

amdocsrating

Rating 6.0

Detailed Interfaces



© 2005 Amdocs

This document contains proprietary and confidential information of Amdocs and shall not be reproduced or transferred to other documents, disclosed to others, or used for any purpose other than that for which it is furnished, without the prior written consent of Amdocs. It shall be returned to the respective Amdocs companies upon request.

The trademark and service marks of Amdocs, including the Amdocs mark and logo, are the exclusive property of Amdocs, and may not be used without permission. All other marks mentioned in this material are the property of their respective owners.

Document Information

Software Version:	6.0
Publication Date:	January 2005; Updated November 2005 for SP5
Catalog Number:	234528

Contents

Contents	iii
Rating 6.0 Input Interfaces	1
1. Introduction	3
Purpose and Scope.....	3
Terminology	3
2. Customer Information to Rating Area.....	5
Overview	5
Customer Extract and Load Results.....	7
3. Customer Extract	11
Process Flow Overview.....	11
Extract Process Modes	11
Detailed Process Flow.....	14
Customer Tables Used.....	15
Distribute Data	15
Write to Files	16
Customer Extract Configuration File	21
Transaction Broker Transaction Configuration File	26
Parameter Definition in Product Catalog	28
4. Customer Load	31
Process Flow.....	31
Pre- and Post-Activity Events.....	32
Generic and Partition File Support	32
Run Modes.....	33
5. Product Catalog Information	35
Product Catalog Subsystems	35
Implementation Repository.....	35
Pricing Package Catalog and Offer Catalog.....	35
Auxiliary Repository.....	36
Product Catalog Data Distribution	36
6. Events.....	37
Network Usage Events.....	37
Non-Usage Events	39
Event Structure	40
Defining New Events	40

Event Rating Core Attributes.....	40
Rating 6.0 Output Interfaces	43
1. Introduction.....	45
Purpose and Scope.....	45
2. Product Catalog Definitions.....	47
External Record Layout Type.....	47
Mapping of Rating Entities to External Records.....	48
3. Event and Performance Indicator Extracts.....	59
Extract Process Overview	59
Process Input Parameters.....	60
Process Flow.....	62
4. Online Usage Queries	67
Online Query Architecture	67
Rating Online Query APIs	68
Rating Reference Tables for Front End	72
5. Rated Event Dispatching Mechanism	75
Product Catalog Definitions.....	76
Dispatching	78
Distributor.....	79
Dispatching Example	80
6. Notification Mechanism	81
Purpose of Notification Mechanism.....	81
Product Catalog Definitions.....	81
Notification Trigger	82
Rating Notification Mechanism.....	82
Distributor.....	82
Appendixes	83
Appendix A. Native XML.....	85
EntityData_t Type.....	85
Input File Example	86
Output File Example.....	87
Index	89

amdocrating

Rating 6.0 Input Interfaces

1. INTRODUCTION

This book describes the information that Amdocs Rating requires to rate an event. This information includes:

- Customer information – The offers acquired by the customer, specific customer parameters, specific customer offer parameters, and cycle change information.
- Product Catalog information – Pricing, service, and discount packages; the Implementation Repository, in which the event types, simple and complex data types, and pricing logic are defined; and the Auxiliary Repository, which defines certain reference data needed for pricing.
- Event information – Events are passed to Rating after being processed by Acquisition & Formatting, which transforms the raw event data records to the Rating-specific event type format and populates Billing customer information for the event.

Purpose and Scope

This part of this book describes Amdocs Rating inputs as follows:

- Chapter 1, “Introduction” (this chapter), briefly introduces this part of the book’s subject and its structure, and defines special terms.
- Chapter 2, “Customer Information to Rating Area,” describes the customer data storage function in Rating.
- Chapter 3, “Customer Extract,” explains the process that extracts customer data from the Customer Management database for use by Rating.
- Chapter 4, “Customer Load,” explains the process that loads customer data into Rating data storage, and sends events for Pricing Engine processing.
- Chapter 5, “Product Catalog Information,” describes the types of Product Catalog information that Rating uses.
- Chapter 6, “Events,” describes the types of events that Rating handles and how they are processed.

Terminology

The following specialized terms are used in this book:

Term	Definition
A&C	<i>Audit & Control.</i> The Amdocs counting application responsible for keeping track of EDR files within the Amdocs Billing platform.

Rating 6.0 Input Interfaces

Term	Definition
A&F	<i>Acquisition & Formatting.</i> A product in the Amdocs Billing Platform responsible for acquiring and formatting, EDRs and afterward, guiding them to the appropriate customer.
AMC	<i>Application Monitoring and Control.</i> The Amdocs application responsible for monitoring the Amdocs Billing platform products.
ART	<i>Amdocs Real Time</i> infrastructure tool.
CET	<i>Charge Entity Type</i>
CSP	<i>Communications service provider</i>
CSR	<i>Customer service representative</i>
DWH	<i>Data warehouse</i>
EDR	<i>Event detail record</i>
IMDB	<i>In-memory database</i>
Implementation Repository	The Product Catalog data dictionary. This include structural definitions for business objects together with business rules defining all possible Pricing Item types
Operational	Operational is a table-driven Amdocs tool with a GUI front-end to make batch job scheduling easier, and to provide a means for real-time monitoring of the running of batch jobs and their dependencies.
PI	<i>Performance Indicator</i> – Contains information on all accumulated rated events for a subscriber or group of subscribers. Any attribute and amount of the event can be accumulated. Usually the chargeable attributes as well as the event rates are accumulated. PIs are used for retaining data in connection with allowances, discounts, and budget control schemes.
PIT	<i>Pricing item type.</i> Defines a template for the functionality of a single, independent pricing element.
Pricing Engine	The core virtual engine in Rating that contains the rating calculations for customers and subscribers.
RMI	<i>Remote Interface</i>
TimesTen	TimesTen is a high-performance relational IMDB that supports the ODBC (Open Database Connectivity) and JDBC (Java Database Connectivity) interfaces. The product is supported on multiple platforms. It allows transaction management, full or partial replication, and also makes its logged data accessible through an open API.

2. CUSTOMER INFORMATION TO RATING AREA

Customer information is extracted from the Customer Management database and loaded to the Rating customer data storage area by two processes: Customer Extract and Customer Load.

In addition, Customer Extract, which captures Customer Management activities, generates subscriber activity events for relevant activities. Rating uses these events to mark the subscriber for rerating, and sometimes for other purposes as well.

This chapter provides a general overview of the Customer Extract and Customer Load processes and their results. Both processes are described in greater detail in the chapters that follow.

Overview

To rate an event, the Pricing Engine uses customer information retrieved from the customer database. The input information comes from two sources:

- Acquisition & Formatting (A&F) – The Guiding to Customer function populates some customer information on the input event, including billing information such as the customer ID, cycle code, account, and pay channel.
- Customer Extract – This Rating information is extracted from the customer database:
 - Offers purchased by each subscriber, used by the Rating in the Guiding to Service stage.
 - Subscriber or customer parameters, used by the Pricing Engine for qualifying Pricing Items and in the computation models of pricing item types (PITs). For example, the Friends and Family (F&F) qualification criterion checks whether the event's dialed number is on the F&F list under the subscriber parameters.
 - Subscriber offer parameters, used by the Pricing Engine for qualifying pricing and in the computation models of PITs. For example, the Mailbox Size attribute is used in the computation model of different PITs where the Mailbox Size value can be different for each PIT.
 - Cycle change, used by the Pricing Engine to control allowance proration and PI rolling and maintenance (for example, a customer's allowance is prorated in the cycle to which he has moved).

Loading customer information into the Rating data storage area complies with the basic objectives of Rating, ensuring that:

- Rating is decoupled from other components, so that the other components can be third-party systems.
- Only required data is stored, minimizing the data storage required by the Rating server.
- Rating's performance is enhanced, because:
 - All data are accessed locally, eliminating redundant network data transmissions.
 - The data are stored in fewer tables, reducing the number of tables Rating must access.
 - Only relevant information is extracted from Customer Management to Rating.
 - Data can be stored in an in-memory database, while Customer Management tables are in an Oracle database. (The Rating information can be stored in Oracle tables as well.)

Customer data are brought into Rating in two stages:

- Extract information from the customer management data area. When Amdocs Customer Management is the customer management application, the Customer Extract process described in this chapter is used. If another customer management application is used, it must supply the information in an XML file in the same structure.
- Load extracted information into the Rating storage area, whether in a database on disk or in memory.

The figure below describes the flow of customer information from Customer Management data storage to the Rating customer data storage.

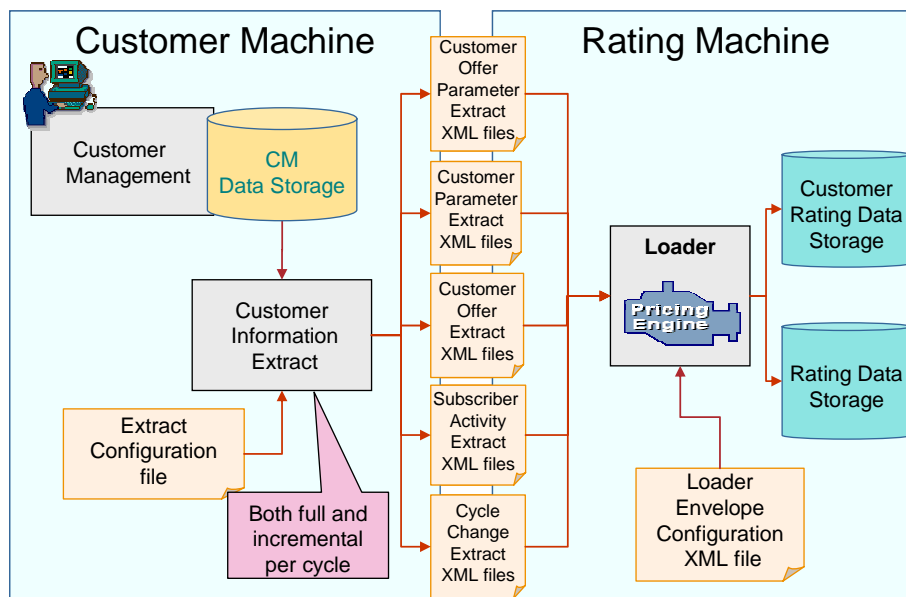


Figure 2-1: Flow of Customer Information

Customer Extract and Load Results

The results of the Extract and the Load processes are as follows:

- Rating customer data are loaded into TimesTen (in-memory database) tables or Oracle tables (described below).
- Subscriber activity events are created that are used for marking subscribers for rerating.

Rating Customer Tables

Rating has four customer data tables:

- Customer Offers – Offers for each subscriber
- Customer Parameters – Parameters for each customer or subscriber
- Customer Offer Parameters – Parameters for each offer of a customer or subscriber
- Change Cycle History – Cycle change history for each customer that changed cycles.

These tables are described in the subsections below.

Customer Offers Table

The Customer Offers table includes this information:

- Subscriber ID
- Dynamic data – a list of offers stored in a binary object, including:
 - Offers
 - Effective date of each offer.
 - Expiration date of each offer.
 - Offer instance – A unique identifier of the offer instance in the subscriber. For example, when two identical allowance offers are attached to the subscriber, Customer Management generates a different offer instance for each occurrence.
 - Offer agreement ID – When the agreement is at the subscriber level, the offer agreement ID is the subscriber ID. Otherwise, the offer ID equals the ID of the agreement to which the offer is attached.
 - Offer agreement instance – A unique identifier of the offer at the agreement level.
 - Effective activity code – Represents the activity code for the move to the new customer, when a subscriber moves from one customer to another.
 - Expiration activity code – Represents the activity code for the move from the new customer, when a subscriber moves from one customer to another.

Customer Parameters Table

The Customer Parameters table includes this information:

- Subscriber ID
- Effective date
- Expiration date
- Dynamic data – Subscriber parameters, including:
 - Parameter name
 - Parameter value

The following *core parameters* are essential for Rating and should always be included in the configuration file:

- Cycle code
- Customer ID
- Customer offer currency
- Customer business entity
- Change cycle indicator

Customer Offer Parameters Table

The Customer Offer Parameters table includes this information:

- Offer-level subscriber ID
- Offer instance
- Offer-level effective date
- Offer-level expiration date
- Dynamic data – Subscriber parameters, including:
 - Parameter name
 - Parameter value

Change Cycle History Table

The Cycle Change table includes this information:

- Customer ID
- Effective date
- Old cycle code
- New cycle code
- Status

Subscriber Activity Event

In incremental mode, the Customer Extract process runs as a daemon in the background of Rating. It checks the Transaction Broker for Customer Management activities such as Add Offer and Cancel Offer. For each relevant activity, the extract process generates a subscriber activity event.

The Customer Load process passes the event to the Pricing Engine, which processes it like all other EDRs.

For these activities to be processed by Rating, an appropriate implementation needs to be defined in the Product Catalog.

The basic implementation is used to mark a subscriber for rerating. The implementation is defined at the market level and in the basic allowance and rate PITs. It implements the behavior required to react to each subscriber activity.

The event can be used for other purposes, such as creating allowance PIs for new subscribers or new offers, so that the allowance budget will be available for online queries.

3. CUSTOMER EXTRACT

This chapter describes the general flow of the Customer Extract process.

Process Flow Overview

These are the stages of the Customer Extract:

- Extract information from customer tables according to the Extract Configuration XML file. The extract is done on the customer server. Queries of the customer database can be performed simultaneously using the multithreaded mechanism embedded into the extract process.
- Write to XML files using the Pricing Engine's generic XML structure.

These stages are performed concurrently, but for purposes of clarity are described separately in the sections that follow.

Extract Process Modes

The process can run in one of the modes described here.

Configuration Mode

The Configuration mode is a full extract mode. When configuration mode is selected, one of the submodes described below must be selected.

Full Submode

The Full submode initializes an empty Rating customer database. Information for all the customers in the entire Customer Management database is extracted and loaded into the Rating customer tables.

FullFill Submode

The FullFill submode updates the Rating customer database in full. Information for all the customers in the entire Customer Management database is extracted and updated into the Rating customer tables.

Provider Submode

Outcollect calls (roaming calls by visitors) are rated according to the Provider ID of the caller. Provider mode extracts information from the Customer Management Provider Agreement table, which associates outcollect providers with the offers relating to them, and loads it into the Rating customer area. The table contains information similar to the information extracted from the Agreement ID, Offer, and Parameters tables.

The Provider mode extract runs once at the beginning of the system's life cycle, and thereafter by request. Each time the process runs, information is updated in the Provider Agreement table.

ManualMark Submode

The ManualMark submode generates subscriber activity events for a requested population. These events are used to mark subscribers for rerating. The process selects the population according to a SQL statement written in a manual mapping definition file.

For example, all subscribers who were connected recently are entitled to an allowance that was not given to them. The allowance is attached to the subscribers afterward. These subscribers should be rerated so as to apply the allowance to previously rated events. The select statement of the manual mode should select all new subscribers who were connected recently.

