amdocs rating

# Rating 6.0
Run Book

amdocs



## Document Information

| | |
|---|---|
| Software Version: | **6.0** |
| Publication Date: | **January 2005; updated July 2006 for SP7** |
| Catalog Number: | **211925** |

# Contents

# 1. INTRODUCTION

Amdocs Rating processes and assigns prices to "events." An event is a usage of the services offered by a communications service provider. Rating applies price plans, allowances, discounts, and additional charges to events, resulting in a rated event that is sent to downstream systems such as Amdocs Billing.

Rating also maintains counters, called performance indicators that accumulate statistics about certain event attributes (for example, the total duration of voice calls during a billing cycle).

## Scope of this Document

This document describes how to run the Amdocs Rating processes.

## Batch Job Information

Each batch job is described in a separate chapter in this document. The following information is provided for each batch job:

| Subject | Description |
|---|---|
| Name | Both operational and full name of the job (It is also the name of the chapter). |
| Description | A description of the job functionality, including the purpose of the job, what it does, and what the results of a successful run are. This includes both the *job type* (the job's relations with the database, for example, update, retrieve, independent, etc.) and the *run frequency* (the frequency of the job runs, for example, daily, monthly, by request, etc.) |
| Activation and Shutdown | This information includes the command line, the script and executable used for activation and shutdown as well as whether the activation or shut down are through the AMC or Screen Composer. |
| Log Files | This includes the *name*, *location*, and *contents* of the log files. |
| Job Flow | This includes the *run level (*the running hierarchy as determined when running with other jobs as a map), the *preceding jobs (*on which the job depends), the *input files (*including their location and format), the *flow* (the flow of the job; and the activities it performs), the *output files (*including their location and format), and the *dependent jobs (*dependent on the successful completion of the job).. |
| Parameters | The parameters (variables) used by the job. |
| Databases and images | The databases the job uses, for example, *Usage1*, *Usage2*, *Customer*, etc. Additionally this section provides information of the shadow database or the partition where the batch job is running. |

| Subject | Description |
|---|---|
| Required Resources and Services | Unique resources required for the job, such as disk space or tape devices and unique services, such as background daemons, system devices, etc. |
| Distribution | The distribution for the job's results. |
| Console Messages | Messages sent to the user about the status of the job. |
| Troubleshooting | Common error messages with brief solutions. |
| Recovery Instructions | Recovery instructions for job failure. |
| Severity Level | The severity level of the job (When running as part of a map) |
| Special Remarks | Special information, such as unique passwords, that are required to run the job. |

# Language

The programming language in which the batch files are written is C++

# Input and Output Directories

Rating input and output files are placed in these permanent directories:

- Interface input placed in: $ABP_PM_ROOT/interfaces/input
- Interface output placed in: $ABP_PM_ROOT/interfaces/output

# General Troubleshooting

This section offers some general troubleshooting suggestions for problems that are not connected to any specific batch job.

## I/O Errors

If a process is aborted because it failed to open, read, write, or close any file, analyze the returned status that appears in the error message, then correct the cause of the error and run the job again. For example, if the error message indicates that the disk quota has been exceeded, clear disk space and the run the job again.

If the problem cannot be solved, contact the software application representative.

## Database Access Errors

Contact the DBA if the process aborted due to such errors as:

- Maximum number of processes exceeded
- Failed to connect to Oracle
- Table or view does not exist

### Missing or Invalid Data Errors

Contact the software application representative if the process aborted due to such errors as:

- No relevant data
- Invalid data
- Key not found

### Other Errors

Contact the software application representative concerning errors that cannot be classified as either database access errors or missing or invalid data errors (see preceding subsections).

## General Environment Settings

The following environment variables must be defined in an account:

| Name | Description | Example |
|---|---|---|
| GN1_ORA_USER | Oracle user | Acmapp2 |
| GN1_ORA_PASS | Oracle password | Acmapp2 |
| GN1_ORA_INST | Oracle instance | Ip8i |
| GN1_ORA_INST_NODE | Oracle instance node | Hobbit |
| ABP_PM_ROOT | The root directory for the Rating processes | Acmwrk2/var/acm/projs/pm |
| ABP_MARKET | Market ID | M3G |
| ABP_VAR | The root account path (determined using the PWD command) | Acmwrk2/var |
| ABP_LOG | The directory where all operational log files are created after the process runs | Acmwrk2/var/acm/log |
| ABP_AC_VFS_PATH | For A&C use | |
| ABP_DEBUG_FLAG | Y/N; opens or closes the A&C interface debugging prints (for private tests, should be Y) | AC_DEBUG_FLAG=Y |
| ABP_PRIVATE_BIN | Points to the directory where private executables and shared libraries are located ( usually ABP_home/AppBin) | ABP_home/AppBin |

# Changing Job Environment Variables

There are three ways to change a parameter for a job:

- If the variable is used by all Rating applications, the variable is defined in the op_pm_env_sh file (for example, `ABP_PM_CONF`, `RTR_IMPLEMENTATION_FILE`).
- A variable that is specific to a job is defined in the XML configuration file of the job (for example, the name or path of the log file).
- Configuration files.

Each process has a template configuration XML file. The template contains various variables that the job requires during run time (for example, where to write the log, to which data storage it needs to connect, etc.).

When the script is run, the script parameters are provided (for example, cycle code, DSN, etc.).

The script retrieves the parameters and performs the following steps:

1. The script calls an expand function that gets both the template configuration file and the "output" configuration file. (The name of the output configuration file comprises the template configuration file name and the parameters that the script received.)

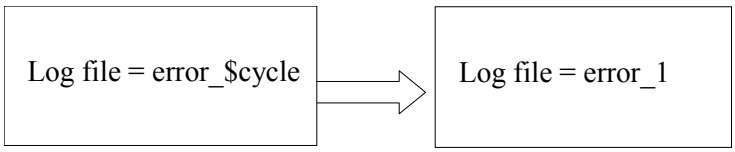
   For example, if the template file name is pm1RaterProcessConfig.xml (the Rater configuration file) and the script is run as follows:

   ```
   pm1PostpaidRater_sh 1 11 NRT
   ```

   Then the new configuration file is called:

   ```
   pm1RaterProcessConfig_1_11_NRT.autoXml.
   ```

2. The expand function changes all the variables in the new configuration file. For example:




3. The script calls the executable. The only argument that the executable gets is the configuration file. For example, if the Rater script is run twice, with different parameters each time, two configuration files will be created with:

| First Run | Second Run |
|---|---|
| cycle = 1 | cycle = 2 |
| task ID = 11 | task_ID = 21 |
| mode = RT | mode = RT |

   The first run is named pm1RaterProcessConfig _1_11_RT.autoXml and the second run is named pm1RaterProcessConfig _2_21_RT.autoXml.

There are also differences between the contents of the two files. In the template, there are several parameters that do not have values. Instead they refer to variables, for example, $RTR_CYCLE_CODE, RTR_TASK_ID.

For example, in the first run file above (PricingEngineConfigST_1_11_RT.autoXml), the number 1 replaces $RTR_CYCLE_CODE wherever it appears in the template.

In the second run file above (PricingEngineConfigST_2_21_RT.autoXml), the number 2 replaces $RTR_CYCLE_CODE wherever it appears in the template.

## Terminology

The following specialized terms are used in this document.

| Term | Definition |
|---|---|
| AMC | Application Monitoring and Control |
| Batch process | Same as batch job |
| Batch program | Same as batch job |
| DBA | Database administrator |
| DSN | Data store name |
| EOC | End of Cycle |

# 2. PM1RUNRATER - RATER

## Description

The Rater process can run in normal mode or in recovery mode. In recovery mode, the operational name of the Rater process is **PM1RECOVERY**.

### Normal Mode

The Rating component handles events that represent all types of usage of telecommunication services, and creates rated events for use by other subsystems. The main Rating functions are:

- Identifying the services according to the customer profile and marketing information.
- Matching the event to the correct services.
- Rating and distributing the event.

### Recovery Mode

Whenever Rating commits the results of its work to the database, the Recovery mechanism updates the table RPR1_POSTPAID_RECOVERY that maintains the last processed events in the transaction being rated.

After an abrupt termination for any reason (such as a power failure, hardware failure, or software failure), the Recovery process takes files in status 'IU' from A&C and creates a transaction for each file. In the Transaction Manager Algorithm that divides the file into buckets only the un-processed events will be taken from the file records (thus skipping previously processed events).

### Job Type

This paragraph contains the type of job (extract, read, write, etc.).

**Job Type:**

In its normal mode, this is a daemon

In its recovery mode, this job is a batch job

### Run Frequency

This paragraph defines how often the job runs; for example, daily, monthly, by request, etc.

**Run Frequency**:

- In normal mode, it is a daemon that is constantly running.
- In recovery mode, it is run by request

# Activation and Shutdown

## Job Activation

### Command Line

**Normal mode**

RunJobs PM1RUNRATER <JOB_REQ>

**Recovery mode**

RunJobs  PM1RECOVERY <JOB_REQ>

**AMC**: Yes

Screen Composer: Yes

### Script Name:

**Normal mode**

pm1PostpaidRater_sh

**Recovery mode**

pm1PostpaidRaterRecovery_sh

**Executable:** pm1EnvelopeMT

## Job Shutdown

**AMC**: Yes

Screen Composer: Yes

## Log Files

### Name

- Log File:POSTPAID<Partition ID>_<Sub Partition ID>_<thread ID>_Log_<date_time>.log
- Trace File :POSTPAID<Partition ID>_<Sub Partition ID>_<thread ID>_Trace_<date_time>.log
- ART error file: apperror_file_POSTPAID<Partition ID>_<Sub Partition ID>_<ART process number>_<date_time>.log
- ART events file: ART_Event_File_POSTPAID<Partition ID>_<Sub Partition ID>_<ART process number>_<date_time>.log
- ART standart output file: console_file_POSTPAID<Partition ID>_<Sub Partition ID>_<ART process number>_<date_time>.log
- Process stdout file: pm1EnvelopeMT_POSTPAID<Partition ID>_<Sub Partition ID>_<process ID>.log
- Pricing Engine Errors: pmPricingEngineError_POSTPAID<Partition ID>_<Sub Partition ID>.log
- Pricing Engine trace (debug mode only): Event<event_ID>.xml

  This file is created only if Tracing mode is enabled in the configuration file.

#### Location

The log files are stored in $ABP_PM_ROOT/work.

# Job Flow

## Run Level

This paragraph defines the level at which the job runs in the End of Day map (if relevant)

**Run Level:** N/A

## Preceding Jobs

This paragraph contains a list of the jobs that must be successfully performed **before** the job is run.

