Oracle® Communications Billing and Revenue Management

Loading Events

Release 12.0
E89247-02
May 2021

Oracle Communications Billing and Revenue Management Loading Events, Release 12.0

E89247-02

Contents

2 Loading Events from External Sources

Preface

This guide describes how to load events from event log files into the Oracle Communications Billing and Revenue Management (BRM) database for rating.

Audience

This document is intended for system administrators and other professionals involved in pricing and rating.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

1

About Rating Events Created by External Sources

This chapter presents an overview of loading events from event log files into the Oracle Communications Billing and Revenue Management (BRM) database for rating.

Table 1-1 lists additional BRM documents that contain information about loading events from event log files:

Table 1-1 Links to Information on Loading Events from Event Log File

Documentation	Topic
About Importing Events from External Sources	Prerequisites for creating an event import template
Online help for the Universal Event (UE) Mapper application (start Developer Center and see UE Mapper online help)	Instructions for creating an event import template
Loading Events from External Sources	Instructions for loading events and handling errors
"Universal Event Loader Utility" utility page	Syntax of utility you use to load events
"pin_uei_deploy" utility page in *BRM Developer's Guide*.	Syntax of utility you use to migrate event import templates from one database to another
"Creating custom bill items" in *BRM Configuring and Running Billing*.	Considerations for using custom bill items

About Importing and Rating Events

BRM can rate service usage in real time. To rate usage in real time, you need a BRM client application such as RADIUS Manager to capture billable events. For some services, it is easier to import and rate events recorded in event log files than to create a BRM client application that captures and rates the events in real time. You can import events from event log files that include data from Web servers and other sources.

UE Importer consists of UE Loader, a command-line event loader that parses the event log file and loads the event into BRM, and UE Mapper, a client GUI application that you use to create event import templates that specify the file format. UE Loader uses the template to parse and load the event log file.

Figure 1-1 shows that:

1. **UE Mapper:**
   a. Reads the event log file when you create an event import template for the log file.
   b. Saves the event import template to the BRM database.

c. Reads the template from the database when you modify the template.

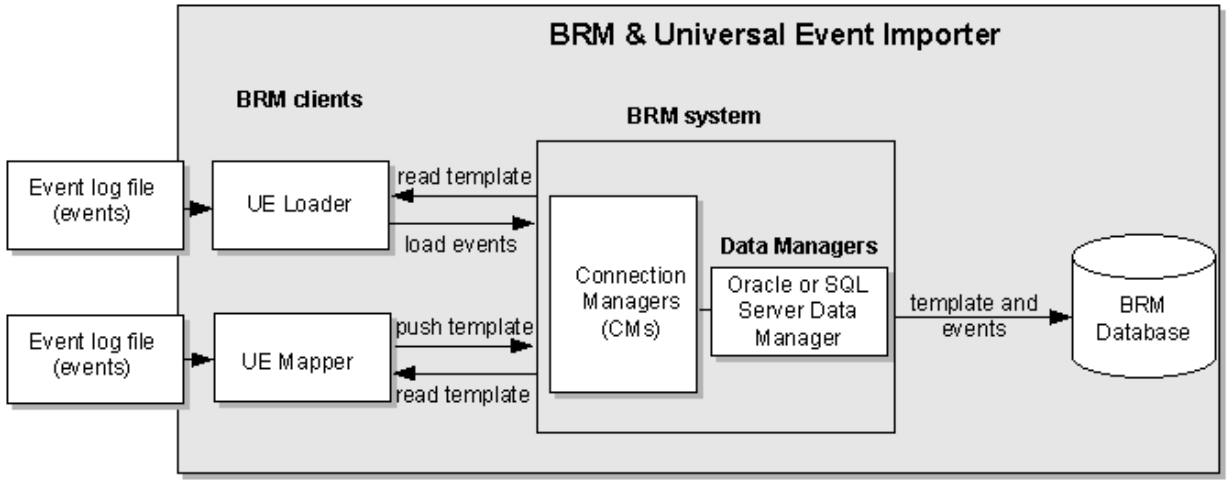
2. **UE Loader:**

   a. Reads the event log file.

   b. Uses the event import template for instructions on loading the events from the log file into the database.

   c. Loads the events into the BRM database.

Figure 1-1 BRM and UE Importer



About Universal Event Mapper

To capture the events you want to import from event log files, use Universal Event (UE) Mapper to create an event import template. You can create multiple templates for your needs. For example, you might need one template for IP telephony data records and another for Internet dialup usage records. You can also create a single template to process both of these record types if they occur in the same event log file. Event import templates are stored in the BRM database in XML format.

For more information on creating event import templates, see "About Importing Events from External Sources". For step-by-step procedures on creating event import templates, see UE Mapper and see the Help.

About Universal Event Loader

To import and rate events from event log files, you use Universal Event (UE) Loader. UE Loader reads the event import template to load data from event log files into BRM as billable events. When the events are loaded, BRM rates the events and updates customer account balances. UE Loader is included in the BRM server software and is located in *BRM_Home***/apps/uel**.