Incremental Mode

Incremental mode updates the Rating customer database with changes to the Customer Management database and Billing new cycle instances. In addition, it captures transactions that may cause rerating, and generates a subscriber activity events for these transactions that are used to mark the subscriber for rerating.

The Transaction Broker is responsible for transferring transactions between components in the system. Customer Management uses the Transaction Broker to publish information on activations, cancellations, service updates, bill cycle changes, and so on.

The Customer Extract process reads the last published Customer Management and Billing transactions from the Transaction Broker and extracts the relevant information from the Customer tables and Transaction Broker XML transactions into files.

In the incremental mode, the old information of the updated subscriber is deleted and the updated information is inserted in its place. To maintain the history, the extract information is for a defined time interval (all the effective records plus a history of n months). Rating sees the new updates after they are committed.

Only a portion of the Customer Management activities influence the change of Rating customer data or are triggers for rerating. The Customer Extract process subscribes only to the relevant activities. The subscription is defined in the Transaction Broker Application Definition reference table.

These are the activities that are monitored by the core implementation:

- ADD_GROUP_MEMBER – Add closed user group (CUG) member
- REMOVE_GROUP_MEMBER – Remove CUG member
- CREATE_AGREEMENT – Create agreement
- AGR_SERVICE_CHANGE – Service change at the agreement level
- AGR_CHANGE_PRICE_PLAN – Price plan change at the agreement level
- CANSUB – Cancel subscriber
- CHANGE_PP – Change price plan
- CHGCYC – Change cycle

- CHGEVENTDIST – Change event distribution
- MIGTOPOST – Migrate to postpaid
- MIGTOPRE – Migrate to prepaid
- MOVE_SUB – Move subscriber
- NEW_SUB_ACTIVATION – Subscriber activation
- SL_NEW_SUB_ACTIVATION – Sale event subscriber activation
- OD_RESTORE_SERVICE – Operator-determined restore service
- OD_RESTORE_SUBSCRIBER – Operator-determined restore subscriber
- OD_SUSPEND_SERVICE – Operator-determined suspend service
- OD_SUSPEND_SUBSCRIBER – Operator-determined suspend subscriber
- RESTORE_SERVICE – Restore service
- RESTORE_SUB – Restore subscriber
- SUSPEND_SERVICE – Suspend service
- SUSSUB – Suspend subscriber
- SERVICE_CHANGE – Service change
- UPDATE_PARAMETERS – Update parameters
- NEW_CYCLE_INSTANCE – New cycle instance
- CANCHGCYC – Cancel cycle change
- RESUME_SUB – Resume subscriber

The Incremental mode can be executed with one of these submodes:

Daemon Submode

In the Daemon submode, the customer extract runs as a daemon and reads Transaction Broker transactions from the TRB1_SUB_LOG table.

History Submode

In the History submode, the customer extract runs as a by-request job that reads transactions from Transaction Broker history table TRB1_MST_LOG, according to a range.

Detailed Process Flow

The Customer Information extract is performed on the customer server.

The extract process is based on the Amdocs Real Time (ART) technology, which enables multithreaded processing of data. The actual queries of the Customer Management database are performed using the Amdocs Generic Extract utility.

This is the general flow of the extract process:

1. The Acceptor thread reads an array of transactions and sends the transactions to the service objects queue. Then it waits for acknowledgment that the processing was completed.
 - For an incremental extract, the acceptor is the TRB Acceptor type and the transactions are Transaction Broker transactions.
 - For a full, provider, or manual extract, the acceptor is the Full Acceptor type. The transactions in this case are fixed, as defined in the extract configuration XML file (for example, extract offers of cycle 1, extract offers of cycle 2, extract parameters of cycle 1, etc.).
2. Service objects are responsible for processing the transactions. Each service object can run multiple threads.

Each service object thread gets the next transaction from the queue and converts it to Transaction Info, which is the internal form of transaction used in the process. The Transaction Info comprises a list of transaction elements, each containing rules for extracting data from Customer Management and formatting it.
3. The generic extract utility performs the data extract. Each thread uses its own instance of the generic extract utility, with its own database connection.
4. The extracted information is then passed to the Formatter, which builds the output record in XML format and sends it to the File Manager.
 - In full extract mode, the File Manager flushes the data to physical files, and updates Audit & Control whenever the number of processed records reaches a specified threshold. The same applies to the provider and manual modes.
 - In incremental extract mode, the File Manager accumulates the output records in memory, in the same sequence they arrived from the Transaction Broker.
5. The Acceptor thread waits for all threads to finish and then signals the File Manager to flush the remaining data it holds, update Audit & Control, and commit the changes to the database.
6. In incremental mode, the Acceptor then reads the next group of transactions. In full mode, the job ends.

Customer Tables Used

These Customer Management tables are used in the typical customer extract implementation:

- *Service Agreement Table* – This table lists the offers that are attached to the subscriber or agreement. This information is extracted to the Rating Subscriber Offers table. When an offer is attached to an agreement that is not a subscriber, Customer Management deploys the offer to all the subscribers under the agreement. Rating does not consider the agreement hierarchy.
- *Agreement Parameters Table* – This table contains the dynamic subscriber parameters. This information is extracted to the Customer Parameters table. For an agreement-level offer, Customer Management deploys the dynamic parameters attached to the agreement to all subscribers under the agreement. The parameters can be overridden on the subscriber level.

For example, when an F&F pricing item is attached to the agreement level, the F&F numbers are populated in the agreement and deployed to all the subscribers. The customer service representative (CSR) may change the parameters at the subscriber level.

- *Subscriber Table* – This table contains general information about the subscriber. Some of the table columns, such as Currency or Business Entity, are extracted and populated as fields or parameters in the Rating Customer Parameters table.
- *Customer Table* – The cycle code is retrieved from this table in the distribution stage.
- *CUG Table* – This table maintains information on Closed User Groups. All its data are extracted to the Rating CUG table.

Distribute Data

The cycle period is the basic unit of information containing all rating data. It is stored as one unit, either billed or unbilled, in a data storage device. Due to the large volume of data, cycles can be maintained in different data storages on different hosts (for example, different remote UNIX servers). Each server contains information about the relevant population of customers, according to a specific distribution principle (the Rating partition). The population of a single cycle can be distributed across multiple Rating partitions.

As a result, separate XML files are created on the Customer server for each Rating partition and file type (subscriber offer, subscriber parameters, subscriber activity, generic and partition).

Write to Files

The extracted information is written to files in XML format. A file is created for each distribution target. The files are written to an Audit & Control table, ready for the next process (the Loader).

The following file types exist:

- Subscriber Offers
- Subscriber Parameters
- Subscriber Parameters Offers
- Subscriber Activities (in incremental and manual run modes)
- Generic – Information that can be inserted into tables by the loader, without using the Pricing Engine APIs, and distributed to all partitions (such as CUG tables).
- Partition – Information that is inserted into tables by the loader, without using the Pricing Engine APIs, but is duplicated for each partition because the PARTITION_ID field is required.

These file types are described in detail in the subsections that follow.

Subscriber Offers File Type

The loader creates a Customer Offers entity from each tag in the file, using a Pricing Engine API.

Below is an example of a Subscriber Offers output XML file:

```
<SubscriberOffersInformation>
<CustomerOffers>
  <Attribute name="Subscriber Number">
    <Value value="6062"/>
  </Attribute>
  <Attribute name="Agreement offers instance">
    <Value value="27862"/>
  </Attribute>
  <Attribute name="Effective activity code">
    <Value value="14"/>
  </Attribute>
  <Attribute name="Expiration activity code">
    <Value value="14"/>
  </Attribute>
  <Attribute name="Offers">
    <Value value="19619"/>
  </Attribute>
  <Attribute name="Offers Agreement ID">
    <Value value="6062"/>
  </Attribute>
```

```
<Attribute name="Offers effective date">
  <Value value="2004-05-01 00:03:29"/>
</Attribute>
<Attribute name="Offers expiration date">
  <Value value=""/>
</Attribute>
<Attribute name="Offers instance">
  <Value value="27862"/>
</Attribute>
<Attribute name="Offers Level ID">
  <Value value="0"/>
</Attribute>
</CustomerOffers>
</SubscriberOffersInformation>
```

Subscriber Parameters File Type

The Subscriber Parameters file includes parameters at the subscriber level. The loader creates a Customer Parameters entity from each tag in the file, using a Pricing Engine API. Below is an example of a Subscriber Parameters output XML file.

```
<SubscriberParametersInformation>
<CustomerParameters>
  <Attribute name="Subscriber ID">
    <Value value="6062"/>
  </Attribute>
  <Attribute name="Effective date">
    <Value value="2004-05-01 00:03:29"/>
  </Attribute>
  <Attribute name="Expiration date">
    <Value value="2015-01-01 00:00:00"/>
  </Attribute>
  <Attribute name="Change cycle indicator">
    <Value value="N"/>
  </Attribute>
  <Attribute name="Customer ID">
    <Value value="14362"/>
  </Attribute>
  <Attribute name="Customer business entity">
    <Value value="0"/>
  </Attribute>
  <Attribute name="Customer offer currency">
    <Value value="USD"/>
  </Attribute>
```

```
</Attribute>
<Attribute name="Cycle code">
  <Value value="3"/>
</Attribute>
</CustomerParameters></SubscriberParametersInformation>
```

Subscriber Parameters Offers File Type

The Subscriber Parameters Offers file includes parameters at the offer level. The loader creates a Customer Parameters entity from each tag in the file, using a Pricing Engine API.

Below is an example of a Subscriber Parameters offers output XML file.

```
<SubscriberParametersOffersInformation>
<CustomerOfferParameters>
  <Attribute name="Offer level subscriber ID">
    <Value value="6063"/>
  </Attribute>
  <Attribute name="Offer instance">
    <Value value="27963"/>
  </Attribute>
  <Attribute name="Offer level effective date">
    <Value value="2004-05-01 00:04:17"/>
  </Attribute>
  <Attribute name="Offer level expiration date">
    <Value value="2004-05-01 00:08:37"/>
  </Attribute>
</CustomerOfferParameters>
</SubscriberParametersOffersInformation>
```

Subscriber Activity File Type

For the Subscriber Activity file, the loader creates an Activity Event entity from each tag in the file, using a Pricing Engine API.

Below is an example of a Subscriber Activity output XML file.

```
<SubscriberActivityInformation>
<Event type="Subscriber activity">
  <Attribute name="Activity type">
    <Value value="POST"/>
  </Attribute>
  <Attribute name="Activity source">
    <Value value="14"/>
  </Attribute>
  <Attribute name="Customer ID">
    <Value value="239573"/>
  </Attribute>
</Event>
</SubscriberActivityInformation>
```

```

</Attribute>
<Attribute name="Cycle code">
  <Value value="1"/>
</Attribute>
<Attribute name="Issue date">
  <Value value="2004-09-02 00:00:00"/>
</Attribute>
<Attribute name="Start time">
  <Value value="2004-09-02 00:00:00"/>
</Attribute>
<Attribute name="Subscriber ID">
  <Value value="239582"/>
</Attribute>
</Event>
<SubscriberActivityInformation>

```

Generic File Type

Each entry in the Generic file can be one of the following:

- Insert – Insert specified fields to the specified table.
- Delete – Delete records from a table according to specified fields.
- Delete and insert – Delete records from a table according to specified fields, and insert specified fields to the specified table.

The Generic file type is distributed to all Rating partitions with the same data. Only one file is created on the disk, but there is an entry in the A&C tables for each Rating partition.

Below is an example of a Generic output XML file.

```

<GenericInformation>
<Generic>
  <Attribute name="EFFECTIVE_DATE">
    <Value value="2004-06-14 00:00:01"/>
  </Attribute>
  <Attribute name="SYS_CREATION_DATE">
    <Value value="NONE"/>
  </Attribute>
  <Statement name="insert"/>
  <Table name="PM1_CYCLE_CHANGE"/>
  <Attribute name="CUSTOMER_ID">
    <Value value="59"/>
  </Attribute>
  <Attribute name="NEW_CYCLE_CODE">
    <Value value="1"/>

```

```
</Attribute>
<Attribute name="OLD_CYCLE_CODE">
  <Value value="2"/>
</Attribute>
</Generic>
</GenericInformation>
```

In this example, an entry from the PM1_CYCLE_CHANGE table will be inserted according to the loader configuration file in the RPR1_XML_CONFIG table.

Partition File Type

The file type is the same as the Generic file type, except the partition file type includes a different Partition ID value for each Rating partition to which the data is distributed. There is a file on disk and a matching entry in A&C for each Rating partition.

Below is an example of a Partition output XML file.

```
<GenericInformation>
<Generic>
  <Statement name="delete_and_insert"/>
  <Table name="PM1_CYCLE_STATE"/>
  <Attribute name="PARTITION_ID">
    <Value value="1"/>
  </Attribute>
  <Attribute name="CYCLE_CODE">
    <Value value="1"/>
  </Attribute>
  <Attribute name="CYCLE_MONTH">
    <Value value="5"/>
  </Attribute>
  <Attribute name="CYCLE_YEAR">
    <Value value="2004"/>
  </Attribute>
  <Attribute name="SYS_CREATION_DATE">
    <Value value="2004-10-11 00:31:46"/>
  <Attribute name="CYCLE_STATE">
    <Value value="0"/>
  <Attribute name="CYCLE_START_DATE">
    <Value value="2004-10-11 00:31:46"/>
  <Attribute name="CYCLE_CLOSE_DATE">
    <Value value="2004-10-11 00:31:46"/>
  <Attribute name="LOCK_FOR_NEW_IND">
    <Value value="N"/>
  </Attribute>
</Generic>
</GenericInformation>
```



```

    <Attribute name="LOCK_FOR_UPD_IND">
      <Value value="N"/>
    <Attribute name="ARCHIVED_IND">
      <Value value="$ARCHUVE_IND"/>
    <Attribute name="MAINTENANCE_STATUS">
      <Value value="C"/>
    <Attribute name="PREV_CYCLE_INST">
      <Value value="4"/>
    <Attribute name="NEXT_CYCLE_INST">
      <Value value="6"/>
    </Attribute>
  </Generic>
</GenericInformation>

```

In this example, an entry from the PM1_CYCLE_STATE table will be deleted and then inserted with new values according to the key specified in the loader configuration file in the RPR1_XML_CONFIG table. The information will be inserted with a Partition ID of "1."

Customer Extract Configuration File

The Customer Extract configuration file maintains the information needed for Customer Extract in all modes. The file is divided into seven sections, each of which is described below.

Configuration Extract Definition

This part of the Customer Extract configuration file contains a list of configuration extract elements, in the two definition sections described below.

Full Extract Definition

This section of the Customer Extract configuration file contains a list of full extract elements. Each describes an area of data to extract (for example, offers, parameters, etc.) and ranges in that area (such as offers of cycle 1, offers of cycle 2, and so on).

