INST W, REP\_ALL\_MAPPLETS M

WHERE A.ATTR\_ID=4

AND

A.WIDGET\_ID IN (SELECT WIDGET\_ID FROM REP\_ALL\_TRANSFORMS WHERE WIDGET\_TYPE\_NAME LIKE 'Sequence')

AND

A.WIDGET\_ID = B.WIDGET\_ID

AND

W.WIDGET\_ID = A.WIDGET\_ID

AND

W.WIDGET\_TYPE = 7

AND

W.MAPPING\_ID = M.MAPPLET\_ID

**PowerCenter query to get Lookup SQL Override and SQL Query of Lookup Procedure and Source Qualifier transformations respectively at mapping level:**

SELECT DISTINCT D.SUBJECT\_AREA FOLDER, D.MAPPING\_NAME, A.WIDGET\_TYPE\_NAME TRANSFORMATION\_TYPE, A.INSTANCE\_NAME TRANSFORMATION\_NAME, B.ATTR\_NAME, B.ATTR\_VALUE, C.SESSION\_NAME FROM REP\_WIDGET\_INST A, REP\_WIDGET\_ATTR B, REP\_LOAD\_SESSIONS C, REP\_ALL\_MAPPINGS D WHERE B.WIDGET\_ID = A. WIDGET\_ID AND B.WIDGET\_TYPE = A. WIDGET\_TYPE AND B.WIDGET\_TYPE IN (3, 11) AND C.MAPPING\_ID = A.MAPPING\_ID AND D.MAPPING\_ID = A.MAPPING\_ID AND B.ATTR\_ID= 1 AND B.ATTR\_DATATYPE=2 AND B.ATTR\_TYPE=3 ORDER BY D.SUBJECT\_AREA, D.MAPPING\_NAME

**Power Center Query to get the following details:**

**• Mapping Name**

**• Sequence Generator used within a mapping**

**• Current Sequence generator value**

**• Session Name and**

**• Workflow Name**

SELECT DISTINCT MAPPING\_ALL\_TRANSFORMS.WIDGET\_NAME, MAPPING\_ALL\_TRANSFORMS.WIDGET\_TYPE\_NAME, ALL\_MAPPINGS.MAPPING\_NAME, MAPPING\_SESSION\_INSTANCES.SESSION\_INSTANCE\_NAME, MAPPING\_TRANSFORM\_WIDGET\_ATTR.ATTR\_NAME,

CASE WHEN MAPPING\_TRANSFORM\_WIDGET\_ATTR.ATTR\_DATATYPE IN (1,4) THEN CASE WHEN MAPPING\_TRANSFORM\_WIDGET\_ATTR.ATTR\_TYPE = 5 THEN CASE WHEN MAPPING\_TRANSFORM\_WIDGET\_ATTR.ATTR\_VALUE = 1 THEN 'Yes' ELSE 'No' END ELSE MAPPING\_TRANSFORM\_WIDGET\_ATTR.ATTR\_VALUE END ELSE MAPPING\_TRANSFORM\_WIDGET\_ATTR.ATTR\_VALUE END

FROM REP\_WIDGET\_INST MAPPING\_WIDGET\_INST, REP\_ALL\_TRANSFORMS MAPPING\_ALL\_TRANSFORMS, REP\_SESSION\_INSTANCES MAPPING\_SESSION\_INSTANCES, REP\_WIDGET\_ATTR MAPPING\_TRANSFORM\_WIDGET\_ATTR, REP\_ALL\_MAPPINGS ALL\_MAPPINGS

LEFT OUTER JOIN REP\_SESS\_PARTITION\_DEF MAPPING\_SESS\_PARTITION\_DEF ON ALL\_MAPPINGS.PARENT\_MAPPING\_ID = MAPPING\_SESS\_PARTITION\_DEF.MAPPING\_ID