About Setting Up Event Importing

To import and rate events from event log files, you must do the following:

1. If you have not already done so, create a price list that includes the products and rate plans used for rating the events. You might need to define new event and service subclasses.
2. Use UE Mapper to create a template that translates the event log file entries into BRM event fields. UE Loader uses the template when loading events. See "About Importing Events from External Sources".
3. Run UE Loader to import and rate the events. See "Loading Events from External Sources". You can load events manually or set up automatic loading by using Batch Controller. For information on automatic loading, see "Controlling batch operations" in *BRM System Administrator's Guide*.

You might need to create custom opcodes to import events if BRM opcodes cannot handle the type of data you must import. See "About Creating Custom Opcodes".

About Supported Event Log Files

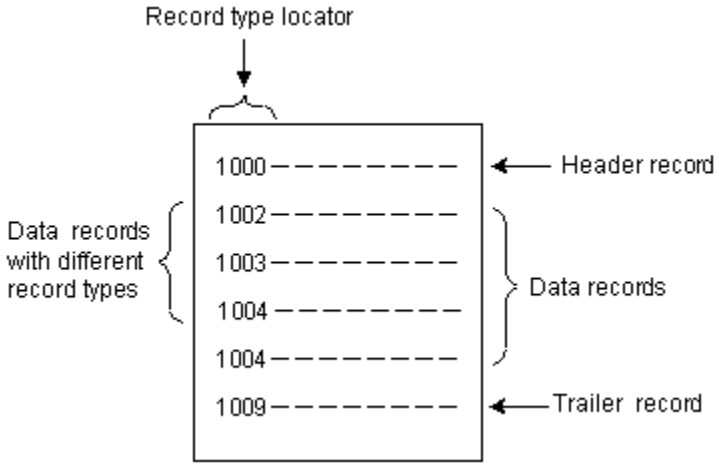
This section describes the data log files you can import (load event data) into the BRM database.

You can import ASCII or binary data from event log files. If your event log file is a binary hex file, you can import records that use a "fixed width" record format. If your event log file is a text file, you can import records that use a "fixed width", "delimited", or "attribute value" record format.

Event Log File Terminology

The event log file can contain a header record (optional), data records, and a trailer record (optional). The header record and the trailer record can be defined by a unique record type locator. The data records can be more than one record type as specified by their record type locators. Figure 1-2 shows a sample event log containing the different record types.

Figure 1-2 Sample Event Log with Records



You specify whether your event log file has a header and trailer record when you create the event import template. If you do not specify that your event log file has a

header and trailer record, UE Loader assumes all records in the event log file are data records.

Each of these three types of records (header, data, and trailer) might be identified by a unique record type. If you have a header and trailer record, and you do not define a record type for each, UE Loader assumes the first record that is imported is the header and the last record is the trailer.

> **Note:**
>
> There can be only one header record and one trailer record in an event log file.

For more information on record types, see "About Loading Event Records Based on Record Type".

Supported File Types

You can import event log files with these file types:

text file type

Consists of data in ASCII text format. The record format for records of this file type must be delimited, attribute value, or fixed width.

binary hex file type

Consists of data as BCD (Binary-Coded Decimal) integers and EBCDIC (Extended Binary-Coded Decimal Interchange Code) strings. The record format for records of this file type must be fixed width.

Supported Record Formats

You can import the following record formats into BRM. All records in a log file must have the same record format.

attribute value

Fields within the records are specified by attribute-value pairs in this format: <name><attribute-value separator><value><delimiter>. For example, **login=bob,** is an attribute-value pair that uses an equal sign as the separator and a comma as the delimiter. Attribute-value pairs can occur in any order in the event log file. Records can be in one line or span multiple lines.

delimited

Records and record fields are delimited by ASCII characters. The delimiter consists of only one character, such as a comma.

fixed width

Fixed width data can be one of the following:

- **Continuous:** Each record has a fixed length
- **Delimited:** The fields have a fixed length

Within their specified length, fields can contain padding characters and can be left or right justified.

> **Note:**
>
> If there are multiple record types, the records may or may not be uniform in length. Record structure can also vary between record types. For example, you can define columns 1 through 8 as the start time for IP telephony record types, and columns 1 through 5 as the start time for Internet dialup record types.

Supported Trailer and Header Records

Typically, there are three kinds of records in an event log file: header records, data records, and trailer records. A file can include one header record, several data records, and one trailer record. Data records can have any number of data record types, such as data records pertaining to email services, dialup IP services, IP telephony services, and so on. You specify record types in UE Mapper by specifying a record type locator, a field that identifies each record's type.

You specify that your event log file has a header and trailer record in the event import template so UE Loader does not load them into the database. The optional header and trailer records may or may not have a record type. If a header and trailer record is specified without a record type, the first record is treated as the header and the last record is treated as the trailer.

> **Note:**
>
> Event log files are not required to have header and trailer records.