The following is a sample of this section:

```

<Extract name="ExtractAgreementOffers"
ScriptName="FullExtractAgreementOffers.xml"
FormatName="FullFormatSubscriberOffers">
  <ExtractParameters>
    <FixedParameters>
      <FixedParameter name="HistoricalDate"
value="2002-01-01 00:00:00"
type="date"/>
    </FixedParameters>
  </ExtractParameters>
</Extract>

```

The script name attribute refers to the generic extract script entry in the RPR1_XML_CONFIG table.

The format name attribute refers to the formatting section name in the configuration XML (described in “Format Definitions“ below).

The extract parameter elements map a list of values to be assigned to the generic extract SQL, in order to define a range of data for a specific query.

Provider Extract Definitions

This section of the Customer Extract configuration file contains a list of provider extract elements. Each describes an area of data to extract (for example, offers, parameters, etc.) and ranges in that area (such as offers of cycle 1, offers of cycle 2, and so on).

The following is a sample of this section:

```
<Extract name="ProviderExtractOffers"
ScriptName="ProviderExtractOffers.xml"
FormatName="ProviderFormatOffers">
    <ExtractParameters>
        <FixedParameters/>
    </ExtractParameters>
</Extract>
```

The script name attribute refers to the generic extract script entry in the RPR1_XML_CONFIG table.

The format name attribute refers to the formatting section name in the configuration XML (described in “Format Definitions“ below).

Manual Mark Definitions

This section of the Customer Extract configuration file contains ManualMarkExtract elements that describe how to extract technical events that will mark the subscriber for rerating.

The following is a sample of this section:

```
<Extract name="ExtractManualMark"
ScriptName="ManualMarkExtract.xml"
FormatName="ManualMarkFormat">
    <ExtractParameters>
        <FixedParameters>
            <FixedParameter name="Cycle"
externalValue="ManualMarkCycleCode"
value="" type="long"/>
            <FixedParameter name="OffersList"
externalValue="ManualMarkOfferList"
value="" type="stringlist"/>
            <FixedParameter name="FromDate"
externalValue="ManualMarkFromDate"
value="" type="string"/>
        </FixedParameters>
    </ExtractParameters>
</Extract>
```

```
        <FixedParameter name="ToDate"
        externalValue="ManualMarkToDate"
        value="" type="string"/>
    </FixedParameters>
</ExtractParameters>
</Extract>
```

The script name attribute refers to the generic extract script entry in the RPR1_XML_CONFIG table.

The format name attribute refers to the formatting section name in the configuration XML (described in “Format Definitions“ below).

The extract parameter elements map a list of values to be assigned to the generic extract SQL, in order to define a range of data for a specific query.

The externalValue tag refers to a value or list of values from the envelope configuration file.

Incremental Extract Definitions

This section of the Customer Extract configuration file lists IncrementalExtract elements. Each describes an area of data to extract (for example, offers, parameters, etc.) and ranges in that area (such as offers of cycle 1, offers of cycle 2, and so on).

The following is a sample of this section:

```
<Extract name="ExtractSubscriberOffersBySubscriber"
ScriptName="IncExtractSubscriberOffersBySubscriber.xml"
FormatName="IncFormatSubscriberOffers">
    <ExtractParameters>
        <TRBParameters>
            <TRBParameter name="Subscriber"
            TRBFieldName="Subscriber Number"/>
        </TRBParameters>
        <FixedParameters>
            <FixedParameter name="HistoricalDate"
            value="2002-01-01 00:00:00"
            type="date"/>
        </FixedParameters>
    </ExtractParameters>
</Extract>
```

The script name attribute refers to the generic extract script entry in the RPR1_XML_CONFIG table.

The format name attribute refers to the formatting section name in the configuration XML (described in “Format Definitions“ below).

The extract parameter elements map a list of values to be assigned to the generic extract SQL, in order to define a range of data for a specific query.

The data can be retrieved from either of two sources, Transaction Broker transactions or fixed parameters in the XML file.

Transaction Broker parameters refer to values in the transactions. The TRBFieldName refers to the ConstantField element in the Transaction Broker Configuration XML.

Format Definitions

This section of the Customer Extract configuration file contains format elements that describe how to build the output of the extracted data.

The following is a sample of this section:

```
<Format name="FullFormatSubscriberOffers"
FormatType="GenericDataFormatter"
FormatFileName="SubscriberOffers"
recordName="CustomerOffers" recordType="">
  <ExtractCursorsRef>
    <ExtractCursorRef
      name="FullSubscriberOffers"/>
  </ExtractCursorsRef>
  <FormatFields>
    <FixedFields>
      <FixedField name="Offers Level ID"
        value="0"/>
    </FixedFields>
  </FormatFields>
  <KeyFields>
    <KeyField name="Subscriber Number"/>
  </KeyFields>
  <Distribution>
    <DistributionField name="Partition ID"/>
    <DistributionField name="New Partition ID"/>
  </Distribution>
</Format>
```

The FormatType attribute refers to the class name that handles the output.

The FormatFileName attribute is the output file initial.

RecordName is the opening tag of the specified format in the output XML file.

The extractCursorRef refers to the extract cursor definition (described in “Extract Cursor Definition“ below).

Key fields are of three types in the format:

- Fixed fields – supplied in this file
- Transaction Broker fields – Taken from the transaction
- Key fields – Keys used to separate entities in the output file

Distribution fields are keys that divide the output files for distribution to different destinations.

File Format Definition

This section of the Customer Extract configuration file describes the structure of the physical output files.

The following is a sample of this section:

```
<FormatFile FormatFileName="SubscriberOffers"
FileType="DefaultXML"
FormatFileMainTagName="SubscriberOffersInformation"
FormatFileAlias="sub_offers"/>
```

Extract Cursor Definition

This section of the Customer Extract configuration file describes the fields to be written to the output file. The Formatter uses the extract cursor as a link between the generic extract definition and the format definition. Each field in the extract has its index and type in the generic extract output.

The following is a sample of this section:

```
<ExtractCursor name="FullSubscriberParameters1">
  <ExtractFields>
    <ExtractField name="Subscriber ID"
extractOutputIndex="1" type="longKey"/>
    <ExtractField name="Subscriber effective
date" extractOutputIndex="3" type="date"/>
    <ExtractField name="Subscriber expiration
date" extractOutputIndex="4" type="date"/>
    <ExtractField name="Customer ID"
extractOutputIndex="2" type="longKey"/>
    <ExtractField name="Customer business entity"
extractOutputIndex="5" type=""/>
    <ExtractField name="Customer offer currency"
extractOutputIndex="6" type=""/>
    <ExtractField name="Change cycle indicator"
extractOutputIndex="9" type=""/>
    <ExtractField name="Partition ID"
extractOutputIndex="10" type=""/>
    <ExtractField name="New Partition ID"
extractOutputIndex="11" type=""/>
    <ExtractField name="Cycle code"
extractOutputIndex="7" type=""/>
  </ExtractFields>
</ExtractCursor>
```

Transaction Broker Transaction Configuration File

The Transaction Broker transaction configuration file contains the information needed for extracting data from Transaction Broker transactions. The file includes all Transaction Broker transactions, with full paths of the required fields in the Transaction Broker XML file.

The Transaction Broker fields are used in the extract queries to retrieve data from the database, or by the Formatter to be inserted into the final XML output. The fields can be constant or variable fields. Variable fields indicate fields that occur more than once in the Transaction Broker XML file.

The Transaction Broker transaction is converted to Transaction Info. The Transaction Info elements are defined in the configuration file. Each element can be an extract or format element. The extract element retrieves information from the database and is directed to the appropriate format element in the Customer Management extract configuration file. The format element only formats information from the Transaction Broker XML file to the output XML file. If the format element requires a variable Transaction Broker field, the command splits the element into a vector of elements according to the number of occurrences of the Transaction Broker variable field in the Transaction Broker XML file.

The following is a sample of this file:

```
<TRBTransaction name="MOVE_SUB">
  <ConstantFields>
    <ConstantField name="Subscriber Number"
      path="/TRB_TRX/DATA/MoveSubscriber/Subscriber
        /SubscriberContext/SubscriberNo"
      type="long"/>
    <ConstantField name="Customer ID"
      path="/TRB_TRX/DATA/MoveSubscriber/Subscriber
        /SubscriberContext/CustomerNo" type="long"/>
    <ConstantField name="Old Customer ID"
      path="/TRB_TRX/DATA/MoveSubscriber/OldCustomer
        Id/CustomerNo"/>
    <ConstantField name="Activity source"
      path="/TRB_TRX/DATA/CmHeaderTransaction/Activity
        GroupId" type="long"/>
    <ConstantField name="Issue Date"
      path="/TRB_TRX/HEADER" attr="IssueDate"
      type="date"/>
    <ConstantField name="Start time"
      path="/TRB_TRX/HEADER" attr="EffectiveDate"
      type="date"/>
    <ConstantField name="Old Price Plan"
      path="/TRB_TRX/DATA/MoveSubscriber/OldPricePlan/
        Soc" type="long"/>
    <ConstantField name="Old Price Plan Instance"
      path="/TRB_TRX/DATA/MoveSubscriber/OldPricePlan/
        OfferInstanceId" type="long"/>
```

```

    <ConstantField name="Partition ID"
    path="/TRB_TRX/DATA/MoveSubscriber/RelatedCustomerInfo/
    CustomerPartitionInfo/PartitionId"
    type="long"/>
    <ConstantField name="New Partition ID"
    path="/TRB_TRX/DATA/MoveSubscriber/RelatedCustomerInfo/
    NewCustomerPartitionInfo/PartitionId" type="long"/>
    <ConstantField name="Bill Cycle"
    path="/TRB_TRX/DATA/MoveSubscriber/CustomerBillingCycleInfo/
    BillCycleNo" type="long"/>
    <ConstantField name="New Cycle"
    path="/TRB_TRX/DATA/MoveSubscriber/RelatedCustomerInfo/
    NewCustomerBillingCycleInfo/BillCycleNo" type="long"/>
    <ConstantField name="New Price Plan"
    path="/TRB_TRX/DATA/MoveSubscriber/PricePlan/Soc" type="long"/>
    <ConstantField name="New Price Plan Instance"
    path="/TRB_TRX/DATA/MoveSubscriber/PricePlan/OfferInstanceId"
    type="long"/>
</ConstantFields>
<VariableFields>
    <VariableField name="Removed Offer id"
    path="/TRB_TRX/DATA/MoveSubscriber/RemovedServices/Service/Soc"
    type="long"/>
    <VariableField name="Removed Offer instance"
    path="/TRB_TRX/DATA/MoveSubscriber/RemovedServices/Service/
    OfferInstanceId" type="long"/>
    <VariableField name="Added Offer id"
    path="/TRB_TRX/DATA/MoveSubscriber/AddedServices/Service/Soc"
    type="long"/>
    <VariableField name="Added Offer instance"
    path="/TRB_TRX/DATA/MoveSubscriber/AddedServices/Service/
    OfferInstanceId" type="long"/>
</VariableFields>
<TRBTransactionElements>
    <TRBTransactionElement
    command="TrxElemBuilderCommandTRBGeneric"
    extract="ExtractSubscriberOffersBySubscriber"
    />
    <TRBTransactionElement
    command="TrxElemBuilderCommandTRBGeneric"
    extract="ExtractSubscriberParametersBySubscriber"
    />
    <TRBTransactionElement
    command="TrxElemBuilderCommandTRBGeneric"
    extract="ExtractSubscriberParametersOffersBySubscriber"
    />

```

```

<TRBTransactionElement
command="TrxElemBuilderCommandTRBActivity"
format="IncFormatSubscriberActivityPreOffers"
/>

<TRBTransactionElement
command="TrxElemBuilderCommandTRBActivity"
format="IncFormatSubscriberActivityPostOffers"
"/>

<TRBTransactionElement
command="TrxElemBuilderCommandTRBActivity"
format="IncFormatSubscriberActivityOldPricePlan"
/>

<TRBTransactionElement
command="TrxElemBuilderCommandTRBActivityMoveSub"
format="IncFormatSubscriberActivityNewPricePlan"
/>

</TRBTransactionElements>
</TRBTransaction>

```

Parameter Definition in Product Catalog

Both dynamic and static parameters should be defined in the Customer entity, which resides in the Implementation Repository of the Product Catalog.

The following Product Catalog screen depicts the customer attribute definitions:

Customer
Customer Telco

Global Offer Level

Attributes

Name	Type
Subscriber ID	Agreement ID
Effective date	Date
Expiration date	Date
Cycle code	Cycle code
Customer ID	Customer ID
Customer offer currency	Currency code
Customer business entity	Business entity
Change activity date	Date
Friend numbers	Telephone number
Family plan	Family plan ID
Activation date	Date
Spending limit	Monetary amount

Description:

Attribute properties

CM properties

Guiding subscriber in	Yes
Is equipment	
Is mandatory	
Is resource	
Is transient	
Mandatory indicator	No
Parameter category	Parameter
Parameter type	Number
Populate level	PC
Should be displayed	
Transient indicator	No

General

Default value	
Dimensions	None
Kind	Core

CM properties

Additional dynamic properties: - +

Figure 3-1: Defining Parameters in the Product Catalog

The following are defined:

- Attribute name (for example, “Friend number” for an F&F plan)
- Attribute type (for example, a telephone number)

The dynamic and static attributes are defined in the same window. The difference is that for dynamic attributes, the Customer Management properties (at the left side of the screen) should be defined. These properties are instructions for the Customer Management front end regarding how to populate the attribute. For static parameters, the Customer Management properties remain empty.

Dynamic Parameter Definition

If one of the PITs of an offer uses a dynamic parameter, and the offer is attached to a subscriber or agreement, Customer Management displays the parameter for the CSR to fill in. The parameter name and parameter value are stored in the CM1_AGREEMENT_PARAM table.

Customer Extract extracts all parameters from this table and populates them in the Rating Subscriber Parameters table.

4. CUSTOMER LOAD

Customer Load runs on different Rating partitions, one process per Rating partition. Offline rater envelope instances are not limited to a single cycle, and can handle all cycles that are stored in a particular partition.

The Customer Load process performs the following:

- Loads customer data to Rating customer tables.
- Sends subscriber activity events for Pricing Engine processing, in order to mark the subscriber for rerating when required.

Process Flow

The Load process performs the following steps:

1. Connects to the Customer Rating data store. The data storage may be Oracle or an in-memory database, depending on the configuration of the environment.
2. Retrieves input files (Subscriber Offer, Subscriber Parameter, Subscriber Parameters Offers, Subscriber Activity, Generic, and Partition) from Audit & Control tables. The files are sorted by subscriber ID, so that the subscribers are processed synchronously.
3. Initializes the Pricing Engine.
4. Performs the following for each subscriber in the files:
 - a. Creates a Subscriber Offers entity, Subscriber Parameter entity, and Subscriber Parameters Offers entity from the matching XML files.
 - b. Calls the Pricing Engine using the Store mode. In this mode, the Pricing Engine only stores data. It stores the Subscriber Offer entity, Subscriber Parameters entity, and Subscriber Parameters Offers entity.
 - c. Creates subscriber pre- and post-activity event objects (see next section) from the Subscriber Activity XML file.
 - d. Sends the Subscriber Activity event to the Pricing Engine. The event is processed by the Pricing Engine, as are all other rating events. The core Implementation Repository provides the PIT (and computation case within the PIT) qualified for the event. In the current implementation, this functionality marks the subscriber for rerating when relevant. (Subscriber activity events are handled in Incremental and Manual modes only).
 - e. Inserts Generic or Partition data from the matching XML files directly into the database objects (see next section).