WHERE (ALL\_MAPPINGS.PARENT\_MAPPING\_ID = MAPPING\_WIDGET\_INST.MAPPING\_ID AND ALL\_MAPPINGS.PARENT\_MAPPING\_VERSION\_NUMBER = MAPPING\_WIDGET\_INST.VERSION\_NUMBER AND ALL\_MAPPINGS.SUBJECT\_ID = MAPPING\_WIDGET\_INST.SUBJECT\_ID AND MAPPING\_ALL\_TRANSFORMS.PARENT\_WIDGET\_ID = MAPPING\_WIDGET\_INST.WIDGET\_ID AND MAPPING\_ALL\_TRANSFORMS.SUBJECT\_ID = MAPPING\_WIDGET\_INST.SUBJECT\_ID AND MAPPING\_ALL\_TRANSFORMS.WIDGET\_TYPE\_ID = MAPPING\_WIDGET\_INST.WIDGET\_TYPE AND MAPPING\_SESSION\_INSTANCES.SESSION\_ID = MAPPING\_SESS\_PARTITION\_DEF.SESSION\_ID AND MAPPING\_SESSION\_INSTANCES.SESSION\_VERSION\_NUMBER = MAPPING\_SESS\_PARTITION\_DEF.VERSION\_NUMBER AND MAPPING\_WIDGET\_INST.WIDGET\_ID = MAPPING\_TRANSFORM\_WIDGET\_ATTR.WIDGET\_ID AND MAPPING\_WIDGET\_INST.WIDGET\_TYPE = MAPPING\_TRANSFORM\_WIDGET\_ATTR.WIDGET\_TYPE AND MAPPING\_TRANSFORM\_WIDGET\_ATTR.SESSION\_TASK\_ID = 0 AND MAPPING\_TRANSFORM\_WIDGET\_ATTR.WIDGET\_TYPE NOT IN (1,2,44)) AND (MAPPING\_ALL\_TRANSFORMS.WIDGET\_TYPE\_NAME IN ('Sequence') AND MAPPING\_TRANSFORM\_WIDGET\_ATTR.ATTR\_NAME IN ('Current Value') )

**PowerCenter query on the repository views to get the mapping which has sorter transformation:**

SELECT c.subject\_area folder\_name,a.task\_name workflow\_name,b.instance\_name session\_name, c.mapping\_name,

d.instance\_name transformation\_name

FROM REP\_ALL\_TASKS a,REP\_TASK\_INST b,REP\_LOAD\_SESSIONS c,REP\_WIDGET\_INST d

WHERE a.subject\_id= c.subject\_id AND a.task\_type=71 AND a.task\_id=b.workflow\_id

AND b.task\_type=68 AND b.task\_id=c.session\_id

AND c.mapping\_id=d.mapping\_id

AND d.widget\_type=80

**PowerCenter query to list the users, their status (Active/In-Active) and the associated Group for 8.1.1 repository version.**

Select USER\_NAME, DECODE(USER\_ISENABLED,0,'IN-Active','Active') as "Status", GROUP\_NAME, USER\_DESCRIPTION from REP\_USERS u, REP\_GROUPS g,

REP\_USER\_GROUPS ug

Where

ug.USER\_ID = u.USER\_ID AND

g.GROUP\_ID = ug.GROUP\_ID

**Check if the non versioned repository contains versioned objects, which may be induced due to Repository Service bugs**

In some cases, a non versioned repository service converts into a versioned repository service without having a team-based development license option.

The following error appears when the repository service is restarted:

REP\_51849 : A versioned repository cannot be started or restored without a team-based development license key.

Run the following queries against the Repository Service Database schema of the non versioned repository:

SELECT \* FROM OPB\_MACRO WHERE version\_number>1

SELECT \* FROM OPB\_MAPPING WHERE version\_number>1

SELECT \* FROM OPB\_MD\_CUBE WHERE version\_number>1

SELECT \* FROM OPB\_MD\_DIMENSION WHERE version\_number>1

SELECT \* FROM OPB\_SCHEDULER WHERE version\_number>1

SELECT \* FROM OPB\_SESSION\_CONFIG WHERE version\_number>1

SELECT \* FROM OPB\_SHORTCUT WHERE version\_number>1

SELECT \* FROM OPB\_SRC WHERE version\_number>1

SELECT \* FROM OPB\_SUBJECT WHERE version\_number>1

SELECT \* FROM OPB\_TARG WHERE version\_number>1

SELECT \* FROM OPB\_TASK WHERE version\_number>1

SELECT \* FROM OPB\_WIDGET WHERE version\_number>1

If the above queries return a set of rows, it can indicate that there is a bug. Contact Informatica Global Customer Support for further assistance.

