Oracle® Retail Promotion Intelligence and Promotion Planning and Optimization

Operations On Demand Guide

Release 12.0.7

October 2007

Welcome to Oracle Retail Promotion Intelligence and Promotion Planning and Optimization 12.0.7. This Operations On Demand Guide offers reference material for systems administrators and others responsible for maintaining the Promotion Intelligence and Promotion Planning and Optimization production hosting environment. This document contains the following sections:

- Locating Host Specific Parameters
- Monitoring the Promote Server
- Monitoring the PCE
- Monitoring Batch Jobs
- Other Maintenance Activities

The release notes contain information on new features and issues that have been fixed as well as known issues that exist in this release.

Locating Host Specific Parameters

The following files contain host specific settings. Many of these default to “localhost” but may be different in a production environment.

```
$installdir/config/SIT/sit-config.properties
$installdir/config/promote/promote.properties
$installdir/modules/tools/bin/storesetupdater.sh
$installdir/modules/tools/conf/jndi.properties
$installdir/modules/tools/conf/promote-cmdline.properties
$installdir/modules/pce/sample/deploy.sh
$installdir/modules/pce/sample/images/install_is.sh
$installdir/config/SIT/server-list.xml
```

Monitoring the Promote Server

Besides the existing Oracle Application Server for monitoring facilities for monitoring applications, the following URL can be used to provide a pulse check of the web component to ensure that it is running properly:

```
http://${WL_HOST}:${WL_PORT}/promote/service.do?command=status
```

If everything is running properly, this command will return the string “OK”

Monitoring Policy Central Enterprise (PCE) Servers

To monitor the PCE servers, execute the following Linux command on each PCE host:

```
bash$ pceserver.sh -query status
Querying PCE server status....
PCE server is up running.
```

If the PCE server is not responding, use the following command to retrieve the server load.

```
bash$ pceserver.sh -query load
Querying PCE server load....
PCE server is running 0 applications.
```

For a listing of additional PCE server tools, run the pceserver.sh command with no options. A list of available commands will be provided.

Monitoring Batch Jobs

The following section provides an introduction batch jobs.

Overview

Batch operations are composed of tasks, steps, and agents.

- Tasks—Tasks are aligned with the major batch operations. These must be scheduled using a management tool and typically are enabled on one host.
- Steps—each task is composed of a number of steps, such as waiting for a file, unzipping it, processing it, etc.
- Agents—Tasks and steps are managed as a process outside the application server. Agents run within each application server.

Tasks

The following are the tasks currently configured in the system:

- weekly.load—this task is responsible for loading and processing the appropriate data passed to the application each week.
- weekly.restate—this task also runs weekly, but it is meant specifically for certain data files separate from weekly.load.
- nightly.load—running nightly, this task loads the data that needs to be processed on a nightly basis.
- nightly.export—this task also runs nightly and is responsible for preparing data for export back to the client.

Steps

Each task defines a list of steps required to execute the task. This list of steps is task specific and can be found within $AUTOMATION_
BASE/config/<task>/Proccess.steps.

The Job Log Table

Weekly and nightly feeds log their activities to the PR_MGMT_TASK table. The table below lists the columns in this table.

Table 1 PR_MGMT_TASK Table Attributes

Column Name	Data Type	Null-able	Data Default	Column ID	Primary Key	Comments
RUN_ID	Number (32,0)	No			1	
TASK_NAME	Varchar2 (40 byte)	No			2	
TASK_ID	Number (32,0)	Yes			3	
STEP_NAME	Varchar2 (80 byte)	No			4	
STEP_START	Date	No			5	
STEP_END	Date	Yes			6	
STEP_DURATION	Number (5.0)	Yes			7	
STEP_RESULT	Number (3,0)	No			8	
HOST	Varchar2 (200 byte)	No			9	
ARGUMENTS	Varchar2 (2048 byte)	Yes			10	
DESCRIPTION	Varchar2 (2048 byte)	Yes			11	

The application writes to this table with each step of its nightly or weekly batch cycle. The following is a brief example of what is written:

Table 2 Sample Nightly Batch Data within the PR_MGMT_TASK Table

RUN_ID	TASK_NAME	TASK_ID	STEP_NAME	STEP_RES	HOST	DESC
1	nightly.load		start	0	dev-app-101	Michaels Arts and Crafts: START in nightly.load (Run ID 20071018-1312) Automation log file is at /home/jputz/builds/mdc/operations/logs/automation/nightly.load.200710181312.log All the logs are at /home/jputz/builds/mdc/operations/logs