5. Closes the files.



The files are processed in a synchronized way, one subscriber at a time. The subscriber data is updated to the data storage in one transaction, to guarantee data integrity.

Pre- and Post-Activity Events

Customer Management activities are mapped to pre and post activity events, according to which the Guiding to Service process selects offers for the subscriber activity event. The pre-activity events analyze the old customer information and offers, while the post-activity events use the new information.

The Extract process that creates the subscriber activity changes the effective date of the event: Relative to the Subscriber activity time, one second is subtracted from the pre-activity event, and one second is added to the post-activity event. This results in the old information being selected for the pre-activity event, and the new information being selected for the post-activity event.

For example, when a subscriber is cancelled and all its offers are expired, the allowance offer should trigger Mark for Rerate, so that the allowance is prorated through rerating according to the new expiration date. In this case, the subscriber activity is a pre-activity event, and:

- The customer information is updated and all offers are expired.
- Because the effective date of the subscriber activity event is earlier than the expiration date of the offers, the Guiding to Service module selects all cancelled offers.
- Allowance offers are marked for rerating so that the quota is prorated by rerating.

Generic and Partition File Support

The extract process creates Generic and Partition files as described above. Those files include information which can be inserted by the loader to the tables without the use of the Pricing Engine APIs. The loader uses the configuration file from the RPR1_XML_CONFIG table, which describes the table names and supported operations on each of the tables.

New tables can be defined in the History or Customer databases. In the following example, a table is added to the History database. The name of the new table is PM1_CYCLE_STATE. The operations to be performed by the loader on this table are Delete and Insert. The Loader looks up the record and deletes, inserts, or deletes and inserts according to the statement name.

```
<loaderConfiguration>
  <dataBaseList name="history">
    <table name="PM1_CYCLE_STATE">
      <statement name="delete_and_insert"
        type="delete_and_insert">
```

```

        <field name="PARTITION_ID"
        type="int" useForDelete="y"/>
        <field name="CYCLE_CODE"
        type="int" useForDelete="y"/>
        <field name="CYCLE_MONTH"
        type="int" useForDelete="n"/>
        <field name="CYCLE_YEAR"
        type="int" useForDelete="n"/>
        <field name="SYS_CREATION_DATE"
        type="date" useForDelete="n"/>
        <field name="CYCLE_STATE"
        type="int" useForDelete="n"/>
        <field name="CYCLE_START_DATE"
        type="date" useForDelete="n"/>
        <field name="CYCLE_CLOSE_DATE"
        type="date" useForDelete="n"/>
        <field name="LOCK_FOR_NEW_IND"
        type="char" useForDelete="n"/>
        <field name="LOCK_FOR_UPD_IND"
        type="char" useForDelete="n"/>
        <field name="ARCHIVED_IND"
        type="char" useForDelete="n"/>
        <field name="MAINTENANCE_STATUS"
        type="char" useForDelete="n"/>
        <field name="PREV_CYCLE_INST"
        type="int" useForDelete="y"/>
        <field name="NEXT_CYCLE_INST"
        type="int" useForDelete="y"/>
    </statement>
</table>
</dataBaseList>
</loaderConfiguration>

```

The loader core configuration file supports these tables:

- PM1_CYCLE_STATE
- PM1_USR_GRP_MEM
- PM1_CYCLE_CHANGE

Run Modes

The Customer Load process can run in either of these modes:

- Full – Runs by request. All customer information is loaded into the data storage. A process should run on each Rating server per partition.
- Incremental – Runs as a daemon in the background and updates the data in storage. The Provider and Manual subscriber activity files are also updated in incremental mode.

5. PRODUCT CATALOG INFORMATION

The Product Catalog contains a catalog of rating schemes, offers, and packages. Rating loads all the information into memory during Rating initialization and uses this information when rating each event.

Product Catalog Subsystems

The Product Catalog organizes its information into these areas:

- Implementation Repository
- Offer Catalog
- Pricing Package Catalog
- Service Package Catalog
- Discount Package Catalog
- Auxiliary Repository

Each of these information types is described briefly in this chapter.

Implementation Repository

The Implementation Repository defines the following:

- Elementary types – Data types, such as the volume data type and its unit of measure.
- Complex data structures – Definitions of events, external records, and the customer model, with mapping and dispatching cases for external records and events.
- Pricing item types – Rating logic, performance indicators (PIs), PI extract definitions, and item parameter definitions.
- Qualification criteria – Definitions of criteria for associating events with pricing items for rating purposes.

Pricing Package Catalog and Offer Catalog

A pricing, service, or discount package groups multiple pricing, service, or discount items, respectively, to be sold together.

An offer groups multiple pricing, service, and/or discount packages.

Rating retrieves the following information from the Pricing Package Catalog, Service Package Catalog, discount Package Catalog, and the Offer Catalog:

- Hierarchy of offers, pricing, service, and discount packages, and pricing, service, and discount items
- Specific parameters for pricing, service, and discount items

- The reference from the pricing item to the PIT, or the service item to the service item type (SIT), that is part of the Implementation Repository and defines the rating logic itself.

Auxiliary Repository

The auxiliary repository includes secondary reference data used by the pricing logic. This includes such objects as period sets, special day sets, and tables based on the generic reference table mechanism.

This data is typically loaded from Oracle database tables during the Pricing Engine load. However, some of this data (data specific to the customization layer) is not known by the core engine and is loaded by extension libraries from other designated data sources.

Product Catalog Data Distribution

Product Catalog distributes its data to a generic distribution table in XML format. The Reference Table Synchronizer (RTS) copies the data to the application distribution tables and activates the relevant callback functions as needed. The callback functions are responsible for extracting the data from the XML files and populating the relevant Rating reference data tables. In addition, some tables reside in the Rating area for greater efficiency.

Pricing Engine, Rating, and Online Charging reference data can be updated at run time. Daemon processes that need ongoing updates of reference data can initiate updates without affecting batch processes, which continue to use the previous reference data until a new version is available.

Product Catalog data loads for offers, packages, and pricing and service items are incremental (that is, only new data are loaded), in contrast to the Implementation Repository, which does not load incrementally.

6. EVENTS

The Pricing Engine rates all types of events in the system. The rating and handling logic of all events is defined in the Product Catalog.

All events may be divided into two principal groups:

- Network usage events – Events whose source is in the network and that result from usage by a subscriber (including online authorization requests and advice-of-charge requests).
- Non-usage events – Events that are generated by different systems (such as Rating or Billing). These are used to control the rating flow (mainly influencing PIs) or to rate non-usage services.



Events are defined in the same way as in the Product Catalog and are handled in the same way by the Pricing Engine.

Network Usage Events

An event can be defined as any use of the operator's network by a user utilizing a service. An event may be caused by any of the following:

- A voice call between two mobile subscribers (as reported by a mobile switching center (MSC))
- A transfer of packet data across a wireless network (as reported by the GPRS gateway support node (GGSN))
- An international voice call by a mobile subscriber (as reported by an MSC)
- An international voice call by a fixed-net subscriber (as reported by a fixed-net switch)
- A query message for the purpose of authorizing a network transaction (as reported by an Intelligent Network platform or other authorization requestor)
- A query message for the purpose of an Advice of Charge request

Each such event may have different properties and may require different processing and rating logic. Events are therefore categorized into different event types. Event types share the same data structure; in other words, all events of the same event type have the same attributes.

For example, a data transaction contains attributes, such as Total Volume and Uplink Volume, that have no significance in a regular voice event. Two different event types may be defined, one for voice traffic and another for data traffic. The rating of the voice event might depend on the length of the call (duration), whereas the data traffic event might be rated according to the volume.

The event type, together with its attributes, plays a significant role in mapping the event to its rating model and to the specific rate.

Offline Charging Event Processing Flow

Events are received into the event processing flow as files from the mediation device. The Acquisition & Formatting (A&F) component receives the event and assigns an event type to it. A&F is a rule-based system, in which rules are formulated for each input source to edit, validate, and classify each event. The A&F module transforms the raw input event to the format that Rating requires.

A&F activates the Guiding to Customer process. Guiding to Customer identifies customer attributes derived from the input event detail record (EDR). These attributes are mainly billing information, and includes the customer ID, event distribution per pay channel, and customer cycle code.

Rating is integrated with A&F by sharing data layout definitions. A&F calls Pricing Engine modules to generate the event entity using the Product Catalog definitions for the event type.

A&F distributes the events into different files or queues according to its distribution rules. Since the Rating server population is defined by the partition and customer billing cycle, the common distribution rule is according to the partition, cycle, and number of Rating processes that run on the cycle. (For more details, see the A&F documentation.)

The figure below illustrates the event processing flow.

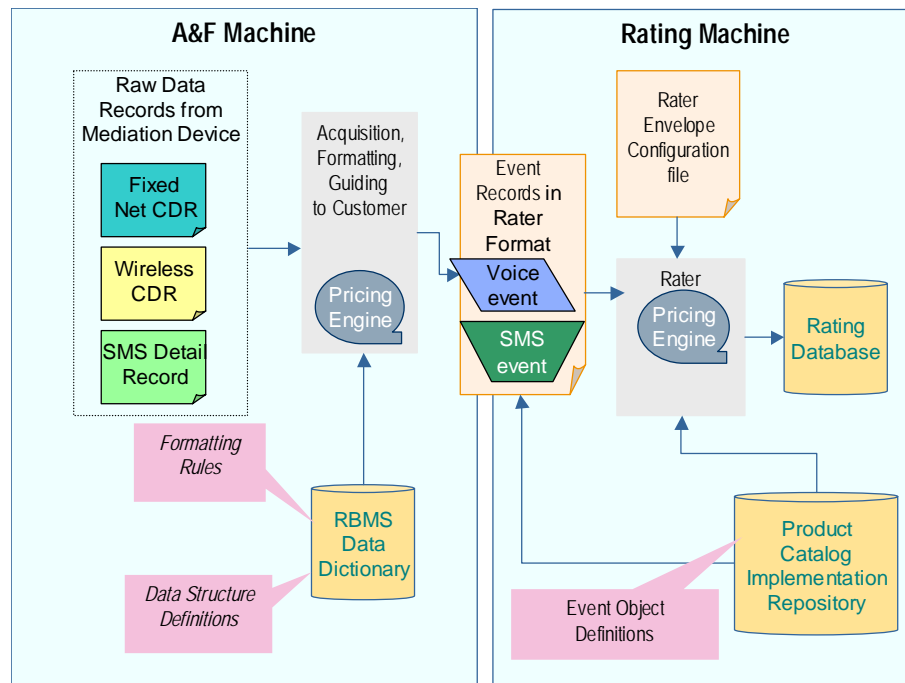


Figure 6-1: Event Processing Flow

Online Charging Event Processing Flow

Request and charging events are transmitted by the mediation device to the Online Charging Formatting and Routing (FR) engine, either directly through a TCP/IP socket-based messaging interface or indirectly through adapters. Relevant rating requests received include those for session-based reservations, charges, and advice of charge.

The Formatting and Routing Engine accepts, formats, and enhances request messages, generating a common event record format. In the enhancement process, it accesses the Customer Identification Data Store to identify the subscriber and account associated with the request (using the Guiding to Customer API). The event record is routed via a socket-based messaging interface to the Charging Engine.

The Charging Engine accesses the Product Catalog Implementation Repository and creates an event object (in Pricing Engine format). The Pricing Engine is called to rate the reservation requests, advice of charge requests, and charge requests. In addition, the Pricing Engine writes rated event information (rated events and PIs) to the Usage database. (For more details, see the Online Charging documentation.)

The figure below illustrates the Online Charging event processing flow.

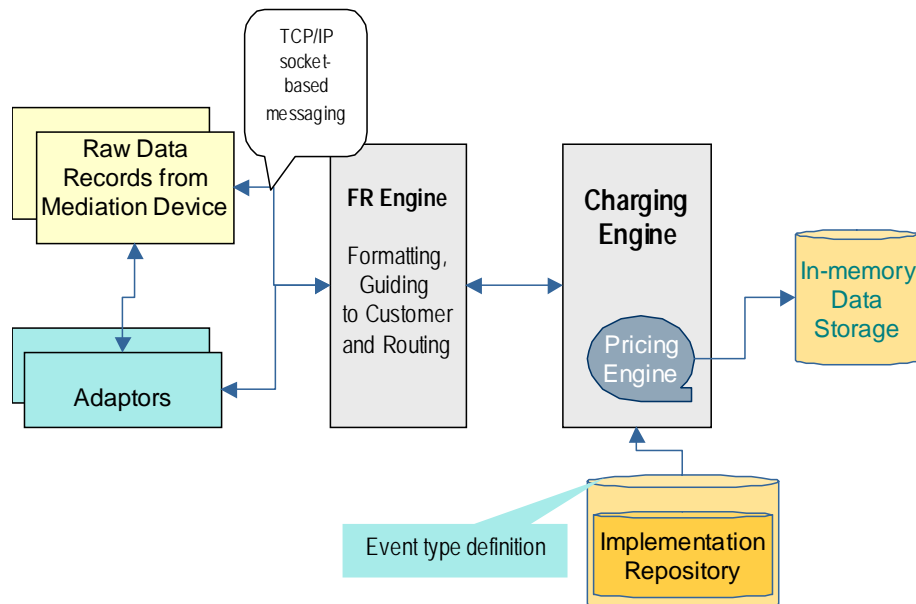


Figure 6-2: Online Charging Event Processing Flow

Non-Usage Events

The concept of events may be extended to:

- Implementing internal flows to control Rating. The core implementation already provides subscriber activity events to control rerating cases. The customization layer can define other internal events to control the granting of benefits or other business events that influence Rating (especially the behavior of performance indicators).

- Implementing recurring charges (RCs) and one-time charges (OCs). The Billing system generates RC events and OC events, which are used to generate RCs and OCs. The events are sent to the Stateless Pricing Engine, which rates the events and creates external records that represent charges or recurring and one-time charge rates.

Event Structure

The event structure is divided into two sections, which comprise different sets of data:

- Fixed structure – Contains base fields common to events. These fields allow for easy database access and Oracle rated event table partitions. Attributes include cycle code, month, and year, Customer ID, and start time.
- Dynamic Structure – Structure and data depend on the event type. The administrator can easily introduce new structures through the Product Catalog.



Some of the event attributes in the dynamic structure are required by Rating and are part of the core Rating.

Defining New Events

Two steps are required to introduce a new event into the system:

- Define the event type data structure for the Pricing Engine in the Product Catalog Implementation Repository.
- Define the rules required to transform the raw input event record received from the network into the event type data structure described above. In Amdocs Billing, this is performed by A&F (for offline charging) or the FR (for online charging).