**PowerCenter query returns all sessions with the "$DB" connection name as the connection object:**

SELECT A.CNX\_NAME, B.SESSION\_NAME

FROM REP\_SESS\_WIDGET\_CNXS A, REP\_LOAD\_SESSIONS B

WHERE A.SESSION\_ID=B.SESSION\_ID AND A.CNX\_NAME LIKE '%$%'

**PowerCenter query to return the start and end time for 5 previous runs of a workflow:**

SELECT \*

FROM REP\_WFLOW\_RUN

WHERE WORKFLOW\_NAME LIKE '' AND rownum < 6

ORDER BY WORKFLOW\_RUN\_ID DESC

**PowerCenter query to find the source name in a mapping:**

SELECT \* from rep\_src\_mapping

where mapping\_name=

and subject\_area like ;

**PowerCenter query to generate a report of the SQL Overrides using the MX views:**

SELECT RAM.SUBJECT\_ID,RAM.MAPPING\_NAME, RWA.ATTR\_NAME, RWA.ATTR\_VALUE

FROM REP\_WIDGET\_ATTR RWA, REP\_ALL\_MAPPINGS RAM,

(SELECT SUBJECT\_ID FROM REP\_SUBJECT WHERE SUBJECT\_AREA = '') X

WHERE RWA.ATTR\_NAME LIKE '%Sql Override'

AND RAM.MAPPING\_ID = RWA.MAPPING\_ID

AND RAM.SUBJECT\_ID = X.SUBJECT\_ID

AND RWA.ATTR\_VALUE IS NOT NULL

Where is the name of a specific folder in the repository.

**PowerCenter Query to list all the parameters and variables used in mappings and workflows in the repository:**

SELECT DISTINCT F.SUBJ\_NAME AS FOLDER\_NAME,

M.MAPPING\_NAME AS Object\_Name, 'Mapping' AS Object\_Type ,

P.PV\_NAME AS PARAMETER\_OR\_VARIABLE\_NAME,

(CASE WHEN P.PV\_FLAG=2 THEN 'Parameter'WHEN P.PV\_FLAG=3 THEN 'Variable' END) AS "TYPE" FROM OPB\_SUBJECT F, OPB\_MAPPING M, OPB\_MAP\_PARMVAR P WHERE F.SUBJ\_ID = M.SUBJECT\_ID

AND M.MAPPING\_ID = P.MAPPING\_ID

AND F.SUBJ\_ID = P.SUBJECT\_ID

UNION

SELECT DISTINCT F.SUBJ\_NAME AS FOLDER\_NAME,

WFR.WORKFLOW\_NAME AS Object\_Name,

'WorkFlow' AS Object\_Type ,

WP.VAR\_NAME,

(CASE WHEN WP.VAR\_TYPE=1 THEN 'User\_Defined' END) AS "TYPE"

FROM OPB\_SUBJECT F, OPB\_WORKFLOW WF, OPB\_WFLOW\_VAR WP , OPB\_WFLOW\_RUN WFR

WHERE WF.WORKFLOW\_ID = WP.WORKFLOW\_ID

AND F.SUBJ\_ID = WP.SUBJECT\_ID

AND WFR.WORKFLOW\_ID = WF.WORKFLOW\_ID

AND WP.VAR\_TYPE <> '0'

ORDER BY Object\_Type

**If it is not required to view the objects already deleted from the repository, then use the following query:**

SELECT DISTINCT F.SUBJ\_NAME AS FOLDER\_NAME,

M.MAPPING\_NAME AS Object\_Name, 'Mapping' AS Object\_Type ,

P.PV\_NAME AS PARAMETER\_OR\_VARIABLE\_NAME,

(CASE WHEN P.PV\_FLAG=2 THEN 'Parameter'WHEN P.PV\_FLAG=3 THEN 'Variable' END) AS "TYPE"

FROM OPB\_SUBJECT F, OPB\_MAPPING M, OPB\_MAP\_PARMVAR P WHERE F.SUBJ\_ID = M.SUBJECT\_ID

AND M.MAPPING\_ID = P.MAPPING\_ID

AND F.SUBJ\_ID = P.SUBJECT\_ID

AND M.MAPPING\_NAME NOT IN (SELECT MAPPING\_NAME FROM OPB\_MAPPING WHERE VERSION\_STATUS = 10)