- A&F Main Driver
- Extract
- Load
- XLA Daemon (run before Rating to set the bookmarks in the XLA log file)

## Dependent Jobs

This paragraph contains a list of the jobs that can only be run after the successful completion of this batch job.

- Dispatcher
- PI and Event extract

## Concurrent Jobs

- XLA Daemon
- Dispatcher


*These processes must run all the time, to ensure that TimesTen does not fill the disk while no process updates the persistent storage (Oracle) with the rated events, and no process cleans the records from TimesTen storage.*

## Input Files

#### Naming Convention

The input file name (created by the Main Driver process) is:

MF1MDTORT_<partition_Id>_<cycle_code>_<task_Id>

#### Location

The location of the input file is taken from Audit and Control.

#### Contents

The input file contains event records.

# Flow

## Normal Mode

The Postpaid rater daemon is a multi-thread process built upon an ART server.

The process flow is divided among several components:

1. **Acceptor layer**: The Acceptor handles the input of the process. It pulls a block of files from A&C, creates a logical transaction and sends it to the Transaction manager layer.
2. **Transaction manager layer**: The Transaction manager breaks the logical transaction into independent working units called Request and sends them to the Service layer. Additionally, it handles the logical transaction database connections (USAGE and CUSTOMER) and closes the A&C files.
3. **Service layer**: This is a multi-thread layer. The Requests are dispatched to the thread queue. Each thread takes a Request from the queue and processes it (calculating the rate). The processing algorithm is as follows:
   - Establishes connections with the USAGE and CUSTOMER databases using the Transaction manager connection pool.
   - Gets an event from the Request.
   - Qualifies the event (guiding to service).
   - Rates the event.
   - Inserts a row into the PI USAGE data storage.
   - Writes the rated event and other external records to the DISPATCHER file.

In the process flow, records containing recovery information are written to the RPR1_POSTPAID_RECOVERY table. After the transaction is processed successfully, the records are deleted. If the transaction is not processed successfully, the process updates the entry in the table with the number of records that were already processed.

## Recovery Mode

The process flow is as that of the Normal mode except for the following changes:

- The Acceptor reads A&C files in recovery mode. This means that the Acceptor gets files in ‘IU’ mode. Each transaction contains only one file.
- Postpaid Transaction Recovery functionality: For each transaction item, the process gets recovery information from the RPR1_FILE_RECOVERY_DATA table and divides the files into buckets using the recovery number_of_buckets data (not the configuration partition key). It uses the recovery NUMBER_SUCCESS+NUMBER_ERROR data to remove all events that were handled by the previous process from the buckets.

## Output Files

### Naming Convention

Events file (for the dispatcher):

DISPATCH_FILE_ POSTPAID<Partition ID>_<Sub Partition ID>_<date time>_<serial number>.dat

### Location

- Pricing Engine trace (debug mode only): The file is stored in $ABP_PM_ROOT/work
- Events file (for the dispatcher): $ABP_PM_ROOT/data

# Parameters

| Parameter | Description |
|---|---|
| Partition | Partition code |
| Sub Partition | Unique ID to differentiate between raters working on the same partition |

# Databases

- Audit and Control and Product Catalog tables (Oracle) – History
- Rated Event data storage (TimesTen or Oracle) – Usage
- Customer data storage (TimesTen or Oracle) – customer

# Distribution

N/A

# Console Messages

N/A

# Troubleshooting

N/A

# Recovery Instructions


*For details see Recovery Specs Document*

# 3. PM1STOPRATER - STOP RATER

## Description

The Stop Rater process stops the Rater process.

### Job Type

This paragraph contains the type of job (extract, read, write, etc.).

**Job Type:** N/A

### Run Frequency

This paragraph defines how often the job runs; for example, daily, monthly, by request, etc.

**Run Frequency**: By request

### Activation and Shutdown

#### Job Activation

**Command Line:** `pm1StopRaterShell_sh 1 1`

**AMC**: Yes

Screen Composer: Yes

Script Name: pm1StopRater_sh

Executable: None

#### Job Shutdown

**AMC**: No

Screen Composer: No

### Log File

None

## Job Flow

### Run Level

This paragraph defines the level at which the job runs in the End of Day map (If relevant)

**Run Level:** N/A

### Preceding Processes

None

### Dependent Processes

None

### Input Files

None

### Flow

1. Finds the Rater PID (Process ID) that is running by the specified partition code and sub-partition code.

   If the process gets only the partition code as an input parameter, all the Rater processes of the partition will be stopped.
2. Stops the Rater process by sending a SIGTERM signal to the server. The server receives the signal and shuts down.

### Output Files

None.

## Parameters

| Parameter | Description |
|---|---|
| Partition | Partition code |
| Sub Partition | Unique ID to differentiate between raters working on the same partition |

## Databases

This paragraph specifies the databases the job uses, for example, *Usage1*, *Usage2*, *Customer*, etc.

N/A

## Distribution

NA

## Console Messages

NA

## Troubleshooting

NA

## Recovery Instructions

This paragraph contains a list of suggestions of how to recover the job.

N/A

# 4. PM1SUSPENDRTR/PM1RESUMERTR - SUSPEND/RESUME RATER

## Description

The Suspend/Resume Rater process suspends or resumes the operation of specific Acceptors inside the Rerate Map.

### Job Type

This paragraph contains the type of job (extract, read, write, etc.).

**Job Type: NA**

### Run Frequency

This paragraph defines how often the job runs; for example, daily, monthly, by request, etc.

**Run Frequency**: By request

### Activation and Shutdown

#### Job Activation

**Command Line**:

- pm1SuspendRaterShell_sh 1 1
- pm1ResumeRaterShell_sh 1 1

**AMC**: Yes

Screen Composer: No

Script Name:

- pm1SuspendRater_sh
- pm1ResumeRater_sh

Executable: None

#### Job Shutdown

**AMC**: Yes

Screen Composer: No

### Log File

None

# Job Flow

## Run Level

This paragraph defines the level at which the job runs in the End of Day map (If relevant)

**Run Level:** N/A

## Preceding Processes

None

## Dependent Processes

None

## Input Files

None

## Flow

1. Gets all cycles that should be suspended/resumed by the Map Key.
2. Sends a request to the Rater Process Administrator to suspend/resume all relevant acceptors.

## Output Files

None

# Parameters

| Parameter | Description |
|---|---|
| Map Key | Rerate map key |
| Partition ID | Unique ID to differentiate between raters working on the same partition. |

# Databases

This paragraph specifies the databases the job uses, for example, *Usage1*, *Usage2*, *Customer*, etc.

N/A

# Distribution

NA

# Console Messages

NA

# Troubleshooting

NA.

# Recovery Instructions

This paragraph contains a list of suggestions how to recover the job.

N/A

# 5. PM1CMGENEXT - CUSTOMER MANAGEMENT EXTRACT

## Description

The Customer Management (CM) Extract process synchronizes Rating data with CM data by extracting CM data to Rating tables.

The extract can run in the following modes/sub-mode:

| Mode | Sub Mode | Type | Purpose |
|---|---|---|---|
| CONFIGURATION | FULL | Batch | Extracts all customers' information from the Customer database according to the transactions in the configuration file |
| CONFIGURATION | FULLFILL | Batch | Extracts all customers' information from the Customer database according to the transactions in the configuration file. This sub mode requires as a pre-condition, empty Customer tables, because a process with SUB MODE=FULLFILL works only in insert mode (not update), which greatly improves performance. Amdocs recommends running Load in FULLFILL mode from the beginning or remember to clean the tables manually before the run. |
| CONFIGURATION | PROVIDER | Batch | Extracts provider's information from the Customer database according to the transactions in the configuration file |
| CONFIGURATION | MANUAL | Batch | Manually marks subscribers for re-rate |
| INCREMENTAL | DAEMON | Daemon | Extracts offers/parameters for a specific customer/subscriber according to the CM TRB transaction<br>Creates Activity for specific CM TRB transactions<br>Updates entries in the CUG table according to the CM TRB transaction<br>Inserts entries into the pm1_change_cycle table according to the CM TRB transaction<br>Insert entries into the PM1_CYCLE_STATE table according to the Billing TRB transaction |
| INCREMENTAL | HISTORY | Batch | Processes TRB transactions from the TRB History table according to the TRB_START and TRB_END parameters |

### Job Type

Batch or Daemon depending on the Run Mode/Sub-mode

## Run Frequency

On request

## Activation and Shutdown

### Job Activation

**Command Line**: Pm1CMGenericExtractShell_sh CONFIGURATION FULL

**AMC**: Yes

Screen Composer: Yes

Script Name: Pm1CMGenericExtract_sh

Executable: Pm1CMGenericExtractMT

### Job Shutdown

**AMC**: Yes

Screen Composer: Yes

## Log File

### Name

- Error File: pm.pm1CMGenericExtractMT_<Run Mode>_<Sub Run Mode>.<Date_Time>_Error.log
- Trace File:  pm.pm1CMGenericExtractMT_<Run Mode>_<Sub Run Mode>.<Date_Time>_Trace.log<Thread ID>
- ART error file: apperror_file_ RPRGEXT1_<ART process number>_<date_time>.log
- ART events file: ART_Event_File_RPRGEXT1_<ART process number>_<date_time>.log
- ART standard output file: console_file_RPRGEXT1_<ART process number>_<date_time>.log

### Location

The log file is stored in $ABP_PM_ROOT/work.

The log file should be checked after each run to verify that the process ran successfully.

# Job Flow

## Run Level

This paragraph defines the level at which the job runs in the End of Day map (If relevant)

**Run Level:** On request

## Preceding Processes

None

## Dependent Processes

These processes can only be run after the successful completion of this batch job:

- Load

## Input Files

None

## Flow

The process flow is as follows:

1. Extracts data from Customer Management tables/Creates Activities/Creates entries for generic tables (PM1_CHANGE_CYCLE, PM1_CYCLE_STATE)
2. Formats the data as XML
3. Saves the data in XML files, ready for the Loader process

## Output Files

This job has the following output file types:

- Subscriber Offers – The Loader inserts them into CUSTOMER_OFFERS
- Subscriber Parameters – The Loader inserts them into CUSTOMER_PARAMETERS
- Activities – The Loader processes each Activity
- Generic – The Loader inserts/deletes them into/from tables according to the information in the file

### File Name Convention

<File Type>-<DistributionCode >-<Serial number>-<Loader Run Mode>-<DateTime>.xml

Where:

- File Type – SubscriberOffers/SubscriberParameters/Activities/Generic
- DistributionCode – The Rating Partition code
- Serial number
- Loader Run Mode – F (full) or I (incremental)
- DateTime – The date and time when the file is created
- .xml – XML file extension

For example: SubscriberOffers-1-5-F-20040714173029.xml

### Location

$ABP_PM_ROOT/interfaces/input

# Parameters

The batch job uses the following parameters:

| Parameter | Description |
|---|---|
| Run mode | CONFIGURATION/INCREMENTAL |
| Sub-Run mode | FULL/PROVIDER/MANUAL/DAEMON/HISTORY |
| TRB start | Optional (only on INCREMENTAL/HISTORY) |
| TRB end | Optional (only on INCREMENTAL/HISTORY) |

# Databases

Audit and Control and CM tables(Oracle) – History

# Distribution

To each of the rater partitions, using AC

# Console Messages

NA

# Troubleshooting

NA

# Recovery Instructions

N/A

# 6. PM1LOAD - LOAD

## Description

The Customer Load process receives input customer XML files from Audit & Control.