Header Records

There can be only one header record in a data log file. The header record must precede all data records and must use the same record format as all other records (delimited, fixed-width, or attribute-value format).

Header records usually contain information relevant to all events in the data records. A header record can also contain information that validates the status of the data log file, such as the number of expected data records in the file. You can set up a header check in UE Mapper to use this data to validate the number of records in the event log file.

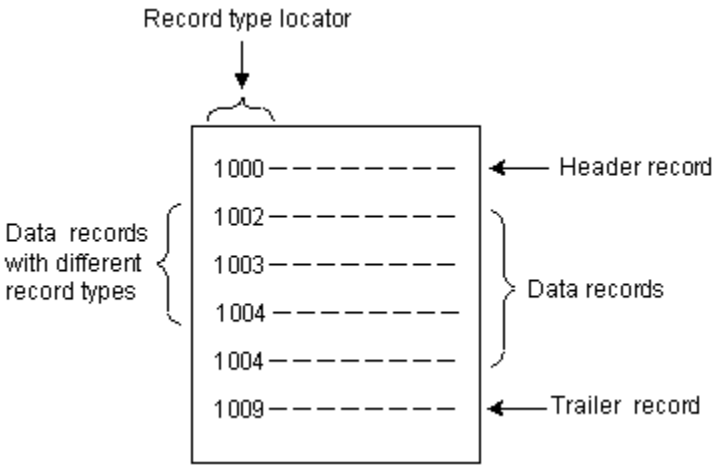
Trailer Records

There can be only one trailer record in an event log file. It must use the same record format as all other records (delimited, fixed-width, or attribute-value format), and it must follow all data records.

Trailer records usually contain information to verify the status of the event log file. If your trailer records include this information, you can set up trailer checks by using UE Mapper to validate the event log file.

Figure 1-3 shows an event log file with a header record, data records of several record types, and a trailer record:

Figure 1-3 Sample Event Log with Records



About Creating Custom Opcodes

If BRM opcodes cannot handle your event log files, you might need to create custom opcodes to load events.

About Custom Opcodes to Map Event Data to Accounts

You might need to create a custom opcode to find the customer accounts associated with the events in your event log files. Usually, the data in the event log file includes login names that existing BRM opcodes can use to find the account for each event. If the event log file does not include login names, you must write a custom opcode that finds the account based on other data in the event log file, such as the customer's IP address. See "Creating Opcodes for Finding Accounts Associated with Event Data".

About Custom Opcodes to Load Event Data

You might need to create a custom opcode to load the events into the BRM database. In many cases, you can use existing BRM opcodes to load event data, but to preprocess the data in the event log file either before rating or before storing the event data in the BRM database, you must create a custom opcode.

For example, if your event log files include a value for downloaded kilobytes, and you need the value represented in bytes, you can create an opcode to preprocess the data before loading the event. Also, if the event log files include area codes and phone numbers in different fields, you may want to concatenate the area code and the phone number in the opcode to store it as a single number in the database. For more information, see "Creating Opcodes for Loading Event Data".

About Loading Event Records Based on Record Type

Data records can have any number of data record types, such as data records pertaining to email services, dialup IP services, IP telephony services, and so on. You specify record types in UE Mapper by specifying a record type locator, a field that identifies each record's type.

One event log file can contain data records of several record types. When you create an event import template for log files with multiple record types, you specify the following items for each record type:

- The BRM event type associated with the data
- The opcode that loads the data
- The opcode that finds the account associated with the data
- The filters that are applied to load or log the data

About Ignoring Record Types When Loading

An event import template specifies which record types are loaded into your BRM database.

You might not want to load all event records from your event log files into BRM. For example, you specify a record type locator so you can use *filters* to prevent unnecessary record types from being loaded into the database. A *record type locator* is a unique field that identifies the different record types.

Using filters can improve loading performance. For example, in many IP telephony CDR files, there are **start accounting** event records and **start stop accounting** event records. The **start stop accounting** event records contain the necessary information to rate IP telephony calls, so it is not necessary to load the **start accounting** event records. Using filters to prevent loading **start accounting** records into the database significantly increases loading performance.

About Using Filters for Loading Events

You can set up filters in the event import template so that events in your event log file that satisfy specified criteria are handled as follows:

- Discarded (not loaded). See "About Discarding Event Records Based on Record Content".
- Logged to a separate file. You can log events that are loaded or discarded to a separate file. See "About Logging Event Records Based on Record Content".

Event filtering occurs after UE Loader does the following:

1. Reads the event log file.
2. Parses the contents of the event log file.

   This task includes verifying whether each event in the log file is associated with a customer account in the BRM database. Thus, account verification is done for some events that might later be filtered out of the event log file.
3. Creates a cache file that contains the results of the parsing.

After creating the cache file, UE Loader applies any filters in the event import template to the events in the cache file, removing events that satisfy the filter-out criteria.

For instructions on setting up filters, start Developer Center and see the UE Mapper Help.

About Discarding Event Records Based on Record Content

Use filters to prevent loading certain event log file records into the database. Records that have particular data field values can be marked and rejected without causing errors in the loading process.