Event Rating Core Attributes

The attributes described below must be populated for each event type. Rating uses these attributes when processing events.

- Cycle code – The customer billing cycle, defined in A&F by guiding rules.
- Cycle month – Determined by Rating according to the current open bill cycle.
- Cycle year – Determined by Rating according to the current open bill cycle.
- Event type ID – Populated in A&F with the Product Catalog event type identifier (for example, voice event type ID or data traffic event type ID).
- Event ID – A&F populates this field with a unique identifier number for the event.
- Network start time – The physical starting date and time of the event

- Start time – A&F populates this field with the same value as in the Network Start Time attribute. If required, this value can be changed during the Recycle process. The attribute is used by Rating in Guiding to Service.
- Service filter – A&F populates this field with a value selected from the A&F (MAF_FEATURE) table. This attribute is used by Rating to qualify an event to a pricing item. Examples of service filters include Mobile to Mobile, Mobile to Fixed, or International for voice calls.
- Subscriber ID – The identifier of the subscriber, determined in A&F by guiding rules according to event attributes, such as IMSI and MSISDN.
- Customer ID – The identifier of the customer to which the subscribers belong, determined in A&F by guiding rules.
- Source ID – The identifier of the input file to which the event belongs.

The following Billing attributes must appear in the event and be populated for each event type when using the Amdocs Billing platform:

- Account – The payer's billing account number.
- Billing Arrangement – The payer's billing arrangement number. A billing arrangement represents a billing document, which is responsible for producing a periodic invoice or bill. Such a document may contain elements from more than one pay channel.
- Pay Channel – The payment channel of the event. The charge amount of the event is referred to this pay channel.

amdocsrating

Rating 6.0 Output Interfaces

1. INTRODUCTION

Amdocs Rating processes events, and stores the results in the Rating tables. These results are the rated events and performance indicators, which are used by downstream and external components.

The components that require the Rating results are:

- Billing – Uses the performance indicators (PIs) of the subscriber to prepare the bill
- Bill Formatter – Needs the rated event information to produce the Call Detail section of the bill
- Customer Management – Displays queries on billed and unbilled usage
- External components, such as:
 - Data Warehouse (DWH)
 - Third-party providers, such as international carriers
 - Value Added Service providers

All data for external components are received from Rating extracts, dispatching activity, and APIs. This approach complies with the basic objectives of Rating, ensuring that:

- Rating is decoupled from other components, permitting the other components to be third-party systems (“Best of Breed” approach)
- External components do not access the Rater area directly, thereby ensuring that external components cannot influence the high performance required by Rating

Purpose and Scope

This part of this book describes the Amdocs Rating output APIs and file interfaces. It further includes related information on other components in order to provide a clear picture of the interfaces.

Elements responsible for providing Rating results to external components described in this part of this book are:

- Rating extracts
- Online queries
- Dispatching mechanism
- Notification mechanism

2. PRODUCT CATALOG DEFINITIONS

External interfaces transfer records with a well-defined structure that can be read by the external systems. This type of record is called the external record, which combines the required attributes with the physical format in which the external system expects to receive the information.

This chapter describes the definitions of the external record and the mapping from a Rating entity (i.e., event or performance indicator) to the external record. These definitions are made in the Product Catalog.

The definition method is common to all output types; therefore this information has been centralized in this chapter.

External Record Layout Type

An external record layout type specifies a set of attributes, similar to other dynamic data structures (entities) defined in the Product Catalog. Since the events and performance indicators are defined dynamically, the external records may be defined accordingly. For example, the Data Usage Charge that is extracted from the Data performance indicator may have a different structure than the Voice Usage Charge that is extracted from the Voice performance indicator. Both may include Total Duration, but the Data Usage Charge may include also a Total Volume attribute.

The Product Catalog screen below depicts an external record layout definition.

Name	T
CET name	<input type="checkbox"/>
Charge type	<input type="checkbox"/>
Charge code	<input type="checkbox"/>
Revenue code	<input type="checkbox"/>
Offer	<input type="checkbox"/>
Amount	<input type="checkbox"/>
Service receiver type	<input type="checkbox"/>
Charge origin	<input type="checkbox"/>
Service receiver ID	<input type="checkbox"/>
Offer instance	<input type="checkbox"/>
Pricing item	<input type="checkbox"/>
Billing arrangement	<input type="checkbox"/>
Effective date	<input type="checkbox"/>

Figure 2-1: External Layout Definition – Charge Entity Type (CET)

Mapping of Rating Entities to External Records

The mapping case defines the mapping of Rating entity (event or PI) attributes to a specific external record. Each mapping case includes one or more handlers that define the mapping logic. The handlers are mainly composed of assignment statements from the event or performance indicator attributes to the external record attributes.

If the Rating entity is mapped to different external record layouts (as a result of different external systems requiring information based on these entities), then multiple mapping cases can be defined. The Extract process, according to the specified order, applies all defined handlers for the external record types specified in the extract configuration.

Event Mapping

The mapping of an event to an external record is defined under the event's definitions (where the PI attributes are defined). Each event might contain a number of mapping cases.

The event mapping handlers can access the following entities:

- Event
- Customer (although this is not recommended from the standpoint of performance, especially when the Customer data resides in an Oracle database).

Performance Indicator Mapping

The mapping of a performance indicator (PI) to an external record is defined in the Pricing Item Type (PIT) definitions. Each PIT may contain a number of mapping cases.

A performance indicator might contain counters defined as complex attributes. In this case, it is necessary to loop over available values and create a separate record for each actual dimension value. For example, Total Duration can be defined with a Period dimension that accumulates duration separately for peak and off-peak segments of events. A record must be created for each period value having a corresponding populated entry in the complex attribute. This function requires iterations on some complex attribute dimensions.

The mapping case enables the definition of one or more iterators. The iterator declares that, when handling this mapping case, the Pricing Engine should invoke all handlers defined in the case iterating over all values in the specified complex attribute dimension.

The PIT mapping handlers can access the following entities:

- Performance Indicator
- Customer (although this is not recommended from the standpoint of performance, especially when the customer data resides in the Oracle database)
- Item parameters

Example: A duration rate PI may have the following mapping cases for different external records:

- Front-end queries
- Billing charges – An iterator on the charge per period vector is defined and may result in more than one record per PI
- Notification – About subscribers that pass a predefined threshold

The Product Catalog screen below depicts the mapping of a performance indicator to an external record.

The screenshot shows a software interface titled "Mapping charge CET" with a "Handler" icon. Below the title is a "General" tab. The main area contains a list of six mappings, each with an "Assign" button, a source field, a "to" button, and a target field. The mappings are:

Source Field	Target Field
Debit	Charge type
Charge code	Charge code
UC	Revenue code
	Charge origin
Subscriber	Service receiver type
Billing arrangement	Billing arrangement

Figure 2-2: Mapping of a Performance Indicator

Formatting

The Formatting library enables applications to format rating data flexibly and efficiently. The Formatting library provides intuitive APIs and is easily integrated into existing applications. The library supports ASCII file output in two formats: flat files and XML files.

The Formatting library supports the formatting of external records or events through easy-to-use APIs that provide maximum flexibility for controlling and changing the output format.

The underlying concept of the Formatting library is the “format.” A format encapsulates the definition of external records (or events), and for each of them provides a list of attributes that describe the way in which the external record or event is to be formatted.

The Formatting library supports the extract of data with the following layouts:


- Separated values (“field1, field2...fieldN”) – Fields are written to the formatted record one after another. The field separator value can be defined by implementers.
- Fixed positions (“field1 field2 field3”) – Each field is written according to a pre-defined offset.

- Name=value combination (field1=FIELD1 Field2=FIELD2...) – Implementers can extract the data as name-value pairs.


The basic structure of the Formatting Definition file is made up of the following components:

- Format – The name and type of the format as required by the application; for example, PI Extract for Billing.
- Formatting information, including:
 - File structure – A set of attributes that describe the general file structure.
 - File records – A declaration of the different external records or events that participate in the current format.
 - Record structure – A detailed description of each external record or event. The description should include a list of the fields and their attributes.





























Example: The format “My Format” is built from two external records (UC Duration CET and UC Data CET). These external records are displayed in Figure 2-3 and Figure 2-4 below:


UC duration CET
 External record
 Telco

General

Based on:  UC base CET Go

Attributes

	Name	Type
▶	 CET name	 String
	 Charge type	 Charge type
	 Charge code	 Charge code
	 Revenue code	 Revenue type
	 Offer	 Catalog ID
	 Amount	 Monetary amo
	 Service receiver type	 Service receiv
	 Charge origin	 Charge origin
	 Service receiver ID	 Agreement ID
	 Offer instance	 Offer instance
	 Pricing item	 Catalog ID
	 Billing arrangement	 Billing arrange
	 Effective date	 Date
	 Amount currency	 Currency code

Description:

Attribute properties

General

Charge discount at	
Default value	
Dimensions	None
Kind	Core
Mandatory	True

PC properties

Attribute category r	
----------------------	--

General

Additional dynamic properties:
 -
+

Figure 2-3: Definition of UC Duration CET External Record in Product Catalog

UC data CET
External record

Telco

General

Based on: [Go](#)

Attributes

Name	Type
CET name	String
Charge type	Charge type
Charge code	Charge code
Revenue code	Revenue type
Offer	Catalog ID
Amount	Monetary amo
Service receiver type	Service receiv
Charge origin	Charge origin
Service receiver ID	Agreement ID
Offer instance	Offer instance
Pricing item	Catalog ID
Billing arrangement	Billing arrange
Effective date	Date
Amount currency	Currency code

Description:

Attribute properties

General

Charge discount att	
Default value	
Dimensions	None
Kind	Core
Mandatory	True

PC properties

Attribute category r	
----------------------	--

General

Additional dynamic properties:

Figure 2-4: Definition of UC Data CET External Record in Product Catalog

Formatting Configuration XML File Layout

The following XML data needs to be defined in order to use “My Format”:

```
<?xml version="1.0" encoding="UTF-8" ?>
<formats>
  <format type="FLAT" value="My Format"
    entity_type="EXTERNAL_RECORD">
    <files_definitions>
      <file>
        <description>Format sample</description>
        <file_structure>
          <begin_offset value="0"/>
          <buffer_size value="20000"/>
          <record_delimiter value="\n"/>
          <field_delimiter value=","/>
          <file_records file_format_type="VARIANT"
            key_field="Entity_Name" def_record_format="">
            <record id="UC data CET" key_value="UC data CET"/>
          </file_records>
        </file_structure>
      </file>
    </files_definitions>
  </format>
</formats>
```



```

        <record id="UC duration CET" key_value="UC duration CET"/>
    </file_records>
</file_structure>
D:\ <record_structure id="UC data CET">
    <field name="Entity_Name" type="STRING" mask="%s">
        <start_position offset="0" keyword=""/>
    </field>
    <field name="CET name" type="RAW" mask="%s"/>
    <field name="Charge type" type="RAW" mask="%s"/>
    <field name="Charge code" type="RAW" mask="%s"/>
    <field name="Revenue code" type="RAW" mask="%s"/>
    <field name="Offer" type="DLONG" mask="%lld"/>
    <field name="Amount" type="DOUBLE" mask="%9.2z"/>
    <field name="Service receiver type" type="RAW" mask="%s"/>
    <field name="Charge origin" type="RAW" mask="%s"/>
    <field name="Service receiver ID" type="DLONG"
mask="%lld"/>
    <field name="Offer instance" type="DLONG" mask="%lld"/>
    <field name="Pricing item" type="DLONG" mask="%lld"/>
    <field name="Billing arrangement" type="DLONG"
mask="%lld"/>
    <field name="Effective date" type="DATE" mask="%s"/>
    <field name="Amount currency" type="RAW" mask="%s"/>
    <field name="Service receiver customer ID" type="DLONG"
mask="%lld"/>
    <field name="Pay channel" type="DLONG" mask="%lld"/>
    <field name="Period name" type="RAW" mask="%s"/>
    <field name="Number of events" type="DLONG" mask="%lld"/>
    <field name="Number of free events" type="DLONG"
mask="%lld"/>
    <field name="First event date" type="DATE" mask="%s"/>
    <field name="Last event date" type="DATE" mask="%s"/>
    <field name="Discount amount" type="DOUBLE"
mask="%9.2z"/>
    <field name="Billing resource type" type="RAW" mask="%s"/>
    <field name="Billing resource ID" type="RAW" mask="%s"/>
    <field name="Volume" type="DOUBLE" mask="%9.2z"/>
    <field name="Free volume" type="DOUBLE" mask="%9.2z"/>
    <field name="Chargeable attribute" type="RAW" mask="%s"/>
    <field name="Volume UOM" type="RAW" mask="%s"/>
    <field name="Additional qualifier" type="RAW" mask="%s"/>
</record_structure>
<record_structure id="UC duration CET">
    <field name="Entity_Name" type="STRING" mask="%s">
        <start_position offset="0" keyword=""/>
    </field>
    <field name="CET name" type="RAW" mask="%s"/>
    <field name="Charge type" type="RAW" mask="%s"/>
    <field name="Charge code" type="RAW" mask="%s"/>
    <field name="Revenue code" type="RAW" mask="%s"/>
    <field name="Offer" type="DLONG" mask="%lld"/>
    <field name="Amount" type="DOUBLE" mask="%9.2z"/>
    <field name="Service receiver type" type="RAW" mask="%s"/>
    <field name="Charge origin" type="RAW" mask="%s"/>

```

```

<field name="Service receiver ID" type="DLONG"
mask="%lld"/>
<field name="Offer instance" type="DLONG" mask="%lld"/>
<field name="Pricing item" type="DLONG" mask="%lld"/>
<field name="Billing arrangement" type="DLONG"
mask="%lld"/>
<field name="Effective date" type="DATE" mask="%s"/>
<field name="Amount currency" type="RAW" mask="%s"/>
<field name="Service receiver customer ID" type="DLONG"
mask="%lld"/>
<field name="Pay channel" type="DLONG" mask="%lld"/>
<field name="Period name" type="RAW" mask="%s"/>
<field name="Number of events" type="DLONG" mask="%lld"/>
<field name="Number of free events" type="DLONG"
mask="%lld"/>
<field name="First event date" type="DATE" mask="%s"/>
<field name="Last event date" type="DATE" mask="%s"/>
<field name="Discount amount" type="DOUBLE"
mask="%9.2z"/>
<field name="Billing resource type" type="RAW" mask="%s"/>
<field name="Billing resource ID" type="RAW" mask="%s"/>
<field name="Duration" type="DOUBLE" mask="%9.2z"/>
<field name="Free duration" type="DOUBLE" mask="%9.2z"/>
<field name="Duration UOM" type="RAW" mask="%s"/>
<field name="Basic service type" type="RAW" mask="%s"/>
</record_structure>
</file>
</files_definitions>
</format>
</formats>

```

This example details the formatting rules for two types of external record. The next section discusses the meaning of each XML tag.