UNION

SELECT DISTINCT F.SUBJ\_NAME AS FOLDER\_NAME,

WFR.WORKFLOW\_NAME AS Object\_Name,

'WorkFlow' AS Object\_Type ,

WP.VAR\_NAME,

(CASE WHEN WP.VAR\_TYPE=1 THEN 'User\_Defined' END) AS "TYPE"

FROM OPB\_SUBJECT F, OPB\_WORKFLOW WF, OPB\_WFLOW\_VAR WP , OPB\_WFLOW\_RUN WFR

WHERE WF.WORKFLOW\_ID = WP.WORKFLOW\_ID

AND F.SUBJ\_ID = WP.SUBJECT\_ID

AND WFR.WORKFLOW\_ID = WF.WORKFLOW\_ID

AND WP.VAR\_TYPE <> '0'

AND WFR.WORKFLOW\_NAME NOT IN (SELECT TASK\_NAME FROM OPB\_TASK WHERE VERSION\_STATUS = 10)

ORDER BY Object\_Type

**To list all unscheduled workflows in a PowerCenter repository do the following:**

1. Run the following query which returns all the scheduled workflows in the repository:

select workflow\_name, scheduler\_name from rep\_workflows where scheduler\_name is not null;

2. Run the pmcmd command to get the currently scheduled workflows for each integration service connecting to the repository.

pmcmd getservicedetails -sv -u -p -scheduled -d

3. The workflow names in step 1 which are not in step 2 are the unscheduled workflows, provided all workflows are scheduled initially.

**PowerCenter query to determine if a particular table is being used as a source or target or lookup:**

SELECT PARENT\_SUBJECT\_AREA FOLDER, PARENT\_MAPPING\_NAME MAPPING

FROM REP\_ALL\_MAPPINGS

WHERE MAPPING\_ID IN (

SELECT MAPPING\_ID FROM REP\_WIDGET\_INST

WHERE WIDGET\_ID IN (

SELECT WIDGET\_ID FROM REP\_WIDGET\_ATTR

WHERE (WIDGET\_TYPE = 11 OR WIDGET\_TYPE = 1 OR WIDGET\_TYPE = 2) AND (ATTR\_ID = 2 OR ATTR\_ID = 31 OR ATTR\_ID = 19) AND ATTR\_VALUE = '' AND WIDGET\_ID IN (

SELECT WIDGET\_ID FROM REP\_ALL\_TRANSFORMS

WHERE

PARENT\_SUBJECT\_AREA = '')))

**PowerCenter query to get a list of sessions and folders in a repository:**

SELECT DISTINCT (TASK\_NAME), TASK\_TYPE\_NAME, SUBJECT\_AREA

FROM REP\_ALL\_TASKS WHERE TASK\_TYPE\_NAME = 'Session'

ORDER BY SUBJECT\_AREA

**PowerCenter query will returns mapping information with associated folders, workflows, and sessions:**

SELECT c.subject\_area folder\_name,a.task\_name workflow\_name,b.instance\_name session\_name, c.mapping\_name

FROM REP\_ALL\_TASKS a,REP\_TASK\_INST b,REP\_LOAD\_SESSIONS c

WHERE a.subject\_id=c.subject\_id AND a.task\_type=71

AND a.task\_id=b.workflow\_id AND b.task\_type=68 AND b.task\_id=c.session\_id

AND c.MAPPING\_ID <> 0

ORDER BY 1,2,3,4

**PowerCenter query given below to list all the SAP sources in a repository.**

SELECT \* FROM REP\_ALL\_SOURCES where PARENT\_SOURCE\_DATABASE\_TYPE like 'SAP%'

**PowerCenter query to generate a list of failed sessions by running a query against the MX views.**

SELECT

rsl.subject\_area as folder,

rw.workflow\_name as workflow,

rsl.session\_name as session\_name,

decode(rsl.run\_status\_code,3,'Failed',4,'Stopped',5,'Aborted',15,'Terminated','Unknown') as status,

rsl.first\_error\_code as first\_error,

rsl.first\_error\_msg as error\_msg,

rsl.actual\_start as start\_time,

rsl.session\_timestamp as end\_time

FROM rep\_sess\_log rsl,

rep\_workflows rw

WHERE rsl.run\_status\_code in (3,4,5,14,15)

and rw.workflow\_id = rsl.workflow\_id

and rw.subject\_id = rsl.subject\_id

ORDER BY rsl.session\_timestamp desc

This query will return the list of sessions that did not complete successfully, ordered by time. This is a basic query that can be modified by the user(s) to suit their needs.