For example, if you have a list of accounts with an inactive account status, you can reject event records that have login fields associated with those accounts.

Another reason for rejecting records based on record content is when you have a minimum amount of usage for which you want to charge users. For example, you can reject records where the dialup session is less than fifteen seconds.

About Logging Event Records Based on Record Content

Use filters to log specific event log file records to a file that you then load into the database. The filter marks records that have particular data field values and logs them to a file.

For example, if you notice accounts with high volume usage, you might want to offer these customers a pricing plan for high volume users. A filter can mark and log the accounts that have a usage value above your filter criteria to create a list of customers to contact. You can also set up a filter to mark and log accounts that show signs of fraudulent activity.

You set up filters when you create the event import template. For instructions, start Developer Center and see the UE Mapper Help.

About Validating the Event Log File

If your event log file includes header and trailer records, you can set up checks to verify that the complete number or size of records is in the log file.

You set up header and trailer checks by using UE Mapper when you create the event import template. For instructions, start Developer Center and see the UE Mapper Help.

About Importing International Log Files

UE Loader supports log files in English, French, German, and Japanese. For instructions on setting up UE Loader for these languages, see "Configuring UE Loader for International Log Files". UE Loader supports UTF8 and Unicode encoding formats.

Importing and Rating Events from External Sources

You can rate service usage by importing usage events from log files, such as Web server log files and IP telephony call detail records (CDR) files, into BRM. This section provides a brief technical overview of how UE Loader imports event data from event

log files into the BRM database for rating. It also explains how to prepare for creating an event import template.

For an overview of importing events, see "About Rating Events Created by External Sources". For instructions on loading data log file event data into the BRM database, see "Loading Events from External Sources".

About Importing Events from External Sources

To import usage events from log files, you create event import templates that specify the usage fields to import and how to import them. You use UE Mapper, a component of Developer Center, to create event import templates. After you create the templates, you use the UE Loader, a command-line utility installed with BRM, to test the templates and load events from your event log files into BRM. UE Loader reads the event import templates in the database for instructions on loading the events.

Overview of Preparing for Loading Events

There are seven steps to prepare for loading external event data into the BRM database:

1. If necessary, create an opcode to find the accounts associated with the event data to load. See "Creating Opcodes for Finding Accounts Associated with Event Data".
2. If necessary, create an opcode to load the event data. See "Creating Opcodes for Loading Event Data".
3. Configure UE Mapper to customize delimiter options. See "Adding Options to UE Mapper Drop-Down Lists".
4. If you use custom events and fields, understand the procedures for importing them. See "Making Custom Fields Available for Mapping".
5. Create an event import template using UE Mapper and save it to a test BRM database. See the UE Mapper online Help.

   The event import template drives the event import process. It specifies:

   - The event data to load from the event log file records.
   - The BRM storable object and storable object fields into which the event data is loaded.
   - The BRM opcode that finds the accounts associated with the event data (optional in some cases).
   - The BRM opcode that loads the event data.
6. Test the event import template by running UE Loader to ensure that event data can be loaded successfully from a sample event log file.
7. Save the event import template to your BRM production database so that UE Loader can load external event data as needed.

How UE Loader Works

UE Loader first loads the event log file records into an object queue. It then reads data from the event import template to construct an flist with the account search criteria and pass the flist to an opcode that finds the account associated with the events. After UE Loader obtains this account, it reads data from the event import template to construct

an flist with the event data and passes the flist to an opcode that loads the event data into the account. If UE Loader is set up to use multiple threads, it can load events into several accounts at once. For more information on multiple threads, see "Setting the Number of Threads for Performance".

Constructing an Flist by Using UE Mapper

For UE Loader to construct an flist, it reads the event import template for the flist instructions you specified. You can specify arrays in the template if the flist should include arrays: for example, to import multiple contacts. You use UE Mapper to populate the fields of an flist with the event data to import into BRM.

To see a valid input flist that was created to load event data from the log file into a specific BRM event object:

1. Load the sample event import template (*BRM_Home*/**apps/sample_handler/sample_template.xml**) into the BRM database:

   ```
   pin_uei_deploy -t SampleTemplate -c -i sample_template.xml
   ```

2. Create a sample event log file named **SampleEvent** in a text editor and add the following contents:

   ```
   Field1Field2Field3Field4Field5
   v01/jun/2003:01:04:00 am PST01/jun/2003:01:07:00 am PST0812
   v01/jun/2004:01:04:00 am PST01/jun/2004:01:07:00 am PST2013
   v01/jun/2005:01:04:00 am PST01/jun/2005:01:07:00 am PST3015
   Field1Field2Field3Field4Field5
   ```

3. Restart Developer Center.
4. Open UE Mapper in Developer Center.
5. Choose **Modify Template** from the **File** menu.
6. In the Modify Template dialog box, select **SampleTemplate** from the **Templates** list and click **Modify**.
7. In the Open Event Log File dialog box, specify the **SampleEvent** event log file and click **Open**.