Details of the Formatting Configuration XML File

XML Tag	Description	Attributes	Recurring/Fixed	Comments
<formats>	The configuration file header.	No	Fixed	
<format>	The start declaration of a specific format.	Attributes: <ul style="list-style-type: none"> Value – Displays the name of the format. Type – Displays the ASCII file type. Valid values are FLAT or XML. Entity_type – Specifies whether the format contains a list of events or a list of external records. Valid values are EVENT or EXTERNAL_RECORD. 	Recurring. For each format under the <formats> tag.	Example: <code><format type="FLAT" value="My Format" entity_type="EXTERNAL_RECORD"></code>
<files_definitions>	The header for the IOFW file definition.	No	Fixed	
<file>	The header for the IOFW file definition.	No	Fixed	
<description>	The format description.	A short description of the format.	Fixed	Example: <code><description>Format sample</description></code>
<file_structure>	The header of the general section of the format (format type).	No	Fixed	
<begin_offset>	The offset in the file of where to write the record.	Attributes: value – offset	Fixed	Example: <code><begin_offset value="0"/></code>
<buffer_size>	The maximum size of the record.	Attributes: value – maximum size	Fixed	Example: <code><buffer_size value="20000"/></code>
<record_delimiter>	The separator between records.	Attributes:	Fixed	Example:

XML Tag	Description	Attributes	Recurring/Fixed	Comments
		value – record separator		<record_delimiter value="n"/>
<field_delimiter>	The separator between fields.	Attributes: value – field separator	Fixed	Example: <field_delimiter value=", "/>
<file_records>	The header for the declaration of external records or events included in the format.	Attributes: <ul style="list-style-type: none"> File_format_type – If the format has only one event or external record the value will be FIXED; otherwise it will be VARIANT. Key_field – The name of the field that defines the external record or event. It is always Entity_Name. 	Fixed	Example: <file_records file_format_type="VARIANT" key_field="Entity_Name" def_record_format="">
<record>	The declaration of a specific event or an external record.	Attributes: <ul style="list-style-type: none"> ID – The name of the external record in the context of format. Key_value – The external record or event name as defined in Product Catalog. 	Recurring, for each external record or event in the format. Located under <file_records> tag.	Example: <record id="UC data CET" key_value="UC data CET"/>
<record_structure> >	The header for each set of record details.	Attributes: <ul style="list-style-type: none"> ID – The name of the record as defined in id_attribute in the <record> tag. 	Recurring, for each <record> that was defined in the <file_records> section. Located under <file> tag.	Example: <record_structure id="UC data CET">
<field>	Specific field definition.	Attributes: <ul style="list-style-type: none"> Name – The name of the field. Must be identical to the related field name in the external record or event. 	Recurring, for each field that needs to be formatted. Located under <record_structure >.	Example: <field name="Charge code" type="RAW" mask="%s"/>

XML Tag	Description	Attributes	Recurring/Fixed	Comments
		<ul style="list-style-type: none"> ■ Type – Must be one of the following values: <ul style="list-style-type: none"> • STRING – Only for the Entity_Name field. • DLONG – Integer numbers • DOUBLE – Real numbers • RAW – String • DATE – Date • BOOL – Boolean ■ Mask – Any legal C mask type and one additional type for the special handling of doubles (e.g., %z, used like %f). <p>For the Date type, the following masks are supported:</p> <ul style="list-style-type: none"> • DD – Day in short format • DDDDDD – Day in long format. • MM – Month in short format. • MMMMM – Month in long format. • YY – Year in short format. 		

XML Tag	Description	Attributes	Recurring/Fixed	Comments
		<ul style="list-style-type: none"> • YYYY – Year in long format. • hh – Hour • mm – Minute • ss – Seconds <p>For the BOOL type, the following masks are supported:</p> <ul style="list-style-type: none"> • YN – Y or N • TF – True or False • YESNO – Yes or No <p>The default is 0 or 1.</p>		
<start_position>	Specific field properties.	<p>Attributes:</p> <ul style="list-style-type: none"> ■ Start_position – Sets a fixed location of the field in the output record. This attribute is not mandatory and can be omitted. ■ Keyword – For Name=Value presentation, represents the key values. 	Fixed. The element can be omitted if not needed.	Example: <code><start_position offset="0" keyword=""/></code>

3. EVENT AND PERFORMANCE INDICATOR EXTRACTS

This chapter describes the Event and Performance Indicator extract processes.

Extract Process Overview

The Rating Extract process consists of parallel batch processes that are used for massive data extracts from the Rating area. The following are examples for the use of this extract process:

- The Billing process requires the performance indicators of the subscriber to prepare the bill. The records may be extracted using the charge structure already prepared for Billing.
- The Bill Formatter requires the customer's rated events as input to the formatting process to print the Call Details section.
- The extract process may also be used as an incremental data extractor; for example, it may be run daily on the Rated Events table to extract all events that were inserted or updated in the table during the day. These data are then available for reports and the like.

The Rating Extract process handles events and performance indicators used by external components. The extract process is responsible for:

- Extracting the necessary information from the Rating data storage area for the required population.
- Creating and populating a file containing records in an external record layout defined in the Product Catalog.

Figure 3-1 depicts the Rating Extract flow.

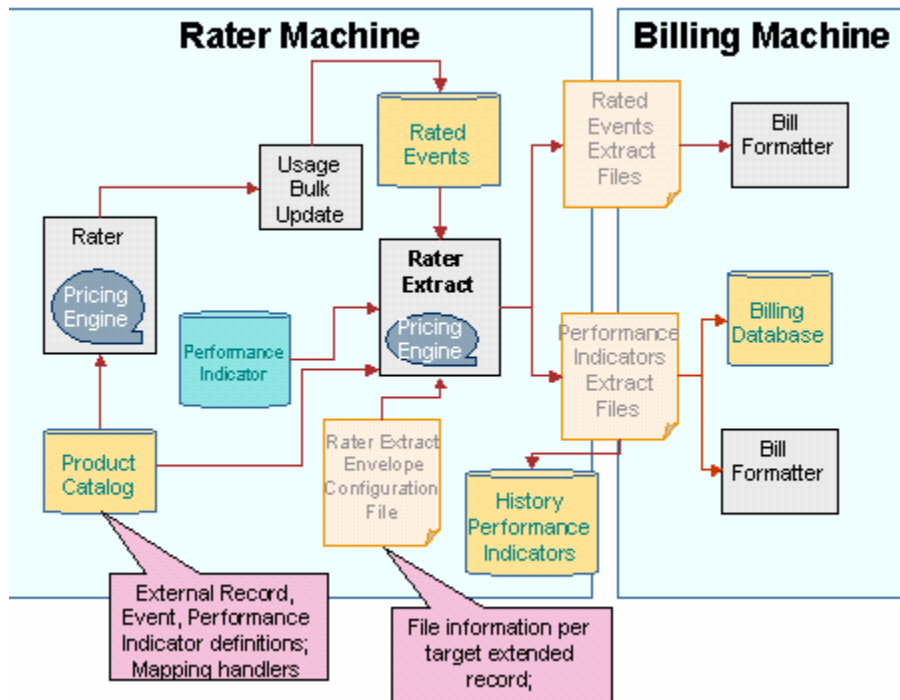


Figure 3-1: Rating Data Extract Flow

Process Input Parameters

The Extract server receives the following operational parameters as input:

- Partition ID – The partition that the server is accessing.

Cycle Extract Mode

For the Cycle extract mode, the “run cycle” request receives the following operational parameters as input:

- Partition ID – The partition that the server is accessing.
- Bill Cycle ID – Bill cycle code, cycle month, and cycle year. This specifies the partition (table instance) from which the data should be extracted.
- Number in Parallel – This specifies the number of concurrent runs requested for the Performance Indicator extract. This parameter is relevant only for the PI extract. In the Event extract, the Rating Extract determines the number of parallel runs based on the subpartition parameter in the Cycle Definition table. (For details, see “Number of Parallel Runs and Rated Event Partitioning” below.)
- From Date/Time and To Date/Time (optional) – The process retrieves all information that was updated in this time interval. This parameter is used by the incremental daily jobs that retrieve the last updated information. The job compares the input interval to the SYS_UPDATE_DATE field.

The format of the Cycle extract run mode is specified in an environment variable:

- CYCLE_PI_FORMAT_NAME for PI extracts
- CYCLE_PI_FORMAT_NAME for Event extracts

The Customer run mode format is included in the header of the customer list.

Customer List Extract Mode

In the Customer Extract run mode, the input is a customer list in a flat text file received from an external system or component. The principal user of this option is the Billing component. The Billing processes run in groups and extract the information according to the groups' population.

Each file can contain a list of customers for more than one cycle. Each cycle group is started with new header.. The customer list files are retrieved from Audit & Control.

File Format

The following is an example of the input Customer List file format:

```
<?xml version="1.0" encoding="UTF-8"?>
<files_definitions xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="~ccip/bb/ggninfra/v60_0/pub/fil
e_format_def.xsd">
  <file>
    <description>List of Customers marked for PI Event
    Extract</description>
    <file_structure>
      <begin_offset value="0"/>
      <buffer_size value="131072"/>
      <record_delimiter value="\n"/>
      <field_delimiter value=";">
      <file_records file_format_type="FIXED" key_field=""
      def_record_format="">
    </file_structure>
    <record_structure id="">
      <field name="f_formatName" type="STRING"/>
      <field name="f_cycleCode" type="INT"/>
      <field name="f_cycleInstance" type="INT"/>
      <field name="f_cycleYear" type="INT"/>
      <field name="f_customerId" type="LONG"/>
      <field name="f_partitionId" type="LONG"/>
      <field name="f_Bald" type="LONG"/>
    </record_structure>
  </file>
</files_definitions>
```

File Example

The following is an example of the input file:

```
Event Format;100;11;2004;0;0;0;
;100;11;2004;451;1;6;
;100;11;2004;500;1;4;
;100;11;2004;600;1;11;
Event Format;1;12;2004;0;0;0;
;1;12;2004;300;1;15;
```

Process Flow

The flow of the Rating Extract process is described below, for both the Cycle and Customer extract modes. In both modes, a daemon listens for an extract request. Once a request is accepted by the daemon, the request is processed by the Rating Extract process.

Cycle Extract Mode FlowFor the Cycle extract mode, the Extract process:

6. Reads the extract configuration data.
7. Retrieves all external records from the configuration file.
8. Provides a cursor to enable the selection of all events and PIs according to the Customer ID and subscriber.

If the extract is taken from the Rated Events table, it is performed for a group of subpartitions. For a detailed description of this process, refer to the “Number of Parallel Runs and Rated Event Partitioning” section below.

9. While reading records from the cursor, the extract process:
 - a. Calls the Pricing Engine to read the next record. According to the format, the process determines whether an event or an external record is to be fetched:
 - i. The Pricing Engine fetches the next record (rated event or performance indicator) from the cursor.
 - ii. In the case of an external record, the Pricing Engine maps the rated event or performance indicator to the input external records. The mapping cases used are those related to the external record list given in the configuration file.

For example, if the performance indicator is related to the ‘Voice Flat Rate’ PIT and the requested external record is ‘Billing Voice’ and ‘Billing Data,’ the Pricing Engine searches the ‘Voice Flat Rate’ PIT for extract cases matching one of the input external records types and finds the ‘Billing Voice,’
 - iii. The Pricing Engine returns the mapped record to the Rating Extract process.

- b. Uses the Formatting library format and writes the event to one of the following:
 - Plain ASCII
 - XML
10. Closes the cursor.
11. Closes the file.

Recovery

For each extract, a table indicates the processing status of each subpartition. When a subpartition's processing is completed, its status in the table is marked as closed. If recovery is needed, the daemon brings up the records in the subpartitions that are not closed for processing.

Customer Extract Mode Flow

In the Customer extract mode, the Rating Extract processes read the customer lists from the Audit & Control table. The customer files are received from Billing after being divided into groups. There are several types of files:

- Files for event extract
- Files for event extract depending on Rerating completion, which are processed only after the Rerate process is complete
- Files for PI extract
- Files for non-usage PI extract
- Files for PI extract depending on Rerating completion, which are processed only after the Rerate process is complete
- Files for non-usage PI extract depending on Rerating completion, which are processed only after the Rerate process is complete

Re-Rated customer groups also are sent to Audit & Control when rerating is completed. The Event extract keeps processing these files as long as the rerating process continues. When all files have been rerated, a "done" message is sent to the Event Extract daemon.

For each file for the Customer extract mode, the Rating Extract process:

1. Reads the extract configuration data.
2. Retrieves as input all external record types from the Customer List file.
3. Loads the customer list into a temporary customer ID table (which is in either Oracle or TimesTen, depending on the extract connection). The table is created so that a "Join" can later be performed between the Customer ID table and the extracted table. The table's structure is:
 - Cycle ID
 - Subpartition ID
 - Customer ID

4. Opens a cursor that “joins” the Customer ID table and the extracted table (Rated Event or PI table).

In the Rated Event extract, a “join” is performed for each partition separately in a loop on all subpartitions.

5. While reading records from the cursor:
 - a. Calls the Pricing Engine to read the next record. According to the format, the process determines whether an event or external record is to be fetched, as follows:
 - i. The Pricing Engine fetches the next record (event or performance indicator) from the cursor.
 - ii. In the case of an external record, the Pricing Engine maps the event or performance indicator to the input external records. The mapping cases used are those related to the external record list given in the configuration file.
 - iii. The Pricing Engine returns the mapped record to the extract process.
 - b. Uses the Formatting library format and writes the event to one of the following:
 - Plain ASCII
 - XML
6. Closes the cursor.
7. Closes the file.

For each request, the Event Extract creates an output file and an error file. If any of the records fail, the entire customer processing fails, and this is reported in the error file.

Recovery

After each customer is processed, its information is saved in a file. This way the file contains the last customer whose processing has been completed.

If recovery is needed, the daemon identifies the last customer completed in the file. Then rating continues from that customer until recovery is completed. The Event Extract proceeds normally from there.

Number of Parallel Runs and Rated Event Partitioning

In the Customer extract run mode, each input file is processed by a single thread.

In the Cycle extract run mode, the number of threads processing one extract request depends on whether PIs or events are being extracted:

- Cycle PI extract – the number of parallel threads is specified in the Operational utility.
- Cycle Event extract – the number of parallel threads is calculated by the Extract process according to the subpartition parameter in the Cycle Definition table.

The Rated Events table is partitioned according to the following fields:

- Cycle code
- Cycle run month
- Subpartition ID – This field assigns an event to a subpartition according to the Customer ID number, and is read at initialization from the Rating configuration file).

4. ONLINE USAGE QUERIES

The customer service front end occasionally requires such data from Rating as:

- Rated Events – A list of rated events over a period of time for a given subscriber, or detailed information on a specific event.
- PIs – A list of performance indicators over a period of time for a given subscriber, or detailed information on a specific PI.