The XML files contain subscriber offers/parameters/activities/generic data. They are read and analyzed, and the relevant subscriber objects are created and loaded into the customer entities in memory.

The Load process can run in either of the following two modes:

- Full Load – The full load runs after the full extract. The full load is a by-request job.
- Incremental Load - The incremental load is a daemon.

### Job Type

This paragraph contains the type of job (extract, read, write, etc.).

**Job Type:** The incremental Load is a daemon.

### Run Frequency

The full-mode Load is run by request.

The incremental Load is a daemon which runs constantly

### Activation and Shutdown

#### Job Activation

**Command Line**: pm1CM2RaterLoadShell_sh FULL 1

**AMC**: Yes

Screen Composer: Yes

Script Name: pm1CM2RaterLoad_sh

Executable: pm1CM2RaterLoad

#### Job Shutdown

**AMC**: Yes

Screen Composer: Yes

### Log File

#### Name

- Error file: pm.pm1CM2RaterLoad_<mode>_<Partition ID>_<Process Instance>_<date_time>_Error.log
- Trace file: pm.pm1CM2RaterLoad_<mode>_<Partition ID>_<Process Instance>_<date_time>_Trace.log

### Location

The log file is stored in $ABP_PM_ROOT/work.

# Job Flow

## Run Level

This paragraph defines the level at which the job runs in the End of Day map (If relevant)

**Run Level:** NA

## Preceding Processes

This paragraph contains a list of the jobs that must be successfully performed **before** the job is run.

**Preceding jobs:** Extract

## Dependent Processes

This paragraph contains a list of the jobs that can only be run after the successful completion of this batch job.

**Dependent jobs:** Rating

## Input Files

See extract output files

The input files are located in $ABP_PM_ROOT/interfaces/input

## Flow

1. Retrieves the input files from A&C tables
2. Initializes the Pricing Engine
3. Parses each file:
   a. Offers – Creates a subscriber offers entity
   b. Parameters – Creates a subscriber parameters entity
   c. Offers' parameters – Creates a customer offers parameters entity
   d. Generic – Prepares an SQL command to insert/delete data from tables
   e. Activity – Prepares an event entity.
3. Loads offers/parameters/offers' parameters entities to tables using PE APIs
4. Processes the activity event
5. Performs an SQL query for the generic table

## Output Files

None

## Parameters

| Parameter | Description |
|---|---|
| Partition ID | The code of the rating partition |
| Mode | Full or Incremental |

## Databases

- Audit and Control and Product Catalog tables (Oracle) – History
- Rated Event data storage (TimesTen or Oracle) – Usage
- Customer data storage (TimesTen or Oracle) – Customer

## Distribution

NA

## Console Messages

N/A

## Troubleshooting

N/A

## Recovery Instructions

The process automatically handles IU entries in AC

# 7. PM1STOPCMGEXT - STOP INCREMENTAL EXTRACT

## Description

The Stop Incremental Extract batch job stops the Customer Management Extract process in incremental mode.

### Job Type

This paragraph contains the type of job (extract, read, write, etc.).

**Job Type:** NA

### Run Frequency

This job runs by-request

### Activation and Shutdown

#### Job Activation

**Command Line**: pm1StopCMGextShell_sh

**AMC**: Yes

Screen Composer: Yes

Script Name: pm1StopCMGext_sh

Executable: None

#### Job Shutdown

**AMC**: Yes

Screen Composer: Yes

### Log File

None

## Job Flow

### Run Level

This paragraph defines the level at which the job runs in the End of Day map (If relevant)

**Run Level:** N/A

### Preceding Processes

None

### Dependent Processes

None

### Input Files

None.

### Flow

The process includes the following steps:

1. Finds the PID of the Extract process that is running in incremental mode.
2. Kills the process by sending a USR1 signal.

### Output Files

None

## Parameters

None

## Databases

This paragraph specifies the databases the job uses, for example, *Usage1*, *Usage2*, *Customer*, etc.

N/A

## Distribution

NA

## Console Messages

NA

## Troubleshooting

NA

## Recovery Instructions

NA

# 8. PM1STOPCMLD - STOP INCREMENTAL LOAD

## Description

The Stop Incremental Load process stops the Load process when running in incremental mode.

### Job Type

This paragraph contains the type of job (extract, read, write, etc.).

**Job Type:** NA

### Run Frequency

This is a By-Request job.

### Activation and Shutdown

#### Job Activation

**Command Line**: pm1StopCM2RaterLoadShell_sh INCREMENTAL 1

**AMC**: Yes

Screen Composer: Yes

Script Name: pm1StopCM2RaterLoad_sh

Executable: N/A

#### Job Shutdown

**AMC**: Yes

Screen Composer: Yes

### Log File

None

## Job Flow

### Run Level

This paragraph defines the level at which the job runs in the End of Day map (If relevant)

**Run Level:** N/A

### Preceding Processes

None

### Dependent Processes

None

### Input Files

None.

### Flow

The process does the following:

1. Finds the PID of the Load process that is running in incremental mode.
2. Kills the process by sending a USR1 signal.

### Output Files

None

## Parameters

The batch job uses the following parameters:

| Parameter | Description |
|---|---|
| Run Mode | FULL/INCREMENTAL |
| Partition ID | Rating Partition ID |

## Databases

This paragraph specifies the databases the job uses, for example, *Usage1*, *Usage2*, *Customer*, etc.

NA

## Distribution

NA

## Console Messages

NA

## Troubleshooting

This paragraph contains a list of suggestions that can explain why the job did not run successfully or to prevent it from failing.

NA

## Recovery Instructions

This paragraph contains a list of suggestions how to recover the job.

NA

# 9. PM1UEXTEVENT - EVENT EXTRACT

## Description

This job extracts rated records from the Rated Events table. The same server handles the cycle mode and the customer mode.

### Job Type

This paragraph contains the type of job (extract, read, write, etc.).

**Job Type:** Daemon

### Run Frequency

This paragraph defines how often the job runs; for example, daily, monthly, by request, etc.

**Run Frequency**: This job is a daemon and it runs constantly.

### Activation and Shutdown

#### Job Activation

**Command Line**: pm1RunEventExtractShell_sh 1

**AMC**: Yes

Screen Composer: Yes

Script Name: pm1RunEventExtract_sh

Executable: pm1EnvelopeMT

#### Job Shutdown

**AMC**: Yes

Screen Composer: Yes

### Log File

#### Name

- Pricing Engine Errors: pmPricingEngineError_UEXT_E_CY_<Partition ID>.log
- Log File: UEXT_E_CY_<Partition ID>_<thread ID>_Log_<date_time>.log
- Trace File: UEXT_E_CY_<Partition ID>_<thread ID>_Trace_<date_time>.log
- ART error file: apperror_file_UEXT_E_CY_<Partition ID>_<ART process number>_<date_time>.log

- ART events file: ART_Event_File_UEXT_E_CY_<Partition ID>_<ART process number>_<date_time>.log
- ART standard output file: console_file_UEXT_E_CY_<Partition ID>_ART process number>_<date_time>.log
- Process stdout file: pm1EnvelopeMT_UEXT_E_CY_<Partition ID>_<process ID>.log

### Location

$ABP_PM_ROOT/work.

# Job Flow

## Run Level

This paragraph defines the level at which the job runs in the End of Day map (If relevant)

**Run Level:** N/A

## Preceding Jobs

This paragraph contains a list of the jobs that must be successfully performed **before** the job is run.

**Preceding jobs:** Dispatcher

## Dependent Jobs

This paragraph contains a list of the jobs that can only be run after the successful completion of this job.

**Dependent jobs:** None

## Input Files

bl.<cycle code><cycle year><cycle month><flow id>_<route><group id><data element>.YYYYMMDD_HHMMSS.temp.txt

## Flow

The job performs the following steps:

1. Connects to the Oracle database
2. Extracts the records from the Rated Events table
3. Writes the records to a file

## Output Files

### Naming Convention

Cycle:

Data_Event_File_<cycleCode>_<cycleIns>_<cycleYear>_<datetime>_<threadID>_<fileIndex>

Customer:

Data_Event_File_<datetime>_<threadID>

Error_Event_File_<datetime>_<threadID>

### Location

$ABP_PM_ROOT/interfaces/output

# Parameters

| Parameter | Description |
|---|---|
| Partition ID | Rating partition ID |

# Databases

- Audit and Control and Product Catalog tables (Oracle) – History
- Rated Event data storage (TimesTen or Oracle) – Usage

# Distribution

In customer mode the files are distributed to billing machine using AC.

# Console Messages

NA

# Troubleshooting

NA

# Recovery Instructions

NA

# 10. PM1UEXTPI - PI EXTRACT

## Description

This job extracts Performance indicators' records from the Performance Indicators table. The same server handles the cycle mode and the customer mode.

### Job Type

This paragraph contains the type of job (extract, read, write, etc.).

**Job Type:** Daemon

### Run Frequency

This paragraph defines how often the job runs; for example, daily, monthly, by request, etc.

**Run Frequency**: This job is constantly running.

### Activation and Shutdown

#### Job Activation

**Command Line**: pm1RunPIExtractShell_sh 1

**AMC**: Yes

Screen Composer: Yes

Script Name: pm1RunPIExtract_sh

Executable: pm1EnvelopeMT

#### Job Shutdown

**AMC**: Yes

Screen Composer: Yes

### Log File

#### Name

- Pricing Engine Errors: pmPricingEngineError_UEXT_P_CY_<Partition ID>.log
- Log File: UEXT_P_CY_<Partition ID>_<thread ID>_Log_<date_time>.log
- Trace File: UEXT_P_CY_<Partition ID>_<thread ID>_Trace_<date_time>.log
- ART error file: apperror_file_UEXT_P_CY_<Partition ID>_<ART process number>_<date_time>.log
- ART events file: ART_Event_File_UEXT_P_CY_<Partition ID>_<ART process number>_<date_time>.log

- ART standard output file: console_file_UEXT_P_CY_<Partition ID>_ART process number>_<date_time>.log
- Process stdout file: pm1EnvelopeMT_UEXT_P_CY_<Partition ID>_<process ID>.log

#### Location

$ABP_PM_ROOT/work.