Table 2 (Cont.) Sample Nightly Batch Data within the PR_MGMT_TASK Table

RUN_ ID	TASK_ NAME	TASK_ ID	STEP_ NAME	STEP_ RES	HOST	DESC
2	nightly.load	1	generic/ wait.for.files.sh	0	dev-app-101	Running step 1/21 (generic/wait.for. files.sh) Step ended
3	nightly.load	1	update	0	dev-app-101	Beginning nightly processing...
...						
39	nighly.load	1	promote/ load/data.sh	0	dev-app-101	Running step 20/21 (promote/load.data. sh) Step ended
40	nightly.load	1	update	0	dev-app-101	Nightly load complete
41	nightly.load	1	end	0	dev-app-101	Michaels Arts and Crafts: SUCCESS in nightly.load (Run ID 200710181312)

As this example shows, each task is made up of many steps linked together using the TASK_ID column. Each step receives one row in the above table.

Understanding Job Failures

The critical column to watch is STEP_RESULT. If the step succeeds, there will be a zero in this column. A non-zero value represents a failure.

Operators should be alerted when a step fails. The following table illustrates an example failure.

Table 3   Sample Task Failures within the PR_MGMT_TASK Table

RUN_ID	TASK_NAME	TASK_ID	STEP_NAME	STEP_RES	HOST	DESC
1	weekly.load		start	0	dev-app-101	Michaels Arts and Crafts: START in nightly.load (Run ID 20071024-1517) Automation log file is at /home/jputz/builds/mdc/operations/logs/automation/weekly.load.200710241517.log All the logs are at /home/jputz/builds/mdc/operations/logs
2	weekly.load	1	generic/wait.for.files.sh	0	dev-app-101	Running step 1/21 (generic/wait.for.files.sh) Step ended
3	weekly.load	1	fail	2	dev-app-101	Michaels Arts and Crafts: FAILURE in weekly.load (RUN ID 200710241517)

Examining the log file noted in the description above, we can see that the client failed to deliver the data files in the specified time window:

```
2007.10.24 15:17:17 Started processing weekly.load on 200710241517 with log
at
/home/jputz/builds/mdc/operations/logs/automation/weekly.load.200710241517.lo
g.
2007.10.24 15:17:18 Running step generic/wait.for.files.sh (1/38):
2007.10.24 15:17:18 Step started at 2007.10.24 15:17:18.
2007.10.24 15:18:33 Have NOT received 'weekly.ash_lh_tbl.txt.gz'.
2007.10.24 15:18:33 Have NOT received 'weekly.ash_mh_tbl.txt.gz'.
2007.10.24 15:18:33 Have NOT received 'weekly.stage_mh_attrs_tbl.txt.gz'.
2007.10.24 15:18:33 Have NOT received 'weekly.bee_pr_like_
merchandise.txt.gz'.
2007.10.24 15:18:33 Have NOT received 'weekly.bee_promo_offer_attr.txt.gz'.
2007.10.24 15:18:33 Have NOT received 'weekly.bee_promo_offer_merch.txt.gz'.
2007.10.24 15:18:33 Have NOT received 'weekly.bee_promo_offer.txt.gz'.
2007.10.24 15:18:33 Have NOT received 'weekly.bee_promo_store.txt.gz'.
2007.10.24 15:18:33 Have NOT received 'weekly.bee_promotions.txt.gz'.
2007.10.24 15:18:33 Have NOT received 'weekly.wk_hist_sales_inv.txt.gz'.
2007.10.24 15:18:33 Have NOT received 'weekly.bee_mb_detail.txt.gz'.
2007.10.24 15:18:33 Have NOT received 'weekly.data_ready.txt'.
2007.10.24 15:18:33 FATAL: Files are late and abort time has passed. Giving
```

```
up.
2007.10.24 15:18:33 Step ended at 2007.10.24 15:18:33. Elapsed time is 75
seconds.
2007.10.24 15:18:34 Running procedure 'pr_mgmt.task_fail' with parameters
'223, 2 , 'Michaels Arts and Crafts: FAILURE in weekly.load (Run ID
200710241517)''
2007.10.24 15:18:34 Ran procedure 'pr_mgmt.task_fail' successfully.
```