Several types of clients use these queries, for example:

- Call center customer service representatives
- Customers of service providers, using Internet-based self-service clients
- IVR machines that provide end users with information on their usage

This chapter provides details about the Rating APIs that support online queries. The APIs access the data storage according to the input criteria (for instance, subscriber, cycle information, and date range), and return a list of either rated events or PIs belonging to the subscriber during the requested period.

The available information is for unbilled and recently billed cycles. The number of billed cycles that can be retrieved depends on the number of history cycles kept in the Usage data storage prior to archiving (usually three months).

Online Query Architecture

The usage query APIs are combined from three layers:

- Front end – The layer that requests the query.
- EJB APIs – The layer that routes the query to the relevant Rating API server (described below).
- Rating APIs – The layer that actually accesses the data storage to retrieve the requested information (described below).

EJB APIs

The EJB API layer runs on the customer machines and receives requests (using WebLogic middleware) from the front end. The client of the Rating Usage API server may be any process working from the same or a different host, an online system, an IVR client, and so on.

Due to the large volume of data, cycles are often maintained in different data storages (such as database table partitions), and on different hosts (such as different remote UNIX servers). The main function of the EJB API layer is to route the query to the relevant Rating server.

Rating API Server

The Rating Usage API server is a daemon process that runs on a Usage server. There may be more than one Usage server, with one server per partition. The Rating API server opens connections to all the data storages that it handles.

The Usage APIs access data storage according to:

- The type of the information required (events or PIs):
 - Events are always retrieved from an Oracle table
 - PIs are retrieved as follows:
 - If the configuration of the environment is Oracle only, then they are always retrieved from the Oracle table.
 - When the PIs of the current cycle reside in the IMDB, then the current unbilled cycle PIs are retrieved from the IMDB, and billed cycles' PIs are retrieved from the Oracle performance indicator table.
- The bill cycle to which the customer belongs.

Rating Online Query APIs

Two types of APIs are used by EJB APIs to access the Rating data. These are internal APIs that are wrapped by external, officially published EJB APIs.

The two API types are:

- Initialization API – This consists of one API that loads the Configuration file and the Product Catalog layout and process definitions into memory. It is invoked to prepare the system for data storage connection and environment initialization.
- Retrieval APIs – These include a number of APIs that access the data storage according to the input criteria, and return a list of either rated events or PIs belonging to the subscriber over the requested period.

The format returned is defined in the external record layout type in the Product Catalog. The Pricing Engine is responsible for mapping the entity to an external record.

The Retrieval APIs are detailed in the subsections that follow.

Get PIs by Customer

This API retrieves a list of all PIs for a given customer and subscriber, and returns it in the requested external record structure.

The calling application determines the required output record structures. For example, the Data performance indicator structure may differ from the Voice performance indicator structure.

Therefore, if the information retrieved is a short summary of all the subscribers' PIs, then the external record is a minimal record with only the principal information. But if a detailed summary is required for the PIs, then more information may be retrieved.

Name

getRatedPerformanceIndicators

Input Parameters

- Bill Cycle ID – Customer’s bill cycle code, cycle year, and cycle instance that defines the PI table to be accessed
- Pagination information:
 - Page size
 - Page number (first page is 1)
- The vector of the desired external record that is to be used for the output
- PI category – the subtype that defines the desired external record where the usage type is ‘P’ and the query name is ‘LIST’
- Customer ID
- Partition ID – the partition where the customer’s PIs reside
- Agreement ID (the subscriber’s ID)
- Additional “where” clause – An optional “where” clause that may be added by the calling application to the “select” statement

Output

A Performance Indicator list in a requested external record structure; this is a vector of name-value objects containing attributes according to the external record structure.

Flow

See “Retrieval API Flow” below.

Get Rated Events by Customer

This API retrieves a list of rated events for a given customer and subscriber, and returns this list in the requested external record structure.

The list may be filtered by event type (Voice, Data, etc.) and time interval.

Name

getRatedEvents

Input

- Bill Cycle ID – Customer’s bill cycle code, cycle year, and cycle instance that defines the Rated Events table to be accessed
- Pagination information:
 - Page size
 - Page number (first page is 1)
- The vector of the desired external record that is to be used for the output
- Event type – The subtype that defines the desired external record where the usage type is ‘E’ and the query name is ‘LIST’; for example, Voice, SMS or MMS

- Customer ID
- Partition ID – the partition where the customer’s events reside
- Agreement ID (the subscriber’s ID)
- BA ID – optional parameter for filtering
- Start Date/Time and End Date/Time – A time interval in which the events are to be extracted. The calling component must verify that the input dates are within the cycle start date and end date
- Additional “where” clause – An optional “where” clause that may be added by the calling system to the “select” statement

Output

Rated events list in the requested external record structure; this is a vector of name-value objects containing attributes according to the external record structure.

Flow

See “Retrieval API Flow” below.

Get Rated Event by ID

This API retrieves information of a single event. The input for the API is the Event ID. The API is usually used to zoom in on the rated event list.

Name

getEventDetails

Input

- Bill Cycle ID – Customer’s bill cycle code, cycle year, and cycle instance
- Customer ID
- Partition ID – the partition where the customer’s event reside
- Subscriber ID
- Event ID – A unique sequence that represents the event in the Rated Events table
- Event type – The subtype that defines the desired external record where the usage type is ‘E’ and the query name is ‘DETAILS’; for example, Voice, SMS or MMS
- The vector of the desired external record that is to be used for the output

Output

A single rated event in the requested external record format; this is a vector of name-value objects containing attributes according to the external record structure.

Flow

See “Retrieval API Flow” below.

Get PI by ID

This API retrieves information about a single performance indicator. The input for the API is the offer instance. The API is normally used for zooming in on the PI list.

Name

getRatedPerformanceIndicatorsDetails

Input

- Bill Cycle ID – Customer’s bill cycle code, cycle year, and cycle instance
- Pagination information:
 - Page size
 - Page number (first page is 1)
- Customer ID
- Partition ID – the partition where the customer’s PIs reside
- Agreement ID (the subscriber’s ID)
- Offer instance – A sequence that represents a unique record in the PI table. The sequence is given by Customer Management when an offer is attached to the agreement. For example, if the subscriber purchases the same duration allowance offer twice, Customer Management will generate different offer instance numbers for each of the entries
- The vector of the desired external record that is to be used for the output
- PI category – The subtype that defines the desired external record, where the usage type is ‘P’ and the query name is ‘DETAILS’.

Output

A single performance indicator in the requested external record format; this is a vector of name-value objects containing attributes according to the external record structure.

Flow

See “Retrieval API Flow” below.

Retrieval API Flow

The following API flow is similar for all APIs:

1. Locate data storage to be accessed.
2. Perform a query using input parameter.
3. Retrieve information from data storage.
4. Map PI to the requested external records (one or many) according to logic defined by mapping handlers of the Product Catalog.
5. Return output in XML format (see Appendix A for Native XML description).

When generating lists, a great deal of information may have to be retrieved; therefore, the API breaks the data into ‘conversations’ with the calling component. For each conversation, a predefined block of information is sent to the calling component. The calling component sends to the API the page size and the requested page. The API in turn:

1. Calculates the number of pages, i.e., the number of retrieved records divided by page size.
2. Locates the requested page in the list and sends the data from this point to the external system.

Rating Reference Tables for Front End

The front-end system uses two reference tables that contain information on the external record and event types while calling the Rating APIs.

The two reference tables are described in the subsections that follow.

Query Output Format Table

The Query Output Format table lists external records used by Usage APIs and relevant information regarding each external record. The table attributes include:

- Usage type:
 - Rated event
 - Performance indicator
- Query Name – The query type:
 - List
 - Details
- Subtype – The type of the retrieved entity in the Product Catalog:
 - For events – The object ID of the event type. This is a unique ID generated by the Product Catalog for each event type (for example, Voice= 34555, Data=78906, etc.).
 - For PIs – The role of the PIT (e.g., Rate, Allowance, etc.)
- External structures – The external record’s name.

Name

RPR1_QRY_OUT_FORMAT

Generic Codes Table

Generic Codes is a Customer Management table containing all the codes used by Customer Management. The Rating Generic Codes table contains the Rating codes. The Rating Generic Codes table is exported to Customer Management and merged with the Customer Management Generic Codes table. Customer Management then uses these codes in the Online Queries screens.

The table is populated with two types of codes:

- Event types – The event type object ID in the Product Catalog, which is the code and the description of the event type (for example, Code = 2345 and Description = Voice).
- Performance indicator category – The role of the PIT to which the performance indicator belongs (for example, Allowance or Rate role; in this case, the code and the description are identical).

Name

RPR1_GENERIC_CODES

5. RATED EVENT DISPATCHING MECHANISM

The rated event dispatching mechanism sends rated events to different targets. Events are dispatched immediately after being rated, so rated events can be sent on an ongoing basis to internal and external destinations.

This chapter details the dispatching mechanism for rated events and describes Product Catalog dispatching definitions.

The following are examples of the use of the rated event dispatching mechanism:

- Sending the rated event to an external system. The external system may be the communications service provider (CSP) to whom the subscriber belongs, the data warehouse, or an international carrier.
- Sending all rated and rejected events from Rating to the Oracle tables.
- Rating an event provided by a service provider or content provider according to the rate plan of the subscriber using the service, and then sending the event to the service provider or content provider account for revenue sharing based on a percentage of the charges.

The dispatching qualification cases and targets are defined in the Product Catalog.

The Rating dispatching mechanism comprises two activities:

- Dispatching – Includes target determination and generation of the output record for each target. Dispatching is performed in Rating by the Pricing Engine.
- Distribution – Distributes the output records to each system as required. Distribution is a dedicated process performed outside of Rating.

The diagram below depicts the Dispatching and Distribution flow.

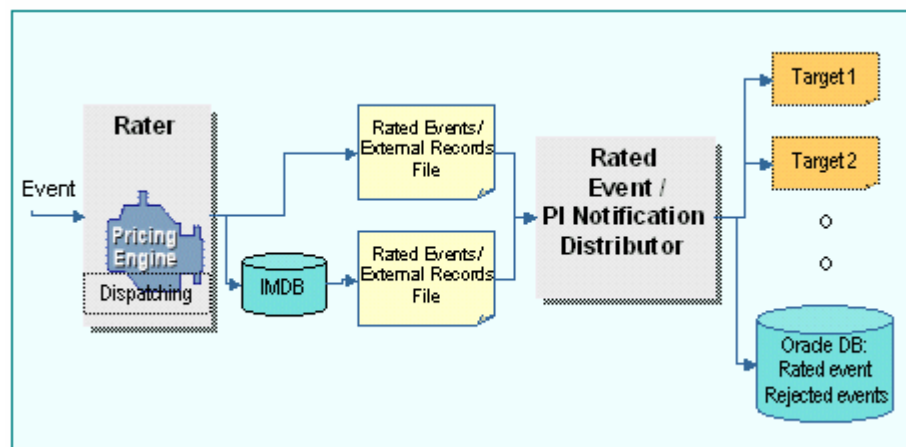


Figure 5-1: Dispatching and Distribution Flow

Product Catalog Definitions

This section describes the different types of dispatching definitions in the Product Catalog.

Dispatching Case

Dispatching is a Product Catalog subnode under the event type definition. It is used for defining the list of targets to which a rated event can be dispatched. Each dispatching case is composed of three elements:

- Qualification criterion
- External record type
- Dispatching target identifier

Qualification Criterion

The qualification criterion qualifies an event type for dispatching to a destination. If no qualification criterion is defined, the dispatching target is automatically qualified for the given event type.

The following entities are available for the qualification criterion:

- The current rated event
- The current customer parameters

External Record Type

The External record type refers to the record to which the rated event is to be mapped. This record is distributed to the dispatching target. The mapping case for the record is taken from the relevant extract section under the given event type.

If no external record type is defined, the rated event object itself will be distributed.

Dispatching Target Identifier

All supported dispatching targets are defined in the Product Catalog in an auxiliary type called Dispatching Target Identifiers. This auxiliary stores information about the supported targets, together with additional information used by the distributing process to distribute the event to its target.

The operational aspects of the various dispatching targets are stored in the Oracle Dispatching Targets reference table. The GUI for creating Dispatching Targets records is part of the Product Catalog. The dispatching target attributes that make up the record include:

- Target ID – The identifier of the dispatching target.
- Description – A text description of the target.
- Logical Format – The Product Catalog entity type used for representing the rated event. The possible values are:
 - External record
 - Event

- **Physical Format** – The physical dispatching format of the rated event. The possible values are:
 - XML – This is the native XML file format (see Appendix A for details)
 - Binary – The binary format
 - Customize – The format defined in the formatting library, which can be flat or XML
 - Bulk Update Format – The update of the Rated Events and Rejected Events tables with the Rating output
- **Dispatching Channel Type** – The possible values are:
 - File – Dispatching to a file
 - Database – Dispatching directly to a database, such as Oracle or TimesTen
- **File Path** – The path where dispatching files are generated
- **File Prefix** – A prefix to the file name of the dispatched files
- **File Alias** – The alias of the dispatching file; this attribute is used by Audit & Control to determine the application that is to receive the dispatching file
- **Max Count** – The maximum number of records in the dispatching files or in the bulk update operation
- **DB Table Name** – The database table to which the events are sent, either the Rated Events table or the Rejected Events table.

Dispatching Case Example

This example shows how all international voice calls made via a certain carrier are dispatched to a file that is sent to the carrier. This dispatching case is defined for a voice event, as follows:

- **Qualification mechanism** – Verifies whether the call is an international call and, if so, identifies the CSP that generated it.
- **Reference structure** – An external record structure is defined by the CSP. The mapping case of this external record is defined in the Event Extract node under the Voice Event Type definition.
- The target is a file called `INTER_CARRIER_NAME`.

Required Dispatching Definitions

The following should always be defined for dispatching and are part of the Rating core definitions:

- Dispatching case for rated events – All rated events should be dispatched to the Rated Events Oracle table. In the core dispatching case, verify that:
 - External record should be empty – The entire event should be dispatched.
 - Qualification criterion should be empty – The dispatching target is automatically qualified for the given event type.
 - Target identifier – Should refer to the Rated Events table.
- Definitions in the Dispatching Target table for:
 - Oracle Rated Events table
 - Oracle Rejected Events table

Dispatching

The Pricing Engine dispatches a rated event after completing all calculations and before writing the event to the output media. Dispatching is performed for all qualified dispatching cases, meaning that one rated event can produce more than one dispatched record.

The dispatch-related activities performed in Rating are discussed below.

Dispatching Qualification

The Pricing Engine scans all dispatching cases that exist for the rated event type. For each dispatching case, it determines whether or not the event is subject for dispatching according to the qualification criterion. An empty qualification criterion is always qualified.

Map Event to External Record

One of the parameters of the dispatching case is the external record.

If the external record parameter is empty, the entire event is dispatched in binary format to the output media.

If the external record parameter is not empty, the rated event is mapped to the external record format according to the extract definitions in the event type node (see below).