**Power Center query returns the Joiner cache information with associated folders, workflows, sessions, mappings, and transformations:**

SELECT c.subject\_area folder\_name,a.task\_name workflow\_name,b.instance\_name session\_name, c.mapping\_name,

d.instance\_name transformation\_name, DECODE(e.attr\_id,10,'data\_cache',11,'index\_cache',2,'cache\_directory') cache\_type,

e.attr\_value

FROM REP\_ALL\_TASKS a,REP\_TASK\_INST b,REP\_LOAD\_SESSIONS c,REP\_WIDGET\_INST d, REP\_WIDGET\_ATTR e

WHERE c.subject\_id=a.subject\_id AND a.task\_type=71 AND a.task\_id=b.workflow\_id AND b.task\_type=68 AND b.task\_id=c.session\_id

AND c.mapping\_id=d.mapping\_id AND d.widget\_id=e.widget\_id

AND d.widget\_type=12 AND e.widget\_type=12 AND e.attr\_id IN (10,11,2)

ORDER BY 1,2,3,4,5,6

**PowerCenter query returns the Rank cache information with associated folders, workflows, sessions, mappings, and transformations:**

SELECT c.subject\_area folder\_name,a.task\_name workflow\_name,b.instance\_name session\_name, c.mapping\_name,

d.instance\_name transformation\_name, DECODE(e.attr\_id,6,'data\_cache',7,'index\_cache', 1,'cache\_directory') cache\_type, e.attr\_value

FROM REP\_ALL\_TASKS a,REP\_TASK\_INST b,REP\_LOAD\_SESSIONS c,REP\_WIDGET\_INST d, REP\_WIDGET\_ATTR e

WHERE a.subject\_id= c.subject\_id AND a.task\_type=71 AND a.task\_id=b.workflow\_id

AND b.task\_type=68 AND b.task\_id=c.session\_id

AND c.mapping\_id=d.mapping\_id AND d.widget\_id=e.widget\_id

AND d.widget\_type=26 AND e.widget\_type=26 AND e.attr\_id IN (1,6,7)

ORDER BY 1,2,3,4,5,6

**PowerCenter query returns the Sorter cache information with associated folders, workflows, sessions, mappings, and transformations:**

SELECT c.subject\_area folder\_name,a.task\_name workflow\_name,b.instance\_name session\_name, c.mapping\_name,

d.instance\_name transformation\_name, DECODE(e.attr\_id,1,'cache\_size', 3,'cache\_directory') cache\_type, e.attr\_value

FROM REP\_ALL\_TASKS a,REP\_TASK\_INST b,REP\_LOAD\_SESSIONS c,REP\_WIDGET\_INST d, REP\_WIDGET\_ATTR e

WHERE a.subject\_id= c.subject\_id AND a.task\_type=71 AND a.task\_id=b.workflow\_id

AND b.task\_type=68 AND b.task\_id=c.session\_id

AND c.mapping\_id=d.mapping\_id AND d.widget\_id=e.widget\_id

AND d.widget\_type=80 AND e.widget\_type=80 AND e.attr\_id IN (1,3)

ORDER BY 1,2,3,4,5,6

**PowerCenter query returns the Aggregator cache information with associated folders, workflows, sessions, mappings, and transformations:**

SELECT c.subject\_area folder\_name,a.task\_name workflow\_name,b.instance\_name session\_name,c.mapping\_name,

d.instance\_name transformation\_name,DECODE(e.attr\_id,4,'data\_cache',5,'index\_cache', 1,'cache\_directory') cache\_type, e.attr\_value

FROM REP\_ALL\_TASKS a,REP\_TASK\_INST b, REP\_LOAD\_SESSIONS c,REP\_WIDGET\_INST d, REP\_WIDGET\_ATTR e

WHERE c.subject\_id=a.subject\_id AND a.task\_type=71 AND a.task\_id=b.workflow\_id AND b.task\_type=68

AND b.task\_id=c.session\_id AND c.mapping\_id=d.mapping\_id AND d.widget\_id=e.widget\_id

AND d.widget\_type=9 AND e.widget\_type=9 AND e.attr\_id IN (4,5,1)