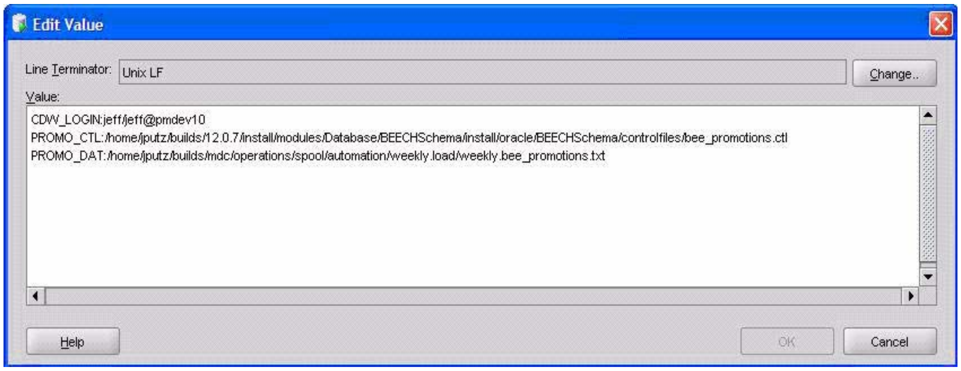
Locating Bad Data

A common failure is bad data received from the client. This may come across several ways. Suppose the client mistakenly puts a header row on the input feed. The following table is a sample task log illustrating the failed step.

Table 4 Sample Bad Data Example

RUN_ ID	TASK_ NAME	TASK_ ID	STEP_ NAME	STEP_ RES	HOST	DESC
239	weekly.load	226	start	2	dev-app-101	Running step 13/38 (db/oracle/stage.file .sh Step ended

Using SQL Developer, the failed file can be located as follows:



CDW_LOGIN:jeff/jeff@pmdev10
PROMO_CTL:/home/jputz/builds/12.0.7/install/modules/Database/BEECHSchema/install/oracle/BEECHSchema/controlfiles/bee_promotions.ctl
PROMO_DAT:/home/jputz/builds/mdc/operations/spool/automation/weekly.load/weekly.bee_promotions.txt

The failed step was a result of a bad weekly.bee_promotions.txt file. Open the log file, and scan it for the failed task as illustrated in the example below:

```
2007.10.24 16:14:35 Running step db/oracle/stage.file.sh (13/38):
2007.10.24 16:14:35 Step started at 2007.10.24 16:14:35.
2007.10.24 16:14:35 Staging logs at
'/home/jputz/builds/mdc/operations/logs/database/stage/weekly.bee_
promotions.txt.200710241614.log'.
2007.10.24 16:14:35 Bad records file at
'/home/jputz/builds/mdc/operations/logs/database/stage/weekly.bee_
promotions.txt.200710241614.bad'.
2007.10.24 16:14:36 Staging file
'/home/jputz/builds/mdc/operations/spool/automation/weekly.load/weekly.bee_
promotions.txt' with control file
'/home/jputz/builds/12.0.7/install/modules/Database/BEECHSchema/install/oracl
e/BEECHSchema/controlfiles/bee_promotions.ctl'...

Load completed - logical record count 23.
2007.10.24 16:14:36 FATAL: sqlldr exited with an error.
2007.10.24 16:14:36 Step ended at 2007.10.24 16:14:36. Elapsed time is 1
seconds
```

By examining the log files as noted above, we can see that the failure was a number format error and the actual offending record.

Restarting Jobs

Failed weekly batch jobs can be restarted using the $AUTOMATION_BASE/scripts/do.weely.load script.

Jobs can also be restarted at a certain step by providing the line number of the job to the task script. For example, running $AUTOMATION_BASE/scripts/do.weekly.load 7 would run weekly batch starting at step 7 assuming that the files have already been placed in their processing/spool directory.

Other Maintenance Activities

The following are additional steps that can be taken to ensure proper maintenance of the production environment.

Starting the Stopping the Promote Server

Use the standard Oracle Application Server facilities to accomplish this.

Starting and Stopping the PCE

Use the following pceserver.sh commands to control and monitor the PCE server:

```
$bash pceserver.sh -start
$bass pceserver.sh -shutdown
```