# Job Flow

## Run Level

This paragraph defines the level at which the job runs in the End of Day map (If relevant)

**Run Level:** N/A

## Preceding Jobs

None

## Dependent Jobs

None

## Input Files

bl.<cycle code><cycle year><cycle month><flow id>_<route><group id><data element>.YYYYMMDD_HHMMSS.temp.txt

## Flow

The process does the following:

1. Connects to the Oracle or TT database
2. Extracts the records from the Performance Indicators table
3. Writes the records to a file

## Output Files

#### Naming Convention

**Cycle:**

Data_PI_File_<cycleCode>_<cycleIns>_<cycleYear>_<datetime>_<threadID>_<fileIndex>

**Customer:**

Data_PI_File_<datetime>_<threadID>

Error_PI_File_<datetime>_<threadID>

#### Location

$ABP_PM_ROOT/interfaces/output

## Parameters

| Parameter | Description |
|---|---|
| Partition ID | The partition ID |

## Databases

- Audit and Control and Product Catalog tables (Oracle) – History
- Rated Event data storage (TimesTen or Oracle) – Usage

## Distribution

NA

## Console Messages

NA

## Troubleshooting

This paragraph contains a list of suggestions that can explain why the job did not run successfully or to prevent it from failing.

NA

## Recovery Instructions

This paragraph contains a list of suggestions how to recover the job.

NA

# 11. PM1EVEXT - CYCLE EVENT EXTRACT REQUEST

## Description

The Cycle Event Extract Request job sends a message to the event extract server to perform an extract for a given cycle code, instance, year and an additional Where clause.

### Job Type

This paragraph contains the type of job (extract, read, write, etc.).

**Job Type:** extract

### Run Frequency

This paragraph defines how often the job runs; for example, daily, monthly, by request, etc.

**Run Frequency**: By request

### Activation and Shutdown

#### Job Activation

**Command Line**: pm1EventExtractShell_sh 1 1 2 2004

**pm1EventExtractShell_sh 1 3 5 2005 none none none none MyAlias none none MyDataGroup**

**AMC**: No

Screen Composer: Yes

Script Name: pm1EventExtract_sh

Executable: ARTServerAdmin

#### Job Shutdown

**AMC**: No

Screen Composer: Yes

### Log File

None

## Job Flow

### Run Level

This paragraph defines the level at which the job runs in the End of Day map (If relevant)

**Run Level:** N/A

## Preceding Jobs

This paragraph contains a list of the jobs that must be successfully performed **before** the job is run.

**Preceding jobs:** Event Extract

## Dependent Jobs

This paragraph contains a list of the jobs that can only be run after the successful completion of this job.

**Dependent jobs:** None

## Input Files

None

## Flow

The job sends a message to the server.

## Output Files

None

# Parameters

| Parameter | Description |
|---|---|
| Partition ID | Rating Partition |
| Cycle code | The cycle code |
| Cycle instance | The cycle instance |
| Cycle year | The cycle year |
| *From Date Time | The From date of the required events |
| *To Date Time | The To date of the required events |
| *Additional where clause | The customization Where clause |
| *Routing Criteria | The output routing criteria |
| *Output File Alias | The output file alias |
| *Output File Name | The output file name |
| *Output File Path | The output file path |
| *Output File Data Group | The output file data group |

* These parameters are not mandatory. If none of them have a value, they can be left empty. If at least one is set to a value, they all become mandatory and must be set according to the following criteria:

- If any of these parameters is set to “none”, the default values of these parameters are taken from the GN1_ART_SECTION_PARAM table.
- If at least one of these parameters is populated with a value, the remaining parameters must be set to “none”.

## Databases

None

## Distribution

NA

## Console Messages

NA

## Troubleshooting

This paragraph contains a list of suggestions that can explain why the job did not run successfully or to prevent it from failing.

NA

## Recovery Instructions

This paragraph contains a list of suggestions how to recover the job.

NA

# 12. PM1PIEXT - CYCLE PI EXTRACT REQUEST

## Description

The PI Extract job sends a message to the server to perform an extract for a given cycle code, instance, year and an additional Where clause.

### Job Type

This paragraph contains the type of job (extract, read, write, etc.).

**Job Type:** extract

### Run Frequency

This paragraph defines how often the job runs; for example, daily, monthly, by request, etc.

**Run Frequency**: By Request

### Activation and Shutdown

#### Job Activation

**Command Line**: pm1PIExtractShell_sh 5 1 2 2004 10

**AMC**: No

Screen Composer: Yes

Script Name: pm1PIExtract_sh

Executable: ARTServerAdmin

#### Job Shutdown

**AMC**: No

Screen Composer: Yes

### Log File

None

## Job Flow

### Run Level

This paragraph defines the level at which the job runs in the End of Day map (If relevant)

**Run Level:** N/A

## Preceding Jobs

This paragraph contains a list of the jobs that must be successfully performed **before** the job is run.

**Preceding jobs:** PI Extract

## Dependent Jobs

This paragraph contains a list of the jobs that can only be run after the successful completion of this job.

**Dependent jobs:** None

## Input Files

None

## Flow

The job sends a message to the server.

## Output Files

None

# Parameters

| Parameter | Description |
|---|---|
| Partition ID | The partition |
| Cycle code | The cycle code |
| Cycle instance | The cycle instance |
| Cycle year | The cycle year |
| Number in Parallel | Number in parallel threads to handle the request. Each thread creates its own output. |
| *From Date Time | The From date of the required events |
| *To Date Time | The To date of the required events |
| *Additional where clause | The customization Where clause |
| *Routing Criteria | The output routing criteria |
| *Output File Alias | The output file alias |
| *Output File Name | The output file name |
| *Output File Path | The output file path |
| *Output File Data Group | The output file data group |

* These parameters are not mandatory. If none of them have a value, they can be left empty. If at least one is set to a value, they all become mandatory and must be set according to the following criteria:

- If any of these parameters is set to “none”, the default values of these parameters are taken from the GN1_ART_SECTION_PARAM table.
- If at least one of these parameters is populated with a value, the remaining parameters must be set to “none”.

## Databases

None

## Distribution

NA

## Console Messages

NA

## Troubleshooting

NA

## Recovery Instructions

NA

# 13. PM1RCVCYCLEXT - PI/EVENT EXTRACT RECOVERY CYCLE MODE REQUEST

## Description

This job sends a recovery message to the PI or Event extract daemon server. The cycle recovery is on the partition level.

### Job Type

This paragraph contains the type of job (extract, read, write, etc.).

**Job Type:** NA

### Run Frequency

This paragraph defines how often the job runs; for example, daily, monthly, by request, etc.

**Run Frequency**: By request

### Activation and Shutdown

#### Job Activation

**Command Line**: pm1CycleRecoveryExtractShell_sh PI 5 1 2 2004 10

**AMC**: No

Screen Composer: Yes

Script Name: pm1CycleRecoveryExtract_sh

Executable: ARTServerAdmin

#### Job Shutdown

**AMC**: No

Screen Composer: Yes

### Log File

None

# Job Flow

## Run Level

This paragraph defines the level at which the job runs in the End of Day map (If relevant)

**Run Level:** N/A

## Preceding Jobs

This paragraph contains a list of the jobs that must be successfully performed **before** the job is run.

**Preceding jobs:** PI/Event extract server

## Dependent Jobs

This paragraph contains a list of the jobs that can only be run after the successful completion of this job.

**Dependent jobs:** None

## Input Files

None

## Flow

The job sends a message to the server.

## Output Files

None

# Parameters

| Parameter | Description |
|---|---|
| Extract Mode | PI or Event |
| Partition ID | The partition |
| Cycle code | The cycle code |
| Cycle instance | The cycle instance |
| Cycle year | The cycle year |
| Number in Parallel | Number in parallel threads to handle the request. Each tread create is one output. |
| *From Date Time | The from date of the required events |
| *To Date Time | The to date of the required events |
| *Additional where clause | The customization Where clause |
| *Routing Criteria | The output routing criteria |
| *Output File Alias | The output file alias |
| *Output File Name | The output file name |
| *Output File Path | The output file path |

| Parameter | Description |
|---|---|
| *Output File Data Group | The output file data group |

* These parameters are not mandatory. If none of them have a value, they can be left empty. If at least one is set to a value, they all become mandatory and must be set according to the following criteria:

- If any of these parameters is set to “none”, the default values of these parameters are taken from the GN1_ART_SECTION_PARAM table.
- If at least one of these parameters is populated with a value, the remaining parameters must be set to “none”.

## Databases

None

## Distribution

NA

## Console Messages

NA

## Troubleshooting

NA

## Recovery Instructions

NA

# 14. PM1RCVCUSTEXT - PI/Event Extract Recovery Customer Mode

## Description

This job runs the PI or Event server in customer recovery mode. The server goes down when the recovery is done.

### Job Type

This paragraph contains the type of job (extract, read, write, etc.).

**Job Type:** NA

### Run Frequency

This paragraph defines how often the job runs; for example, daily, monthly, by request, etc.

**Run Frequency**: By request

### Activation and Shutdown

#### Job Activation

**Command Line**: pm1CustomerRecoveryExtractShell_sh PI 5

**AMC**: No

Screen Composer: Yes

Script Name: pm1CustomerRecoveryExtract_sh

Executable: pm1EnvelopeMT

#### Job Shutdown

**AMC**: No

Screen Composer: Yes

### Log File

Same as for PI/Event Extract servers

## Job Flow

### Run Level

This paragraph defines the level at which the job runs in the End of Day map (If relevant)

**Run Level:** N/A

### Preceding Jobs

None

### Dependent Jobs

None

### Input Files

None

### Flow

The flow is the same as PI Extract and Event Extract in recovery mode.

### Output Files

#### Naming Convention

**PI:**

Data_PI_File_<datetime>_<threadID>

Error_PI_File_<datetime>_<threadID>

**Event:**

Data_Event_File_<datetime>_<threadID>

Error_Event_File_<datetime>_<threadID>

#### Location

$ABP_PM_ROOT/interfaces/output

## Parameters

The batch job uses the following parameters:

| Parameter | Description |
|---|---|
| Partition ID | The partition ID |
| Extract Mode | PI or Event |

## Databases

- Audit and Control and Product Catalog tables (Oracle) – History
- Rated Event data storage (TimesTen or Oracle) – Usage
- Customer data storage (TT or Oracle) – Customer

## Distribution

NA

## Console Messages

NA

# Troubleshooting

This paragraph contains a list of suggestions that can explain why the job did not run successfully or to prevent it from failing.