ORDER BY 1,2,3,4,5,6

**PowerCenter query will return workflow name, task name, and the location for the user-defined event.**

SELECT DISTINCT b.task\_name as WorkflowName,c.task\_name as TaskName, a.attr\_value

FROM rep\_task\_attr a, rep\_all\_tasks b, rep\_all\_tasks c

WHERE a.task\_id = c.task\_id AND

b.task\_id = c.task\_id AND

a.task\_type = 61

AND a.task\_id in (SELECT task\_id FROM rep\_task\_attr

WHERE attr\_name='User Defined Event')

**PowerCenter query helps to retrieve persisted value for each session:**

select distinct e.mapping\_name,a.task\_name, c.pv\_value from

opb\_task a,opb\_task\_inst b,OPB\_MAP\_PERSISVAL c,opb\_Session d,opb\_mapping e

where e.mapping\_id=c.mapping\_id and

c.subject\_id=e.subject\_id and

b.instance\_id=c.session\_inst\_id and

b.task\_id=a.task\_id and a.task\_type=68 and b.task\_type=68 and

a.subject\_id=c.subject\_id and

d.session\_id=a.task\_id and

d.session\_id=b.task\_id and

d.mapping\_id=e.mapping\_id

**PowerCenter query will return all the tasks and sessions within a workflow in a certain folder and also link conditions between each task:**

SELECT DISTINCT b.INSTANCE\_NAME AS FROM\_INST, a.CONDITION as condition\_in\_bw, c.INSTANCE\_NAME AS TO\_INST

FROM REP\_WORKFLOW\_DEP a, REP\_TASK\_INST b, REP\_TASK\_INST c

WHERE a.FROM\_INSTANCE\_ID = b.INSTANCE\_ID

AND a.TO\_INSTANCE\_ID = c.INSTANCE\_ID

AND a.WORKFLOW\_ID = b.WORKFLOW\_ID

AND a.WORKFLOW\_ID = c.WORKFLOW\_ID

AND b.WORKFLOW\_ID in (SELECT WORKFLOW\_ID FROM REP\_WORKFLOWS WHERE WORKFLOW\_NAME = '{WorkflowName}' AND SUBJECT\_AREA ='{FolderName}')

**PowerCenter query can be run on the PowerCenter repository to find the sessions which are currently running:**

SELECT

SESSION\_TASK\_INST\_RUN.SERVER\_NAME, SESSION\_TASK\_REPOSIT\_INFO.REPOSITORY\_NAME, TO\_DATE(TO\_CHAR(SESSION\_TASK\_INST\_RUN.END\_TIME, 'yyyy-MM-dd'), 'yyyy-MM-dd'), CASE WHEN SESSION\_LOG.RUN\_STATUS\_CODE = 1 THEN 'Succeeded' ELSE CASE WHEN SESSION\_LOG.RUN\_STATUS\_CODE = 2 THEN 'Disabled' ELSE CASE WHEN SESSION\_LOG.RUN\_STATUS\_CODE = 3 THEN 'Failed' ELSE CASE WHEN SESSION\_LOG.RUN\_STATUS\_CODE = 4 THEN 'Stopped' ELSE CASE WHEN SESSION\_LOG.RUN\_STATUS\_CODE = 5 THEN 'Aborted' ELSE CASE WHEN SESSION\_LOG.RUN\_STATUS\_CODE = 6 THEN 'Running' ELSE CASE WHEN SESSION\_LOG.RUN\_STATUS\_CODE = 7 THEN 'Suspending' ELSE CASE WHEN SESSION\_LOG.RUN\_STATUS\_CODE = 8 THEN 'Suspended' ELSE CASE WHEN SESSION\_LOG.RUN\_STATUS\_CODE = 9 THEN 'Stopping' ELSE CASE WHEN SESSION\_LOG.RUN\_STATUS\_CODE = 10 THEN 'Aborting' ELSE CASE WHEN SESSION\_LOG.RUN\_STATUS\_CODE = 11 THEN 'Waiting' ELSE CASE WHEN SESSION\_LOG.RUN\_STATUS\_CODE = 12 THEN 'Scheduled' ELSE CASE WHEN SESSION\_LOG.RUN\_STATUS\_CODE = 13 THEN 'UnScheduled' ELSE CASE WHEN SESSION\_LOG.RUN\_STATUS\_CODE = 14 THEN 'Unknown' ELSE CASE WHEN SESSION\_LOG.RUN\_STATUS\_CODE = 15 THEN 'Terminated' ELSE NULL END END END END END END END END END END END END END END END, (COUNT(SESSION\_TASK\_INST\_RUN.TASK\_ID))