Creating an Event Import Template

Before you can import events, you use UE Mapper to create an event import template. This section describes tasks you might need to do before you can create the template.

UE Mapper is a Developer Center application. UE Mapper includes detailed help for creating event import templates.

Before You Begin

To create an event import template:

- You should be familiar with the contents of the event log file you are importing, including:
  - What each record and field in the event log file means. Each record corresponds to an event in BRM, and each piece of data corresponds to a field in an event.

  - How the event log file is formatted, including which character signifies the end of a record and which character separates data fields in a record. For example, records might be separated by line ends, fields might be separated by commas, or records might contain continuous data in a fixed-width format.
- You must understand BRM events, in particular, which type of BRM events are associated with the log file events you are loading.
- You must understand BRM opcodes, including required flist fields and the data types of those fields.

Before you create an event import template, you might need to perform one or more of the following tasks:

- Preparing Event Log File Data
- Creating Opcodes for Finding Accounts Associated with Event Data
- Creating Opcodes for Loading Event Data
- Making Custom Fields Available for Mapping
- Adding Options to UE Mapper Drop-Down Lists
- Migrating Event Import Templates from One BRM Database to Another

Preparing Event Log File Data

In some cases, you must preprocess the records in your event log file before you can use the file to create an event import template. You might need to inspect the data in the input records for errors and manipulate the event data before it can be imported into BRM.

Creating Opcodes for Finding Accounts Associated with Event Data

The BRM opcode commonly used to find customer accounts associated with event data is PCM_OP_ACT_FIND. This opcode uses the customer login name in the event record to search the account table in the database associated with the event. If your event records do not contain the customer login name, you must create a custom opcode that uses another field to find the accounts.

> **Note:**
>
> The custom opcode must return the account Portal object ID (POID). It can also return the service, but it is not required to.

For example, if you have IP telephony call detail records (CDRs) that include the call origination number or call ID number of the user instead of the login name, you could create a custom opcode to map the call ID in the event to the account associated with that call ID.

Creating Opcodes for Loading Event Data

In some cases, you must create custom opcodes to load event data.

The BRM opcodes that are commonly used to load data from event log files into BRM are PCM_OP_ACT_USAGE and PCM_OP_ACT_LOAD_SESSION. If you use a method for rating events that is more complex than these opcodes can handle (the data you need to load is not in the format these opcodes expect), you cannot map data to them. You must create a custom opcode that can load your event data.

For example, when importing IP telephony call detail records (CDRs), you probably want to perform duplicate session checking and handle start and stop accounting events. PCM_OP_ACT_USAGE and PCM_OP_ACT_LOAD_SESSION do not handle that data, so you create custom opcodes to handle these transactions. For other examples for creating a custom opcode to load event data, see "About Custom Opcodes to Load Event Data".

Making Custom Fields Available for Mapping

If you create custom BRM fields, you must make these fields available to UE Mapper before they can be displayed for mapping. You use Storable Class Editor to create the fields. For instructions about making custom fields available to Java applications such as UE Mapper, see *BRM Developer's Guide*.

Adding Options to UE Mapper Drop-Down Lists

You might need to add items, such as delimiters, to some drop-down lists in the **Data Definition** tab of UE Mapper. The drop-down lists include popular options, but you can edit the lists in the **UEMapper.properties** file to include other options you need. Follow the instructions in the **UEMapper.properties** file in **/opt/portal/***release***/DevCenter/UEMapper.properties**.

Migrating Event Import Templates from One BRM Database to Another

Use the **pin_uei_deploy** utility to migrate an event import template from one database to another. For example, use the utility when you move a template from a test database to a production database. Using the **pin_uei_deploy** utility, you read an event import template from one database, save it as an output file on a local system, and then load it into another database.

> **Note:**
>
> Do *not* modify the event import template when it is saved as a file on the local system. To modify an event import template, use UE Mapper in Developer Center.

To migrate an event import template from one database to another:

1. Ensure that BRM is running on both servers.
2. Ensure that you have a **pin.conf** file in the same directory as **pin_uei_deploy**.
3. Ensure that you have write privileges for the directory where you intend to save the output file.
4. Check the output directory to ensure that you do not overwrite output files you want to save. The **pin_uei_deploy** utility overwrites an output file of the same name without displaying a warning.

5. In the **pin_uei_deploy** utility **pin.conf** file, set the connection information to point to the database you are migrating the event import template *from*.
6. List all the event import templates stored on that database.

   ```
   pin_uei_deploy -l
   ```

   Note the name of the template you are migrating.
7. Download the template from the database to a local file. You can use any name for the output file. The output file is saved to the current directory unless you specify a path.

   ```
   pin_uei_deploy -t template_name -r -o output_file_name
   ```
8. In the **pin_uei_deploy** utility **pin.conf** file, set the connection information to point to the database you are migrating the event import template *to*.
9. List all the event import templates stored in that database.