NA

# Recovery Instructions

This paragraph contains a list of suggestions how to recover the job.

NA

# 15. PM1POREREXT - POST RE-RATE EXTRACT REQUEST

## Description

This job sends a message to the PI and Event extract daemon server. The server adds a new acceptor, which processes the files of the customer that have already been re-rated.

### Job Type

This paragraph contains the type of job (extract, read, write, etc.).

**Job Type:** NA

### Run Frequency

This paragraph defines how often the job runs; for example, daily, monthly, by request, etc.

**Run Frequency**: By request. This process is part of the re-rate map.

### Activation and Shutdown

#### Job Activation

**Command Line**: pm1PostReRateExtractShell_sh 1 1_1

AMC: **Yes** as part of the re-rate map.

**Screen Composer:** No

**Script Name:** pm1PostReRateExtractShell_sh 1 1_1

**Executable:** ARTServerAdmin

#### Job Shutdown

**AMC**: Yes

Screen Composer: No

### Log File

None

## Job Flow

### Run Level

This paragraph defines the level at which the job runs in the End of Day map (If relevant)

**Run Level:** NA

### Preceding Jobs

This paragraph contains a list of the jobs that must be successfully performed **before** the job is run.

**Preceding jobs:** PI/Event Extract Server

### Dependent Jobs

This paragraph contains a list of the jobs that can only be run after the successful completion of this job.

**Dependent jobs:** None

### Input Files

None

### Flow

The job sends a message to the server.

### Output Files

None

## Parameters

| Parameter | Description |
|---|---|
| Partition ID | Rating partition code |
| Map key | Rerate Map Key - The data group prefix for the input files |

## Databases

None

## Distribution

Billing machine

## Console Messages

NA

## Troubleshooting

This paragraph contains a list of suggestions that can explain why the job did not run successfully or to prevent it from failing.

When running a request in customer mode, if the server falls with an error in Parsing, check that the input to the server is in the new format as seen in the pm1CustomerUsageExtarctInputFileFormat.xml. For example,

and
SYSDATE=SYSDATE;myrouting;;MYNAME;/dmhuser1.p712/dmh/users/dmhwrk12/var/m3g/projs/pm/interfaces/;DATAGROOP;Event Format;1;4;2004;0;0;0;

;;;;;;Event Format;1;5;2004;0;0;0;

;;;;;;;1;5;2004;3;1;3;

## Recovery Instructions

This paragraph contains a list of suggestions how to recover the job. NA

# 16. PM1USAGEQUERY - USAGE QUERY

## Description

The server performs queries on rated event and the Performance Indicators table. The result is placed into an xml file and passed to WebLogic.

### Job Type

Daemon

### Run Frequency

This paragraph defines how often the job runs; for example, daily, monthly, by request, etc.

**Run Frequency**: Runs constantly

### Activation and Shutdown

#### Job Activation

**Command Line**: pm1RunUsageQueryExtractShell_sh 1

**AMC**: Yes

Screen Composer: Yes

Script Name: pm1RunUsageQueryExtract_sh

Executable: pm1EnvelopeMT

#### Job Shutdown

**AMC**: Yes

Screen Composer: Yes

### Log File

#### Name

- Pricing Engine Errors: pmPricingEngineError_UEXT_UQ_<Partition ID>.log
- Log File: UEXT_UQ_<Partition ID>_<thread ID>_Log_<date_time>.log
- Trace File: UEXT_UQ_<Partition ID>_<thread ID>_Trace_<date_time>.log
- ART error file: apperror_file_UEXT_UQ_<Partition ID>_<ART process number>_<date_time>.log
- ART events file: ART_Event_File_UEXT_UQ_<Partition ID>_<ART process number>_<date_time>.log

- ART standard output file: console_file_UEXT_UQ_<Partition ID>_ART process number>_<date_time>.log
- Process stdout file: pm1EnvelopeMT_UEXT_UQ_<Partition ID>_<process ID>.log

### Location

$ABP_PM_ROOT/work.

# Job Flow

## Run Level

This paragraph defines the level at which the job runs in the End of Day map (If relevant)

**Run Level:** N/A

## Preceding Jobs

None

## Dependent Jobs

None

## Input Files

None

## Flow

1. Connects to the Oracle/TT database
6. Extracts the records from the Rated Events or PerformanceInd table
7. Sends the result to WebLogic

## Output Files

None

# Parameters

| Parameter | Description |
|---|---|
| Partition ID | The partition ID |

# Databases

- Audit and Control and Product Catalog tables (Oracle) – History
- Rated Event data storage (TimesTen or Oracle) – Usage

# Distribution

NA

## Console Messages

NA

## Troubleshooting

This paragraph contains a list of suggestions that can explain why the job did not run successfully or to prevent it from failing.

NA

## Recovery Instructions

This paragraph contains a list of suggestions how to recover the job.

NA

# 17. STOP EVENT/PI/USAGE QUERY EXTRACT

## Description

This job stops the PI/event/usage query extract process.

The process is defined in the operational system according to the entity it stops:

For event stopping- PM1STOPEVUXT

For PI stopping - PM1STOPPIUXT

For Usage Query stopping - PM1STOPUQUXT

### Job Type

This paragraph contains the type of job (extract, read, write, etc.).

**Job Type:** NA

### Run Frequency

This paragraph defines how often the job runs; for example, daily, monthly, by request, etc.

**Run Frequency**: By request

### Activation and Shutdown

#### Job Activation

**Command Line**: pm1StopPIExtractShell_sh 1

**AMC**: Yes

Screen Composer: Yes

Script Name:

- Event - pm1StopEventExtract_sh
- PI - pm1StopPIExtract_sh
- Usage Query - pm1StopUsageQuery_sh

Executable: None

#### Job Shutdown

**AMC**: Yes

Screen Composer: Yes

### Log File

None

# Job Flow

## Run Level

This paragraph defines the level at which the job runs in the End of Day map (If relevant)

**Run Level:** N/A

## Preceding Jobs

None

## Dependent Jobs

None

## Input Files

None

## Flow

The process sends a signal to the server.

## Output Files

None

# Parameters

| Parameter | Description |
|---|---|
| Partition ID | The partition ID |

# Databases

This paragraph specifies the databases the job uses, for example, *Usage1*, *Usage2*, *Customer*, etc.

NA

# Distribution

NA

# Console Messages

NA

# Troubleshooting

This paragraph contains a list of suggestions that can explain why the job did not run successfully or to prevent it from failing.

NA

## Recovery Instructions

This paragraph contains a list of suggestions how to recover the job.

NA

# 18. PM1GENMANMARK - MANUAL MARKING

## Description

This job manually marks subscribers for re-rating.

### Job Type

This paragraph contains the type of job (extract, read, write, etc.).

**Job Type:** extract

### Run Frequency

This paragraph defines how often the job runs; for example, daily, monthly, by request, etc.

**Run Frequency**: By request

### Activation and Shutdown

#### Job Activation

**Command Line**: pm1CMGenericExtractManualMarkingShell_sh '2' '19619,19620' '2002-01-01' '2005-12-31'

**AMC**: No

Screen Composer: Yes

Script Name: pm1CMGenericExtractManualMarking_sh

Executable: gext1GenericExtractMT

#### Job Shutdown

**AMC**: No

Screen Composer: Yes

### Log File

#### Name

Same as for CM Extract process

#### Location

The log file is stored in $ABP_PM_ROOT/work.

# Job Flow

## Run Level

This paragraph defines the level at which the job runs in the End of Day map (If relevant)

**Run Level:** N/A

## Preceding Jobs

This paragraph contains a list of the jobs that must be successfully performed **before** the job is run.

**Preceding jobs:** None

## Dependent Jobs

This paragraph contains a list of the jobs that can only be run after the successful completion of this job.

**Dependent jobs:** Manual Marking Load

## Input Files

None

## Flow

The process does the following:

1. Activates the CM generic extract process in mode CONFIGURATION/MANUAL
2. Creates Activities for all customers that own one of the offers in the parameters list.

## Output Files

### Naming Convention

Same as for CM Extract process

### Location

$ABP_PM_ROOT/interfaces/output

# Parameters

| Parameter | Description |
|---|---|
| Cycle code | The code of the cycle |
| Offer list | Offer ID 1, Offer ID 2, etc. |
| From date | Start of the date range |
| To date | End of the date range |

## Databases

Audit and Control and CM tables(Oracle) – History

## Distribution

NA

## Console Messages

NA

## Troubleshooting

This paragraph contains a list of suggestions that can explain why the job did not run successfully or to prevent it from failing.

NA

## Recovery Instructions

This paragraph contains a list of suggestions how to recover the job.

NA

# 19. PM1MANUALLOAD - MANUAL MARKING LOAD

## Description

The Manual Marking Load process loads the XML output of the manual marking process to the Subscriber Rerate table.

### Job Type

This paragraph contains the type of job (extract, read, write, etc.).

**Job Type:** NA

### Run Frequency

This paragraph defines how often the job runs; for example, daily, monthly, by request, etc.

**Run Frequency**: By request

### Activation and Shutdown

#### Job Activation

**Command Line**: pm1ManualMarkingLoadShell_sh 1 12 2002

**AMC**: No

Screen Composer: Yes

Script Name: pm1ManualMarkingLoad_sh

Executable: pm1CM2RaterLoad

#### Job Shutdown

**AMC**: No

Screen Composer: Yes

### Log File

#### Name

Same as for the Load process

#### Location

The log file is stored in $ABP_PM_ROOT/work.

# Job Flow

## Run Level

This paragraph defines the level at which the job runs in the End of Day map (If relevant)

**Run Level:** N/A

## Preceding Jobs

This paragraph contains a list of the jobs that must be successfully performed **before** the job is run.

**Preceding jobs:** Manual Marking Extract

## Dependent Jobs

This paragraph contains a list of the jobs that can only be run after the successful completion of this job.