FROM

REP\_TASK\_INST\_RUN SESSION\_TASK\_INST\_RUN, REP\_REPOSIT\_INFO SESSION\_TASK\_REPOSIT\_INFO, REP\_SESS\_LOG SESSION\_LOG

WHERE

(SESSION\_TASK\_INST\_RUN.TASK\_ID <> SESSION\_TASK\_REPOSIT\_INFO.REPOSITORY\_ID AND SESSION\_TASK\_INST\_RUN.TASK\_TYPE = 68 AND SESSION\_LOG.SESSION\_ID = SESSION\_TASK\_INST\_RUN.TASK\_ID AND SESSION\_LOG.INSTANCE\_ID = SESSION\_TASK\_INST\_RUN.INSTANCE\_ID AND SESSION\_LOG.TASK\_VERSION\_NUMBER = SESSION\_TASK\_INST\_RUN.TASK\_VERSION\_NUMBER AND SESSION\_LOG.WORKFLOW\_ID = SESSION\_TASK\_INST\_RUN.WORKFLOW\_ID AND SESSION\_LOG.WORKFLOW\_VERSION\_NUMBER = SESSION\_TASK\_INST\_RUN.VERSION\_NUMBER AND SESSION\_LOG.WORKFLOW\_RUN\_ID = SESSION\_TASK\_INST\_RUN.WORKFLOW\_RUN\_ID AND SESSION\_TASK\_INST\_RUN.TASK\_TYPE = 68 ) AND (SESSION\_TASK\_INST\_RUN.END\_TIME IS NULL )

GROUP BY

SESSION\_TASK\_INST\_RUN.SERVER\_NAME, SESSION\_TASK\_REPOSIT\_INFO.REPOSITORY\_NAME, TO\_DATE(TO\_CHAR(SESSION\_TASK\_INST\_RUN.END\_TIME, 'yyyy-MM-dd'), 'yyyy-MM-dd'), CASE WHEN SESSION\_LOG.RUN\_STATUS\_CODE = 1 THEN 'Succeeded' ELSE CASE WHEN SESSION\_LOG.RUN\_STATUS\_CODE = 2 THEN 'Disabled' ELSE CASE WHEN SESSION\_LOG.RUN\_STATUS\_CODE = 3 THEN 'Failed' ELSE CASE WHEN SESSION\_LOG.RUN\_STATUS\_CODE = 4 THEN 'Stopped' ELSE CASE WHEN SESSION\_LOG.RUN\_STATUS\_CODE = 5 THEN 'Aborted' ELSE CASE WHEN SESSION\_LOG.RUN\_STATUS\_CODE = 6 THEN 'Running' ELSE CASE WHEN SESSION\_LOG.RUN\_STATUS\_CODE = 7 THEN 'Suspending' ELSE CASE WHEN SESSION\_LOG.RUN\_STATUS\_CODE = 8 THEN 'Suspended' ELSE CASE WHEN SESSION\_LOG.RUN\_STATUS\_CODE = 9 THEN 'Stopping' ELSE CASE WHEN SESSION\_LOG.RUN\_STATUS\_CODE = 10 THEN 'Aborting' ELSE CASE WHEN SESSION\_LOG.RUN\_STATUS\_CODE = 11 THEN 'Waiting' ELSE CASE WHEN SESSION\_LOG.RUN\_STATUS\_CODE = 12 THEN 'Scheduled' ELSE CASE WHEN SESSION\_LOG.RUN\_STATUS\_CODE = 13 THEN 'UnScheduled' ELSE CASE WHEN SESSION\_LOG.RUN\_STATUS\_CODE = 14 THEN 'Unknown' ELSE CASE WHEN SESSION\_LOG.RUN\_STATUS\_CODE = 15 THEN 'Terminated' ELSE NULL END END END END END END END END END END END END END END END

ORDER BY

1, 2, 3, 4

**To determine the instance ID of a session task, execute the following query:**

select distinct instance\_id from REP\_TASK\_INST\_RUN where task\_name='' and rownum<2