   ```
   pin_uei_deploy -l
   ```

   If the name of the template you are migrating exists in that database, ensure that you want to delete or overwrite the existing template as described in the next step.
10. Load the template using these options:

    If the name of the template you are migrating does *not* already exist on the database, load the template using the *create* (**-c**) mode:

    ```
    pin_uei_deploy -t template_name -c -i output_file_name
    ```

    If the name of the template you are migrating already exists in the database, do one of the following:

    - Delete the existing template in the database before loading using the **-c** mode:

      ```
      pin_uei_deploy -t template_name -d
      ```
    - Load the template using the *modify* (**-m**) mode of operation to overwrite the existing template:

      ```
      pin_uei_deploy -t template_name -m -i output_file_name
      ```
11. Verify that the template has been loaded by listing all templates in the database:

    ```
    pin_uei_deploy -l
    ```

2
Loading Events from External Sources

This chapter describes how to load event data from event log files into the Oracle Communications Billing and Revenue Management (BRM) database to be rated. It also provides troubleshooting information for loading events.

For general information about importing events, see "About Rating Events Created by External Sources".

About Loading Events

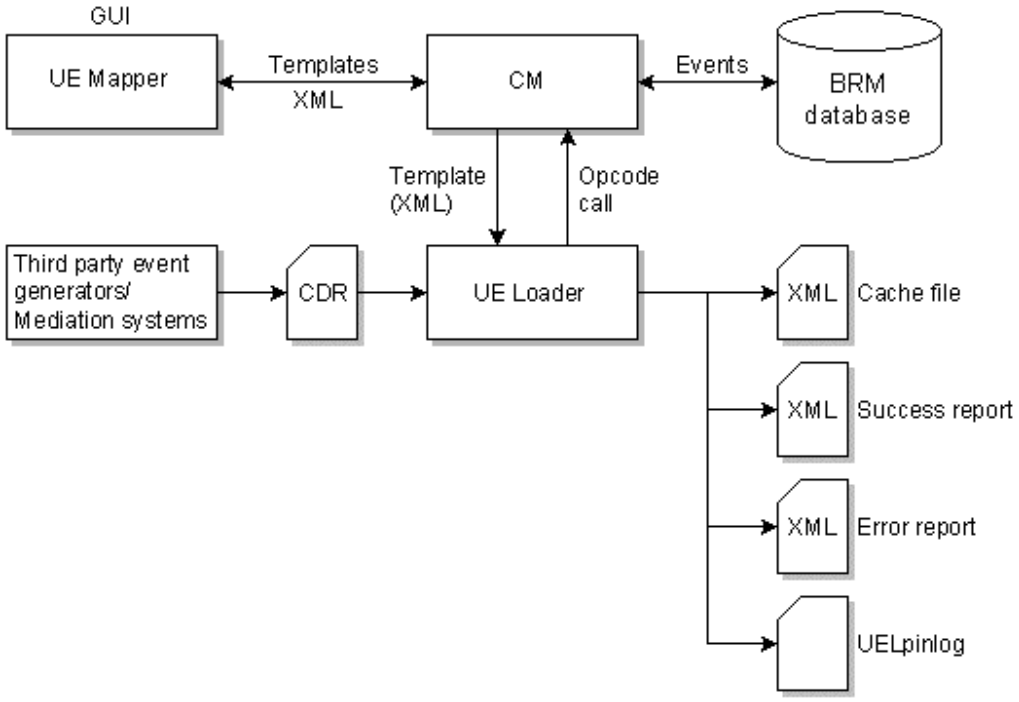
You load data from an event log file into the BRM database by creating an event import template with UE Mapper. You then load the event data manually by using UE Loader or automatically by using Batch Controller and a batch handler.

UE Loader is a command-line utility that reads data about service usage from the event log file, reads the event import template for instructions on loading the data, such as which opcodes to call to load the data, and loads the data into the BRM database. When the events are loaded, BRM rates the events and changes customer account balances. You can set Batch Controller and create a UE Loader batch handler to run UE Loader automatically (to load event data on a regular basis). For information about Batch Controller, see "Controlling batch operations" in *BRM System Administrator's Guide*.

For information on creating an event import template and a technical overview of the event import process, see "About Rating Events Created by External Sources".

Figure 2-1 shows how UE Mapper and UE Loader work to load events into the BRM database:

Figure 2-1 Loading Events into BRM Using UE Mapper and UE Loader



Configuring UE Loader

To configure UE Loader, edit the UE Loader properties file (*BRM_Home***/apps/uel/Infranet.properties**). Follow these procedures:

- Configuring Connection Information for UE Loader
- Specifying the Event Log File Location
- Specifying Whether to Log Duplicate Event Log Files
- Setting the Number of Threads for Performance
- Setting Memory Options
- Specifying How to Load Events
- Configuring Log File Locations for UE Loader
- Specifying the Maximum Load Errors to Process before Quitting
- Specifying the Directory to Store Logged Event Records

Configuring Connection Information for UE Loader

To specify connection information:

1. Open UE Loader properties file (*BRM_Home***/apps/uel/Infranet.properties**).
2. Follow the instructions in the file to set the standard connection information, such as the Connection Manager (CM) name and port number.
3. Save and close the file.