**Dependent jobs:** None

## Input Files

### Naming Convention

Same as for the load process

### Location

$ABP_PM_ROOT/interfaces/output

## Flow

Same flow as for the Load process (the only type of files in this case is Activities files)

## Output Files

None

# Parameters

| Parameter | Description |
|---|---|
| Partition ID | Rating partition ID |
| Cycle code | The code of the cycle |
| Cycle month | The month of the cycle |
| Cycle year | The year of the cycle |

# Databases

- Audit and Control and Product Catalog tables (Oracle) – History
- Rated Event data storage (TimesTen or Oracle) – Usage
- Customer data storage (TimesTen or Oracle) – Customer

## Distribution

NA

## Console Messages

NA

## Troubleshooting

This paragraph contains a list of suggestions that can explain why the job did not run successfully or to prevent it from failing.

NA

## Recovery Instructions

This paragraph contains a list of suggestions how to recover the job.

NA

# 20. PM1REREXTRACT - RERATE EXTRACT

## Description

This job extracts records from the Performance Indicator and Rated Event tables.

### Job Type

This paragraph contains the type of job (extract, read, write, etc.).

**Job Type:** Extract

### Run Frequency

This paragraph defines how often the job runs; for example, daily, monthly, by request, etc.

**Run Frequency**: By request

### Activation and Shutdown

#### Job Activation

**Command Line**: pm1RerateExtractShell_sh 1 1 5

**AMC**: Yes as part of the Rerate map.

**Screen Composer:** No

**Script Name:** Script: pm1RerateExtract_sh

**Executable:** pm1RerateExtract

#### Job Shutdown

**AMC**: Yes

Screen Composer: No

### Log File

#### Name

Error: Pm.pm1RerateExtract_<Map Key>_<Partition ID>_<Process Instance>.<date_time>_Error.log

Trace: Pm.pm1RerateExtract_<Map Key>_<Partition ID>_<Process Instance>.<date_time>_Trace.log

#### Location

The log file is stored in $ABP_PM_ROOT/work.

# Job Flow

## Run Level

This paragraph defines the level at which the job runs in the End of Day map (If relevant)

**Run Level:** NA

## Preceding Jobs

This paragraph contains a list of the jobs that must be successfully performed **before** the job is run.

**Preceding jobs:** None

## Dependent Jobs

This paragraph contains a list of the jobs that can only be run after the successful completion of this job.

**Dependent jobs:** Rerating

## Input Files

None

## Flow

1. Fetches the relevant run ID from the RPR1_REREXT_DETAILS table
2. Gets all relevant subscribers from the RPR1_SUBS_RERATE table
3. Extracts all the rated events and PIs into files
4. Directs the PI and event files to Rerate in case of rerate
5. Directs the event file to Guiding in case of guiding.

## Output Files

### Naming Convention

Event-<date>-<sequence number>

PI-<date>-<sequence number>

### Location

$ABP_PM_ROOT/work

# Parameters

| Parameter | Description |
|---|---|
| Map Key | Rerate Map Key |
| Partiton ID | Rating Partition ID |
| Number in Parallel | Number of Rerate Extract processes to be executed in parallel |

## Databases

- Audit and Control and Product Catalog tables (Oracle) – History
- Rated Event data storage (TimesTen or Oracle) – Usage
- Customer data storage (TimesTen or Oracle) – Customer

## Distribution

NA

## Console Messages

NA

## Troubleshooting

This paragraph contains a list of suggestions that can explain why the job did not run successfully or to prevent it from failing.

NA

## Recovery Instructions


*For details see Rating Recovery Specs document*

# 21. PM1RERATE - RERATE

## Description

The Rerate process re-rates events for a group of marked subscribers.

### Job Type

This paragraph contains the type of job (extract, read, write, etc.).

**Job Type:** NA

### Run Frequency

This paragraph defines how often the job runs; for example, daily, monthly, by request, etc.

**Run Frequency**: By request

### Activation and Shutdown

#### Job Activation

**Command Line**: pm1ReraterShell_sh 1 1 8

**AMC**: Yes, as part of the rerate map

**Screen Composer:** No

**Script Name:** pm1Rerater_sh

**Executable:** pm1Rerater

#### Job Shutdown

**AMC**: Yes

Screen Composer: No

### Log File

#### Name

Error: Pm.pm1Rerate_<Map Key>_<Partition ID>_<Process Instance>.<date_time>_Error.log

Trace: Pm.pm1Rerate_<Map Key>_<Partition ID>_<Process Instance>.<date_time>_Trace.log

#### Location

$ABP_PM_ROOT/work.

# Job Flow

## Run Level

This paragraph defines the level at which the job runs in the End of Day map (If relevant)

**Run Level:** N/A

## Preceding Jobs

This paragraph contains a list of the jobs that must be successfully performed **before** the job is run.

**Preceding jobs:**

- Rating extract
- Reguiding

## Dependent Jobs

This paragraph contains a list of the jobs that can only be run after the successful completion of this job.

**Dependent jobs:** None

## Input Files

None

## Flow

- For all pairs (PI & Event files):
  - Fetches pairs of PI and Events files.
  - Re-initializes the PI and rate the events
- If Re-guiding finishes processing all events files
  - Fetches single PI files if any and re-initializes the PIs.

## Output Files

### Naming Convention

Event-<date>-<sequence number>

PI-<date>-<sequence number>

### Location

$ABP_PM_ROOT/work

## Parameters

| Parameter | Description |
|---|---|
| Map Key | Rerate Map Key |
| Partition ID | Rating Partition ID |
| Number in Parallel | Number of Rerate processes to be executed in parallel |

## Databases

- Audit and Control and Product Catalog tables (Oracle) – History
- Rated Event data storage (TimesTen or Oracle) – Usage
- Customer data storage (TimesTen or Oracle) – Customer

## Distribution

NA

## Console Messages

NA

## Troubleshooting

This paragraph contains a list of suggestions that can explain why the job did not run successfully or to prevent it from failing.

NA

## Recovery Instructions


*For details see Rating Recovery Specs Document*

# 22. PM1PREREP - PREPARE AND REPORT

## Description

This is the first job in the rerate map. It prepares entries for the rerate extract processes and sends a report to Billing that includes all customers that are about to be rerated.

### Job Type

This paragraph contains the type of job (extract, read, write, etc.).

**Job Type:** NA

### Run Frequency

This paragraph defines how often the job runs; for example, daily, monthly, by request, etc.

**Run Frequency**: By request

### Activation and Shutdown

#### Job Activation

**Command Line**: pm1PreRerShell_sh 1 2 2002

**AMC**: Yes, as part of the rerate map

**Screen Composer:** No

**Script Name:** pm1PreRer_sh

**Executable:** pm1PrepareAndReport

#### Job Shutdown

**AMC**: Yes

Screen Composer: No

### Log File

#### Name

- Pm.pm1PrepareAndReport_PRERER_<Map key>_<Partition ID>_<Process Instance>.<Date Time>_Error.log
- Pm.pm1PrepareAndReport_PRERER_<Map key>_<Partition ID>_<Process Instance>.<Date Time>_Trace.log

#### Location

$ABP_PM_ROOT/work.

# Job Flow

## Run Level

This paragraph defines the level at which the job runs in the End of Day map (If relevant)

**Run Level:** N/A

## Preceding Jobs

This paragraph contains a list of the jobs that must be successfully performed **before** the job is run.

**Preceding jobs:**

- Rerating extract
- Reguiding

## Dependent Jobs

This paragraph contains a list of the jobs that can only be run after the successful completion of this job.

**Dependent jobs:** None

## Input Files

None

## Flow

1. Gets a cycle list XML file from A&C by the map key (data group)
2. Splits the population by cycle codes and sub-partition, allocates a run ID to each group and copies all groups from the usage database to the history database.
3. Generates an xml output file with all the customers' IDs.
4. Directs the file to the Billing application via A&C.

## Output Files

### Naming Convention

BillingReport_<DateTime>_<partition id>_<cycle code>.txt

### Location

$ABP_PM_ROOT/work

## Parameters

| Parameter | Description |
|---|---|
| Map Key | Rerate Map key |
| Partition ID | Partition ID |
| Run Mode | Customer/Cycle |
| Map Mode | END/MID |

## Databases

Audit and Control and Product Catalog tables (Oracle) – History

## Distribution

NA

## Console Messages

NA

## Troubleshooting

This paragraph contains a list of suggestions that can explain why the job did not run successfully or to prevent it from failing.

NA

## Recovery Instructions

This paragraph contains a list of suggestions how to recover the job.

NA

# 23. PM1POSTRER - REPORT AND FINALIZE

## Description

This is the last job in the rerate map. It reports to Billing about subscribers that still exist in the SUBSCRIBER_RERATE table.

### Job Type

This paragraph contains the type of job (extract, read, write, etc.).

**Job Type:** NA

### Run Frequency

This paragraph defines how often the job runs; for example, daily, monthly, by request, etc.

**Run Frequency**: NA

### Activation and Shutdown

#### Job Activation

**Command Line**: pm1PostRerShell_sh

**AMC**: Yes, as part of rerate map.

**Screen Composer:** No

**Script Name:** pm1PostRer_sh

**Executable:** Pm1ReportAndFinalaize

#### Job Shutdown

**AMC**: Yes

Screen Composer: No

### Log File

#### Name

- Pm.pm1PrepareAndReport_POSTRER_<Map key>_<Partition ID>_<Process Instance>.<Date Time>_Error.log
- Pm.pm1PrepareAndReport_POSTER_<Map key>_<Partition ID>_<Process Instance>.<Date Time>_Trace.log

#### Location

$ABP_PM_ROOT/work.

# Job Flow

## Run Level

This paragraph defines the level at which the job runs in the End of Day map (If relevant)

**Run Level:** N/A

## Preceding Jobs

This paragraph contains a list of the jobs that must be successfully performed **before** the job is run.

**Preceding jobs:** Rerate

## Dependent Jobs

This paragraph contains a list of the jobs that can only be run after the successful completion of this job.

**Dependent jobs:** None

## Input Files

None

## Flow

1. Fetches the cycle list from the history database using the map key.
2. From the usage database, fetches customers for rerate for all the relevant cycles. Sends those customers to Billing using an XML file.

## Output Files

### Naming Convention

BillingFinReport_<date_time>_<partition ID>.log

### Location

$ABP_PM_ROOT/interfaces/output

# Parameters

| Parameter | Description |
|---|---|
| Map Key | Rerate Map key |
| Partition ID | Partition ID |
| Map Mode | END/MID |

# Databases

- Audit and Control and Product Catalog tables (Oracle) – History
- Rated Event data storage (TimesTen or Oracle) – Usage

## Distribution

Billing machine

## Console Messages

NA

## Troubleshooting

This paragraph contains a list of suggestions that can explain why the job did not run successfully or to prevent it from failing.

NA

## Recovery Instructions

This paragraph contains a list of suggestions how to recover the job.

NA

# 24. PM1PIMAIN - PI MAINTENANCE

## Description

This job archives, reconstructs, and purges performance indicators.