Write Dispatched Event

The events and external record, along with the dispatching target identifier for each one, are written to the rated event output media. The rated event output media can differ from one environment to the other, and the possible options are:

- TimesTen
- Files
- Cache memory



For TimesTen, the Replicator daemon process reads the rated event output from TimesTen logs into files. After this, the files are sent to the Distributor.

Distributor

The Distributor is a daemon that reads files ready for processing. The Distributor uses the following algorithm:

1. Fetch the next record from the input file.
2. Determine the associated dispatching target according to the target identifier populated by Rating.
3. Retrieve the target description from the Dispatching Target table. The relevant information is:
 - Channel – File or Database
 - File name, alias, and path – When the channel is File
 - Table name – When the channel is Database
 - Physical format – Either XML, binary, or bulk database update
4. Convert the object to the physical format defined for the dispatching target.
5. Distribute the resulting event to the target system:
 - If the target is a database, such as the Oracle Usage database, the event is inserted into the Rated Events or Rejected Events table (depending on the nature of the event).
 - If the target is a file, the event is written to the file specified in the Dispatching Target table.

When the Distributor finishes processing an input file, all resulting target output files are delivered to their respective targets via Audit & Control. This is done in order to keep a synchronization point at the input file level and enable a simple recovery mechanism.

Dispatching Example

The business requirement is to send all rated events made to a weather forecast service provider named 'Sunny.' The output structure is a special structure requested by the provider.

The Voice Event type has the following dispatching case:

- The qualification criterion checks if the dialed number equals that of the special 'Sunny' number.
- The external record is defined as the format requested by 'Sunny.'

After the call has been rated, Rating checks whether the dialed number of the event equals the Sunny number. If it does, the call is mapped to the Sunny format and written to the Rating output media. The Distributor then distributes the records to the Sunny file.

6. NOTIFICATION MECHANISM

The dispatching of PIs or information based on PIs is also called Notification. Notification is performed while the event is being rated. This enables the sending of notification events on an ongoing basis to internal and external targets.

This chapter details the Notification mechanism.

Purpose of Notification Mechanism

The following are examples of the use of the Notification mechanism:

- Spending limit or budget control uses
A subscriber may request a spending limit program that limits the usage charges in a bill cycle. When a spending limit is reached (the performance indicator accumulators reach the threshold), two types of actions can be activated:
 - Enforcement – Enforcing the spending limit by barring the service that is monitored by the spending limit.
 - Alert – Sending an alert message to the subscriber that the limit has been reached.
- Granting a benefit
A benefit may be awarded to a subscriber when a predefined threshold on usage has been reached. For example, the Notification mechanism may be used to inform subscribers that they are entitled to a gift when the threshold is reached.

In the cases above, a notification is sent to the target system, which in turn implements the actions required.

Product Catalog Definitions

The following are defined in the Product Catalog for all Rating output:

- External record
- Mapping case of the PI to an external record (defined in the context of the PIT)
- Target in the Dispatching Target table

Notification Trigger

The Notification trigger resides in the PIT handlers. The notification extension function is called according to the PIT logic. The functional inputs are the external record and the Dispatching Target ID. The function generates the external record.

For example, given a Spending Limit PIT flow, the notification trigger:

1. Accumulates the total charge amount for each event
2. Checks if the total charge amount is greater than the customer parameter spending limit threshold. If it is, then the notification function is executed and a record is generated for the target system, for example, an SMS machine.

Rating Notification Mechanism

The Rating Notification mechanism is based on the following flow:

1. The Create Notification core extension function generates a notification record in the form of an external record. The function performs the following:
 - a. Receives as input the external record and dispatching target ID.
 - b. Generates the requested external record by activating the relevant mapping case from the current PIT to the given external record.
 - c. Adds the resulting external record to an internal list of notification external records.
2. At the end of event processing, the list of notification external records is delivered to the Distributor for further processing, together with the rated events or external records generated by the current event's Dispatching cases.

Distributor

The Distributor delivers the notification external records to the relevant targets. (See “Distributor” in the Rated Event Dispatching Mechanism chapter for details.)

Appendixes

Appendix A. Native XML

“Native XML” is a generic XML format that Rating uses to describe:

- Input XML structures, such as:
 - Events
 - Customer offers and parameters
- Output information, such as:
 - Usage and PI query APIs
 - Dispatching information (when requested in XML)
 - Rated event and PI extract information (when requested in XML)

This chapter describes the native XML entity data type format, and provides examples of input and output files.

EntityData_t Type

This type describes the native XML entity format. It is a dynamic entity that is composed of a list of attribute value pairs.

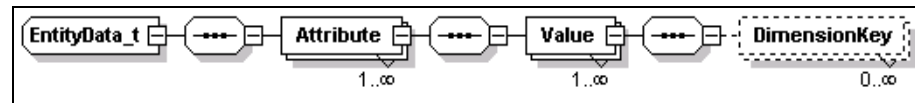


Figure A-1: EntityData_t Type

Element Name	Description	Attribute	Description
<i>EntityData_t</i> type	Describes the structure of the entitydata_t type.	(None)	
<i>Attribute</i> element	Describes a single attribute.	<i>name</i>	The name of the attribute.
		<i>basicType</i>	The type of the attribute. Possible values are: <ul style="list-style-type: none">■ Numeric■ String■ DateTime■ Boolean
		<i>Elementary Type</i>	Elementary type of attribute as defined in the Product Catalog. This attribute is optional.
		<i>uom</i>	Unit of measure for the attribute. This attribute is required for quantity-based attribute types.

Element Name	Description	Attribute	Description
<i>Value</i> element	Describes a single value of an attribute. There may be multiple values if the attribute is complex. In this case, one or more dimension keys may be attached to each value.	<i>value</i>	The value of the attribute.
<i>DimensionKey</i> Element	Describes a single dimension key for a value of a complex attribute.	<i>name</i>	The descriptive name of the dimension key value. This attribute is optional and is used in cases where the dimension is based on a list of values, where each value has a value and a value name pair.
		<i>Value</i>	The value of the dimension key.
		<i>dimensionName</i>	The name of the dimension.

Input File Example

The following is an example of an input file.

```

<CustomerOffers>
  <Attribute name="Subscriber Number">
    <Value value="200000050"/>
  </Attribute>
  <Attribute name="Offers">
    <Value value="14448"/>
    <Value value="19619"/>
  </Attribute>
  <Attribute name="Offers effective date">
    <Value value="2002-11-15 00:33:24"/>
    <Value value="2002-11-15 00:01:23"/>
  </Attribute>
  <Attribute name="Offers expiration date">
    <Value value=""/>
    <Value value=""/>
  </Attribute>
  <Attribute name="Offers Agreement ID">
    <Value value="200000050"/>
    <Value value="200000050"/>
  </Attribute>
  <Attribute name="Offers instance">
    <Value value="1000500"/>
    <Value value="1000210"/>
  </Attribute>
  <Attribute name="Offers Level ID">
    <Value value="0"/>
    <Value value="0"/>
  </Attribute>
</CustomerOffers>
<CustomerOffers>

```

Output File Example

This is a Dispatching example, in which usage information is sent to the MVNO:

```
<EntityData>
  <ExternalRecord type="Wholesale/Retail MVNO voice">
    <Attribute name="Event type" basicType="String">
      <Value value="GSM voice"/>
    </Attribute>
    <Attribute name="Dispatching purpose" basicType="String">
      <Value value=" "/>
    </Attribute>
    <Attribute name="Original event ID" basicType="Numeric">
      <Value value="203619112536000001"/>
    </Attribute>
    <Attribute name="Start time" basicType="DateTime">
      <Value value="2002-11-16 20:40:00"/>
    </Attribute>
    <Attribute name="Original currency charge amount"
      basicType="Numeric">
      <Value value="5"/>
    </Attribute>
    <Attribute name="Original currency" basicType="String">
      <Value value="USD"/>
    </Attribute>
    <Attribute name="Network start time" basicType="DateTime">
      <Value value="2002-11-16 20:40:00"/>
    </Attribute>
    <Attribute name="Provider ID" basicType="String">
      <Value value=""/>
    </Attribute>
    <Attribute name="Calling number" basicType="String">
      <Value value="0011112220036"/>
    </Attribute>
    <Attribute name="Duration" basicType="Numeric" uom="Seconds">
      <Value value="1800"/>
    </Attribute>
    <Attribute name="Call category" basicType="String">
      <Value value="0"/>
    </Attribute>
  </ExternalRecord>
</EntityData>
```


amdocsrating

Index

Access Data Storage Flow	67	Customer Tables	7, 15
Activity Event	9	Cycle Extract Flow	62
Agreement Parameters table	15	Cycle Extract Mode, Extract Process Input ..	60
APIs		Data Distribution, Product Catalog	36
Get PI by ID	71	Dispatch Mechanism	
Get PIs by Customer	68	Dispatching Case	76
Get Rated Event by ID	70	Product Catalog Definitions for	76
Get Rated Events by Customer	69	Required Dispatching Definitions	78
Rating Online Query	68	Dispatched Event, Writing	79
Auxiliary Repository	36	Dispatching	75, 78, 80
Configuration Extract Definition	21	Definitions, Required	78
CUG table	15	Map Event to External Record	78
Customer Extract	5, 11	Qualification	78
Configuration file	21	Write Dispatched Event	79
Customer Tables	15	Dispatching Case	76
Distribute Data	15	Dispatching Target Identifier	76
Modes	11	Example	77
Process Flow	14	External Record Type	76
Results	7	Qualification Criterion	76
Write to Files	16	Distribute Data	15
Customer Extract Configuration File		Distributor	79
File Format Definition	25	Notification Mechanism	82
Format Cursor Definition	25	Dynamic Parameter Definition	29
Format Definitions	24	EntityData_t Type	85
Full Extract Definitions	21	Event	
Incremental Extract Definitions	23	Subscriber Activity	9
Manual Mark Definitions	22	Event Processing	
Provider Extract Definitions	22	Offline Charging Flow	38
Customer Extract Flow	11, 63	Event Processing Flow	
Customer Extract Modes		Online Charging	39
Daemon Submode	13	Event types	37
Full	11	Events	37
Full Submode	11	Core Rating Attributes	40
FullFill Submode	11	Defining New Events	40
History Submode	13	Extracts	59
Incremental	12	Map to External Record	78
ManualMark Submode	12	Mapping	48
Provider Submode	11	Network Usage Events	37
Customer information to Rating area	5	Non-esage Events	39
Customer List Mode, Extract Process Input ..	61	Structure	40
Customer Load		External Record Layout Type	47
Results	7	Extract	
Customer Load	31	Configuration Extract Definition	21
Customer Load		Customer Extract	11
Process flow	31	Customer Extract results	7
Customer Load		Extract Mapping	48
Pre- and Post-activity Events	32	Event	48
Customer Load		Formatting	49
Pre- and Post-activity Events	32	Performance Indicator	48
Customer Load		Extract Process	
Run Modes	33	Number of Parallel Runs and Rated Event	
Customer table	15	Partitioning	64

Rating 6.0 Detailed Interfaces

Extract Process Flow	62	Non-usage Events.....	39
Customer Extract Mode	63	Notification Mechanism	81
Cycle Extract Mode.....	62	Distributor	82
Extract Process Input		Product Catalog Definitions	81
Customer List Mode.....	61	Rating	82
Cycle Extract Mode.....	60	Notification Trigger, Product Catalog	
Extract Process Input Parameters	60	Definitions.....	82
Extracts		Number of Parallel Runs and Rated Event	
Event and Performance Indicator	59	Partitioning.....	64
Process, Flow	62	Offer Catalog.....	35
Processes	59	Online Charging Event Processing Flow	39
Processes, Input Parameters	60	Online Usage Queries	67
File Format Definition	25	Access Data Storage Flow.....	67
File Types		Architecture	67
Subscriber Offers.....	16, 17, 18, 19, 20	Rating APIs	68
Flow		Rating for Customer Management	72
Customer Extract.....	11	Parallel Runs	
Retrieval API.....	71	Number.....	64
Format Cursor Definitions	25	Parameter Definition	
Format Definitions	24	Product Catalog.....	28
Formatting.....	49	Partition File Type.....	20
Formatting Configuration XML File.....	52	Performance Indicator	
Full Extract.....	11	Extracts.....	59
Full Extract Definitions.....	21	Mapping	48
Generic Codes Table.....	72	PI <i>See</i> Performance Indicator	
Generic File Type	19	Pricing Package Catalog.....	35
Get PI by ID API.....	71	Product Catalog	
Get PIs by Customer API.....	68	Parameter Definition	28
Get Rated Event by ID API.....	70	Product Catalog Data Distribution	36
Get Rated Events by Customer API.....	69	Product Catalog Definitions	47
Implementation Repository	35	External Record Layout Type	47
Incremental Extract.....	12	Extract Mapping.....	48
Incremental Extract Definitions	23	Notification Mechanism	81
Interfaces		Notification Trigger.....	82
Customer Management, Rating Online		Rated Event Dispatch Mechanism.....	76
Query APIs	68	Product Catalog information	
Customer Management, Rating Reference		Implementation Repository	35
Tables	72	Product Catalog Information	35
Event and Performance Indicator	59	Auxiliary Repository	36
Online Query Architecture	67	Offer Catalog.....	35
Product Catalog Definitions	47	Pricing Package Catalog.....	35
Rating, Output	45	Product Catalog Subsystems	35
Layout		Provider Extract Definitions.....	22
Formatting Configuration XML File.....	52	Queries, Online Usage	67
Load		Query Output Format Table	72
Customer Load Results	7	Rated Event Dispatch Mechanism	75
Load Process	31	Dispatching.....	78, 80
Manual Mark Definitions	22	Distributor.....	79
Mapping		Product Catalog Definitions for.....	76
Event	48	Product Catalog Definitions for,	
Extract	48	Dispatching Case	76
Performance Indicator	48	Product Catalog Definitions for, Required	
Native XML	85	Dispatching Definitions.....	78
EntityData_t Type	85	Rated Event Partitioning	64
Network Usage Events.....	37	Rater Online Query APIs	

Get PI by ID	71	Subscriber Parameters	
Get PIs by Customer	68	Dynamic Parameter Definition.....	29
Get Rated Event by ID	70	Subscriber Parameters File Type	17
Get Rated Events by Customer	69	Subscriber Parameters Offers File Type	18
Rating Customer Tables.....	7	Subscriber table.....	15
Rating Notification Mechanism	82	Tables	
Rating Online Query APIs	68	Agreement Parameters table.....	15
Rating Output Interfaces	45	CUG table.....	15
Rating Reference Tables		Customer table	15
Generic Codes	72	Service Agreement table	15
Query Output Format	72	Subscriber table	15
Reference Tables		Transaction Broker	
Generic Codes	72	Transaction Configuration File	26
Query Output Format	72	Transaction Configuration File	26
Reference Tables, Rating	72	Types	
Retrieval API Flow	71	Event types.....	37
Service Agreement table	15	Write to Files.....	16
Subscriber Activity Event.....	9	XML	
Subscriber Activity File Type	18	Native	85
Subscriber Offers File Type.....	16		