Enabling UE Loader to Convert Flists to Strings

> **Note:**
>
> This is a mandatory configuration.

Ensure that UE Loader is configured to convert flists to strings:

1. Open UE Loader properties file (*BRM_Home*/**apps/uel/Infranet.properties**).
2. Ensure that the **infranet.flist.tostring.enable** entry is set to **true**:

   ```
   infranet.flist.tostring.enable = true
   ```

   If this entry is not present or is set to **false**, UE Loader might load records into the wrong accounts.

Specifying the Event Log File Location

When you run UE Loader, you specify only the name of the event log file, not the file location. You specify the file location in the **Infranet.properties** file.

For information about UE Loader log files, see "Troubleshooting Event Loading".

1. Open the properties file (*BRM_Home*/**apps/uel/Infranet.properties**).
2. Enter the directory name in the **infranet.uel.event_log_file_location** entry. This is the directory that stores event log files for loading.

   When specifying the path, always use forward slashes (/), and type a slash after the last directory name.

   ```
   infranet.uel.event_log_file_location=/var/portal/7.3.1/apps/uel/
   ```
3. Save and close the file.

Specifying Whether to Log Duplicate Event Log Files

By default, UE Loader does not notify you when a duplicate event log file is loaded into the database.

You can configure UE Loader to log an error when a duplicate event log file is loaded by using the **infranet.uel.duplicate_check** entry. When this entry is set to **True**, UE Loader creates a **DuplicateFiles.txt** file and adds the following for each duplicate file:

- The duplicate event log file's name
- The timestamp of when the duplicate event log file was processed

To log an error when duplicate event log files are loaded into the database:

1. Open UE Loader properties file (*BRM_Home*/**apps/uel/Infranet.properties**).
2. Change the **infranet.uel.duplicate_check** entry to **True**:

   ```
   parameter infranet.uel.duplicate_check=True
   ```

3. Save and close the file.

Setting the Number of Threads for Performance

You can improve UE Loader performance by changing the maximum number of threads. Running multiple threads loads multiple events simultaneously, which gives faster performance but puts a greater load on the BRM system. The number of threads that you specify depends on your BRM system configuration, for example, the number and speed of CPUs.

> **Note:**
>
> The number of threads used when processing also depends on the event import template that you use to load the events. For example, if you load events based on their occurrence in the event log file, they are processed one at a time, so using multiple threads does not improve performance. For more information, see "Specifying How to Load Events".

1. Open the UE Loader properties file (*BRM_Home***/apps/uel/Infranet.properties**).
2. Change the **infranet.uel.max_threads** entry, for example:

   ```
   infranet.uel.max_threads=5
   ```

3. Save and close the file.

> **Note:**
>
> You can run multiple UE Loaders to improve performance. Each UE Loader can run multiple threads. If you have only one CM and one DM, however, there is a limit to how much you can improve performance by using multiple UE Loaders.

Setting Memory Options

Avoid memory errors with these guidelines:

- UE Loader can process large log files, such as files with 200,000 records. The memory required to load events depends on the number of records in your event log file.
- If you have a large number of records and UE Loader reports an out-of-memory error (error code **-2**), change the **infranet.uel.queue_size** entry in the **Infranet.properties** file. Setting this entry causes UE Loader to read records into memory in sizable chunks rather than reading all records into memory at once.

> **Note:**
>
> If the **infranet.uel.queue_size** entry in the **Infranet.properties** file is set to **0**, all records are read into memory at once.

- You can configure the memory available to UE Loader by modifying the **-mx** setting in the **uel** script.

Specifying How to Load Events

You can load events based on their occurrence in the event log file, or you can load them based on their account identifier. Specify the loading method when you create the event import template by using the **Group events by account identifier** option in UE Mapper. For more information, start Developer Center and see the UE Mapper Help.

> **Note:**
>
> If a custom bill item is rated cumulatively, UE Loader must group events by account identification for loading. For more information, see UE Mapper Help.

Configuring Log File Locations for UE Loader

To specify locations of log files:

1. Open the UE Loader properties file (*BRM_Home*/**apps/uel/Infranet.properties**).
2. Enter directory names in the following entries:
   - To set the location of the error file that records errors found in event loading mode, use the **infranet.uel.load_error_file_location** entry.
   - To set the location for the record of successfully loaded events, use the **infranet.uel.load_success_file_location** entry.
   - To set the location for the standard BRM client error log file, use the **infranet.uel.error_log_file_location** entry.
   - To set the location for the file that records events you filter by event field values, use the **infranet.uel.filter_log_file_location** entry.

   When specifying the path, always use forward slashes (/), and type a slash after the last directory name.

   ```
   infranet.uel.error_log_file_location=/var/portal/7.3.1/apps/uel/
   ```
3. Save and close the file.