The process performs the following subtasks regarding the Performance Indicator table in TimesTen, according to a flag defined in the configuration file:

- Archive – Reads PI information (including static columns and dynamic data as is) from the TimesTen Performance Indicator table into an output file, from which it is loaded to the PI History database and the Oracle Performance Indicator table.
- Purge – Cleans unneeded memory by removing records or setting columns holding dynamic data to "null." Note that the presence of the "reconstruct" verb enables the processing of cross-cycle PIs in memory.
- Reconstruct – Copies the values of cross-cycle PIs from the previous cycle to the new cycle and invokes initialization handlers.
- Re-archive – Deletes the relevant PIs from PI History database and then archives them in the same way as in Archive mode.

This process is subject to the following restrictions:

- Archive and Re-archive mode – A month should not be archived unless the preceding month already has been archived. The Archive mode of the Maintenance process should only be run once on the same cycle . To rerun the archive process on the same cycle, use the Re-archive mode.
- Archive and Purge – These tasks are mutually exclusive. While the cycle month is being archived, the Cycle State archived indicator directs Customer Management queries to TimesTen. Since the archiving process might fail before completion, purging while archiving could result in part of the cycle data being sent to TimesTen and part to Oracle. The two operations therefore are not performed concurrently.
- Reconstruct – A month should not be reconstructed unless the preceding month already has been archived, because reconstruction can delete or overwrite the preceding month's data.
- Purge – A month should not be purged unless that month already has been archived.

### Job Type

This paragraph contains the type of job (extract, read, write, etc.).

**Job Type:** NA

### Run Frequency

This paragraph defines how often the job runs; for example, daily, monthly, by request, etc.

**Run Frequency**: By request

## Activation and Shutdown

### Job Activation

**Command Line**: pm1PIMaintenanceShell_sh true false false false 1 2 2002 1

**AMC**: No

Screen Composer: No

Script Name: pm1PIMaintenance_sh

Executable: pm1PIMaintenance

### Job Shutdown

**AMC**: No

Screen Composer: No

## Log File

### Name

Pm. pm1PIMaintenance.<partition ID>_<cycle instance>_<cycle year>.<date_time>_Error.log

Pm. pm1PIMaintenance.<partition ID>_<cycle instance>_<cycle year>.<date_time>_Trace.log

### Location

$ABP_PM_ROOT/work.

# Job Flow

## Run Level

This paragraph defines the level at which the job runs in the End of Day map (If relevant)

**Run Level:** N/A

## Preceding Jobs

None

## Dependent Jobs

None

## Input Files

None

## Flow

The process does the following:

1. Performs initialization
2. Reads PI maintenance configuration data and performs validity checks according to the process restrictions
3. Initializes the Pricing Engine
4. Prepares three contexts:
   - CurrentEventPI
   - HistoryEventPI
   - Customer
5. Prepares a "spare" context for CurrentEventPI for purge, reconstruct, and commit purposes
6. Creates query parameters (PI, cycle code, cycle month, cycle year, etc.).
7. Uses the Query API to retrieve the PI records for the given parameters
8. Opens a cursor
9. While fetching the PI entity:
   - Writes the entity to a file. On reaching a predefined Number of Records per File, changes the file name so that the Replicator reads the file.
   - Calls the Rater API to purge or reconstruct (with the "spare" context)
   - If the predefined Number of Transactions to Commit is reached, closes the file and renames it for the Replicator to read and commit (with the "spare" context) to the database
10. Closes the cursor

## Output Files

### Naming Convention

None

### Location

None

# Parameters

The batch job uses the following parameters.

| Parameter | Description |
|---|---|
| Cycle code | The code of the cycle |
| Cycle instance | The instance of the cycle |
| Cycle year | The year of the cycle |
| Partition ID | Rating partition ID |
| Archived | True or false |
| Reconstruct | True or false |
| Purge | True or false |
| Rearchived | True or false |

# Databases

- Audit and Control and Product Catalog tables (Oracle) – History
- Rated Event data storage (TimesTen or Oracle) – Usage

# Distribution

NA

# Console Messages

NA

# Troubleshooting

This paragraph contains a list of suggestions that can explain why the job did not run successfully or to prevent it from failing.

NA

# Recovery Instructions

This paragraph contains a list of suggestions how to recover the job.

NA

# 25. UPDATE CYCLE STATE JOBS

## Description

The Update Cycle State process performs several actions to update the Cycle State table.

The process consists of 3 separate jobs:

- PM1CYCLESTATE
- PM1LOCKBILL
- PM1CYCREOPEN

### Job Type

This paragraph contains the type of job (extract, read, write, etc.).

**Job Type:** NA

### Run Frequency

This paragraph defines how often the job runs; for example, daily, monthly, by request, etc.

**Run Frequency**: By request

### Activation and Shutdown

#### Job Activation

**Command Line**:

- pm1UpdateCycleStateShell_sh 1 1 LOCK_RERATE
- pm11OpenCycleShell_sh  1 1 REOPEN_NEW

**AMC**: Yes, as part of rerate map

**Screen Composer:** No

**Script Name:** pm1UpdateCycleState_sh/ pm1OpenCycle_sh

**Executable:** pm1CycleEnvelop

#### Job Shutdown

**AMC**: Yes

Screen Composer: No

### Log File

#### Name

PM1CYCLESTATE / PM1LOCKBILL - Pm. pm1CycleEnvelop.UpdateCycle_<Map Key>_<PartitionID>_<ProcessInstance>_<DateTime>.log

PM1CYCREOPEN - Pm. pm1CycleEnvelop.OpenCycle_<Partiton ID>_>_<ProcessInstance>_<DateTime>.log

#### Location

$ABP_PM_ROOT/work.

## Job Flow

### Run Level

This paragraph defines the level at which the job runs in the End of Day map (If relevant)

**Run Level:** N/A

### Preceding Jobs

None

### Dependent Jobs

None

### Input Files

None

### Flow

The process does the following:

1. Connects to the Oracle database
2. Brings the map key, partition ID, run mode from the configuration file (either LOCK_RERATE, LOCK_BILL, REOPEN_NEW )
3. Updates the Cycle State table according to all cycles that are handled in the appropriate rerate map

### Output Files

None

## Parameters

| Parameter | Description |
|---|---|
| Map Key | Rerate Map Key |
| Partition ID | Rating Partition ID |
| Run mode | LOCK_RERATE/LOCK_BILL/REOPEN_NEW |

## Databases

Audit and Control and Product Catalog tables (Oracle) – History

## Distribution

NA

## Console Messages

NA

## Troubleshooting

This paragraph contains a list of suggestions that can explain why the job did not run successfully or to prevent it from failing.

NA

## Recovery Instructions

This paragraph contains a list of suggestions how to recover the job.

NA

# 26. PM1TRUNCCYCLE - TRUNCATE CYCLE

## Description

This process gets the partition ID, cycle code, cycle month, and cycle year as parameters. It truncates the relevant partitions from the RATED_EVENT, RPR1_SUBS_RERATE and PERFORMANCE_IND tables.

### Job Type

This paragraph contains the type of job (extract, read, write, etc.).

**Job Type:** NA

### Run Frequency

This paragraph defines how often the job runs; for example, daily, monthly, by request, etc.

**Run Frequency**: By request

### Activation and Shutdown

#### Job Activation

**Command Line**: pm1TruncateCycleShell_sh 1 1 2 2002

**AMC**: No

Screen Composer: Yes

Script Name: pm1TruncateCycle_sh

Executable: pm1CycleEnvelop

#### Job Shutdown

**AMC**: No

Screen Composer: Yes

### Log File

#### Name

Pm. pm1CycleEnvelop.UpdateCycle_<Map Key>_<PartitionID>_<ProcessInstance>_<DateTime>.log

#### Location

$ABP_PM_ROOT/work.

# Job Flow

## Run Level

This paragraph defines the level at which the job runs in the End of Day map (If relevant)

**Run Level:** N/A

## Preceding Jobs

None

## Dependent Jobs

None

## Input Files

None

## Flow

1. Connects to the Oracle database
2. Checks the Maintenance Indicator field in the PM1_CYCLE_STATE table and proceeds only if the Maintenance Indicator is set to “B” (backed up)
3. Truncates the Rated Event, Rejected Event, and Performance Indicator tables according to the cycle code, cycle month, and cycle year

## Output Files

None

# Parameters

| Parameter | Description |
|---|---|
| Cycle code | The code of the cycle |
| Cycle month | The month of the cycle |
| Cycle year | The year of the cycle |
| Partition ID | Rating partition ID |

# Databases

This job requires access to the Rated Events database (Oracle).

# Distribution

NA

# Console Messages

NA

## Troubleshooting

This paragraph contains a list of suggestions that can explain why the job did not run successfully or to prevent it from failing.

NA

## Recovery Instructions

This paragraph contains a list of suggestions how to recover the job.

NA

# 27. PM1DISPATCHER - DISPATCHER

## Description

This process dispatches events to specified destinations, such as the Rated Events table or output files.

### Job Type

This paragraph contains the type of job (extract, read, write, etc.).

**Job Type:** Daemon

### Run Frequency

This paragraph defines how often the job runs; for example, daily, monthly, by request, etc.

**Run Frequency**: Runs constantly

### Activation and Shutdown

#### Job Activation

**Command Line**: pm1DispatcherShell_sh DSN1 0

**AMC**: Yes

Screen Composer: Yes

Script Name: pm1Dispatcher_sh

Executable: pm1Dispatcher

#### Job Shutdown

**AMC**: Yes

Screen Composer: Yes

### Log File

#### Name

- pm.pm1Dispatcher_<DSN>_<Dispatcher ID>_<process Instance>.<date time>_Error.log
- pm.pm1Dispatcher_<DSN>_<Dispatcher ID>_<process Instance>.<date time>_Trace.log

#### Location

# $ABP_PM_ROOT/work. Job Flow

## Run Level

This paragraph defines the level at which the job runs in the End of Day map (If relevant)

**Run Level:** N/A

## Preceding Jobs

This paragraph contains a list of the jobs that must be successfully performed **before** the job is run.

**Preceding jobs**: PM1XLA2FILE or Postpaid Rater

## Dependent Jobs

This paragraph contains a list of the jobs that can only be run after the successful completion of this job.

**Dependent jobs:** None

## Input Files

### Naming Convention

RD_<unique ID>_dispatching_file_<date>.dat

### Location

$ABP_PM_ROOT/data

## Flow

The process does the following:

1. Connects to Audit & Control
2. Gets the input file
3. Reads the input record
4. Formats the record according to the destination's requirements, defined in RPR1_DISP_TARGET
5. Distributes the record according to the target ID on the record

## Output Files

### Naming Convention

<prefix>_time

The prefix is taken from the rpr1_disp_target table.

### Location

rpr1_disp_target.file_path

## Parameters

| Parameter | Description |
|---|---|
| DSN | Data store name, run time |
| Dispatcher ID | ID of the Dispatcher run time |

## Databases

- Audit and Control and Product Catalog tables (Oracle) – History
- Rated Event data storage (TimesTen or Oracle) – Usage

## Distribution

NA

## Console Messages

NA

## Troubleshooting

This paragraph contains a list of suggestions that can explain why the job did not run successfully or to prevent it from failing.

NA

## Recovery Instructions

This paragraph contains a list of suggestions how to recover the job.

NA

# 28. PM1STOPDISP - STOP DISPATCHER

## Description

This process stops the Dispatcher process.

### Job Type

This paragraph contains the type of job (extract, read, write, etc.).

**Job Type:** NA

### Run Frequency

This paragraph defines how often the job runs; for example, daily, monthly, by request, etc.

**Run Frequency**: By request

### Activation and Shutdown

#### Job Activation

**Command Line:** `pm1StopDispatcher_sh DSN1 0`

**AMC**: Yes

Screen Composer: Yes

Script Name: pm1StopDispatcher

Executable: None

#### Job Shutdown

**AMC**: No

Screen Composer: No

### Log File

None

## Job Flow

### Run Level

This paragraph defines the level at which the job runs in the End of Day map (If relevant)

**Run Level:** NA

### Preceding Processes

None

### Dependent Processes

None

### Input Files

None

### Flow

- Finds the Dispatcher PID (Process ID) that runs by the specified partition code and sub-partition code.
- Stops the Dispatcher process by sending a SIGTERM signal to the server. The server receives the signal and shuts down.

### Output Files

None.

## Parameters

| Parameter | Description |
|---|---|
| DSN | Partition code |
| Dispatcher ID | Unique ID to differentiate between dispatchers working on the same partition |

## Databases

This paragraph specifies the databases the job uses, for example, *Usage1*, *Usage2*, *Customer*, etc.

NA

## Distribution

NA

## Console Messages

NA

## Troubleshooting

NA

## Recovery Instructions

This paragraph contains a list of suggestions of how to recover the job.

NA

# 29. PM1XLA2FILE - XLA TO FILE

## Description

This job is a daemon process responsible for creating files containing the records stored in TimesTen. This process selects a portion of the records in the Rated Event and Rejected Event usage tables from the TimesTen logs, removes the Rater engine task ID from the records, and inserts them into a file.

### Job Type

This paragraph contains the type of job (extract, read, write, etc.).

**Job Type:** Daemon

### Run Frequency

This paragraph defines how often the job runs; for example, daily, monthly, by request, etc.

**Run Frequency**: Runs constantly

### Activation and Shutdown

#### Job Activation

**Command Line**: pm1XLAReplicatorShell_sh DSN1

**AMC**: Yes

Screen Composer: Yes

Script Name: pm1XLAReplicator_sh

Executable: xladaemon

#### Job Shutdown

**AMC**: Yes

Screen Composer: Yes

### Log File

#### Name

- event_file_XLA_<date>.log
- console_file_XLA_<date>.log – prepaid daemon loading log
- error_dispatch_file_XLA_<date>.dat – application errors
- pm1XLA2File_<process ID>.log

#### Location

$ABP_PM_ROOT/work

# Job Flow

## Run Level

This paragraph defines the level at which the job runs in the End of Day map (If relevant)

**Run Level:** N/A

## Preceding Jobs

None

## Dependent Jobs

None

## Input Files

None

## Flow

This process does the following:

- Connects to TimesTen
- Reads the records from the XLA log (the TimesTen log)
- Writes the records into files

## Output Files

### Naming Convention

RD_<unique ID>_dispatching_file_<date>.dat

### Location

$ABP_PM_ROOT/data

# Parameters

| Parameter | Description |
|---|---|
| DSN | Data store name, run time |

# Databases

This job requires access to TimesTen.

# Distribution

NA

# Console Messages

NA

## Troubleshooting

This paragraph contains a list of suggestions that can explain why the job did not run successfully or to prevent it from failing.

NA

## Recovery Instructions

This paragraph contains a list of suggestions how to recover the job.

NA

# 30. PM1SPXLA2FILE – STOP XLA

## Description

This job stops the XLA2File daemon.

### Job Type

NA

### Run Frequency

By request

### Participating in Generic Maps

N/A

### Run Level

N/A

### Activation and Shutdown

#### Job Activation

**Command Line**:

RunJobs PM1SPXLA2FILE <JOB_REQ>

**AMC**: Yes

**Screen Composer:** Yes

**Script Name:** pm1StopXLAReplicator_sh

**Executable:** None

#### Job Shutdown

**AMC**: NA

**Screen Composer:** NA

### Log File

None

## Job Flow

### Run Level

N/A

### Preceding Jobs

XLA2File must be active

### Dependent Jobs

None

### Input Files

None

### Flow

The process sends a signal to the server that stops it gracefully.

### Output Files

None

## Parameters

| Parameter | Description |
|---|---|
| DSN | Data store name, run time |

## Connection Details

NA

## Distribution

NA

## Console Messages

NA

## Troubleshooting

NA

## Recovery Instructions

NA

# 31. PM1MARKNEXT - MARK NEXT FOR RERATE

## Description

The Mark Next for Rerate process runs as part of the end-of-cycle rerate map. It ensures that rerate subscribers whose performance indicators (PIs) were constructed incorrectly for the next cycle are marked.

### Job Type

This paragraph contains the type of job (extract, read, write, etc.).

**Job Type:** NA

### Run Frequency

This paragraph defines how often the job runs; for example, daily, monthly, by request, etc.

**Run Frequency**: By request

### Activation and Shutdown

#### Job Activation

**Command Line**: pm1MarkNextShell_sh 1 1

**AMC**: Yes, as part of the Rerate map

**Screen Composer:** No

**Script Name:** pm1MarkNext_sh

**Executable:** pm1MarkNext

#### Job Shutdown

**AMC**: Yes

Screen Composer: No

### Log File

#### Name

- pm.pm1MarkNext_<DSN>_<Partition ID>.< date time>_Error.log
- pm.pm1MarkNext_<DSN>_<Partition ID>.< date time>_Trace.log

#### Location

$ABP_PM_ROOT/work.

# Job Flow

## Run Level

This paragraph defines the level at which the job runs in the End of Day map (If relevant)

**Run Level:** N/A

## Preceding Jobs

This paragraph contains a list of the jobs that must be successfully performed **before** the job is run.

**Preceding jobs:** Rerate

## Dependent Jobs

This paragraph contains a list of the jobs that can only be run after the successful completion of this job.

**Dependent jobs:** None

## Input Files

None

## Flow

1. Connects to TimesTen
2. Reads the records from PI table
3. Checks if the PI was constructed incorrectly
4. Marks the records of subscribers whose PIs are incorrect

## Output Files

None

# Parameters

| Parameter | Description |
|---|---|
| Map Key | Rerate Map Key |
| Partition ID | Rating partition ID |

# Databases

- Audit and Control and Product Catalog tables (Oracle) – History
- Rated Event data storage (TimesTen or Oracle) – Usage

# Distribution

NA

## Console Messages

NA

## Troubleshooting

This paragraph contains a list of suggestions that can explain why the job did not run successfully or to prevent it from failing.

NA

## Recovery Instructions

This paragraph contains a list of suggestions how to recover the job.

NA

# 32. PM1BCKEVNT - BACKUP EVENT DB

## Description

The Event tables are very large, so it is impossible to store all their data in the production database. Therefore, there is a need to reuse the space of these tables. All four tables are created using the Oracle partition mechanism, which enables different cycles and months to reside on different partitions. (The Rated Event table also uses a sub-partition as a part of the partition key, so a cycle and month can reside on multiple partitions.)

The Backup Event DB job backs up a database partition, so it can later be truncated, and its space reused for a different cycle and month.

### Job Type

This paragraph contains the type of job (extract, read, write, etc.).

**Job Type:** backup

### Run Frequency

This paragraph defines how often the job runs; for example, daily, monthly, by request, etc.

**Run Frequency**: By request

### Activation and Shutdown

#### Job Activation

**Command Line**: pm1bckevnt_sh 1 2 2003 amtwrkl amtwrkl. EXP ammh9i 1 or (interactive script)

pm1bckevnt_pid _sh

AMTI9i

**AMC**: No

Screen Composer: Yes

Script Name: pm1bckevnt_pid_sh

pm1bckevnt_opr_sh (runs from operational)

Executable: None

#### Job Shutdown

**AMC**: No

Screen Composer: Yes

### Log File

#### Name

The log file name is M_<date>.log.

#### Location

$ABP_PM_ROOT/work.

# Job Flow

## Run Level

This paragraph defines the level at which the job runs in the End of Day map (If relevant)

**Run Level:** NA

## Preceding Jobs

None

## Dependent Jobs

None

## Input Files

None

## Flow

The process does the following:

1. Connects to the database.
2. Performs a backup.

## Output Files

None

# Parameters

The batch job uses the following parameters.

| Parameter | Description |
|---|---|
| Table owner | The user name of the owner of the RATED_EVENT, PERFORMANCE_IND, REJECTED_EVENT, and RPR1_SUBS_RERATE tables<br>Given manually when running shell script or by GN1_TASK_CONNECT for operational run |
| Table owner password | Password of the table owner<br>Given manually when running shell script or by GN1_TASK_CONNECT for operational run |
| Instance | The name of the instance<br>Given manually when running shell script or by GN1_TASK_CONNECT for operational run |
| PARTITION_ID | Partition to backup |
| CYCLE_CODE | The code of the cycle to backup |
| CYCLE_MONTH | The month of the cycle to backup |
| CYCLE_YEAR | The year of the cycle to backup |
| ABP_PM_WORK | Directory for backup |
| BCK_TYPE | EXP for Export and FUO for Fast Unload |
| ALL_PARTITIONS | Y or N |

# Databases

This job requires access to TimesTen.

# Distribution

NA

# Console Messages

NA

# Troubleshooting

This paragraph contains a list of suggestions that can explain why the job did not run successfully or to prevent it from failing.

NA

# Recovery Instructions

This paragraph contains a list of suggestions how to recover the job.

NA

# Document Release Information

| Case No. | Service Pack No. | Description of Change |
|---|---|---|
| 300073-032606 | SP7 | Rearchive mode was added to the PI Maintenance process in chapter 24. |