Specifying the Maximum Load Errors to Process before Quitting

The maximum number of errors applies to errors generated from all threads, not errors per thread.

1. Open the properties file (*BRM_Home*/**apps/uel/Infranet.properties**).
2. Set the **infranet.uel.max_errors** entry, for example:

   ```
   infranet.uel.max_errors=100
   ```

3. Save and close the file.

Specifying the Directory to Store Logged Event Records

You can set up filters so that particular records are logged to a file, whether you load them or not. See "About Discarding Event Records Based on Record Content" and "About Logging Event Records Based on Record Content".

To specify the directory to store the files that log filtered records:

1. Open the UE Loader properties file (*BRM_Home*/**apps/uel/Infranet.properties**).
2. Set the **infranet.uel.filter_log_file_location** entry.
3. Save and close the file.

Specifying the Default Date/Time Format for Timestamps

You can specify a timestamp format to support loading events from different countries that format dates and times in different ways. The timestamp format must match the date/time format in the event log file that you are importing. The timestamp format is used by fields with the data type **PIN_FLDT_TSTAMP**.

You can specify the date/time format for each timestamp field (fields with the data type **PIN_FLDT_TSTAMP**) in your event log file when you create the event import template. This is necessary if you have records that use different timestamp formats in the same event log file.

If you do not specify the date/time format for a record on a per field basis when you create the event import template, UE Loader uses the date/time format specified in its **Infranet.properties** file as the default format.

To set the default date/time format:

1. Determine the date format by looking at the event log file you are importing.
2. Open the properties file (*BRM_Home*/**apps/uel/Infranet.properties**).
3. Change the **infranet.uel.date_pattern** entry to match the date format in the event log file you are importing. For example. if your records use a time format specified in a.m./p.m. hours (01 - 12), you specify hours by using a lowercase **h** and including the **a** parameter to indicate that the time is in a.m./p.m. format:

   ```
   infranet.uel.date_pattern=dd/MMM/yyyy:hh:mm:ss a zzzz
   ```

   See "Specifying Time Format Syntax".
4. Save and close the file.

Specifying Time Format Symbols

Table 2-1 shows examples of date formats:

Table 2-1 Example Date/Time Formats

Date in Event Log File	Date/time Format in Infranet.properties File
26/October/2005:10:55:49 Pacific Standard Time	dd/MMM/yyyy:hh:mm:ss a zzzz
12:08 PM	h:mm a
12 o'clock PM, Pacific Daylight Time	hh 'o''clock' a, zzzz
2005.07.10 AD at 15:08:56 PDT	yyyy.MM.dd G 'at' hh:mm:ss z
Wed, July 10, '05	EEE, MMM d, ''yy
0:00 PM, PST	K:mm a, z
2005.July.10 AD 12:08 PM	yyyyy.MMMMM.dd GGG hh:mm aaa

Specifying Time Format Syntax

Each element in the time format syntax is represented by a character. For example, the year is represented by the "y" character. You control how each element is matched by the number of characters you use. For example, with the year 2001:

- yy matches 01
- yyyy matches 2001

There are also different symbol types, such as Text and Number. The number of characters has a different meaning for different symbol types as shown in Table 2-2:

Table 2-2 Time Format Syntax

Symbol Type	Rules for Number of Characters
Text	4 characters or more uses the full word. Fewer than 4 characters presents an abbreviation. For example, for E (day of the week): • E = Wed • EEEE = Wednesday
Number	Any number of characters uses the minimum required number of digits, except for years. With years, 2 characters uses the short version, for example: yy = 99
Text and number	3 characters or more uses text, 1 or 2 characters uses numbers. For example: • MMM = July • MM = 07

In addition, you can use an escape character to match text as shown in Table 2-3:

Table 2-3 Use of Escape Character to Match Text

Log File	Date/Time Symbols
July 2 at 12:01	MMM d 'at' hh:mm

To match a single quotation mark, use double-single quotation marks as a literal, as shown in Table 2-4:

Table 2-4 Use of Quotation Marks to Match Text

Log File	Date/Time Symbols
July 2 "99	MMM d "yy

Any characters outside the range of a-z and A-Z, such as colon (:), semicolon (;), and at symbol (@), are treated as quoted text, even if they are not enclosed in single quotation marks.

> **Note:**
>
> Characters within the ranges of a-z and A-Z that are *not* used as symbols, such as C, and are not enclosed in quotes, cause an error.

Allowing for Leniency in Time References

When a CDR contains a date value that uses a +1300 time reference, the system cannot read the date properly and returns an error. For example, the date entry might look like this: 20061202000000+1300.

To enable the system to process CDRs that use the +1300 time zone reference, set the **Infranet.uel.lenient** value to **true** in the UE Loader properties file (*BRM_Home***/apps/uel/Infranet.properties**) The default value is **false**:

```
Infranet.uel.lenient = true
```

Configuring UE Loader for International Log Files

This section provides instructions for configuring UE Loader to use international log files.

Configuring the UE Loader Infranet.properties File

UE Loader supports English, French, German, and Japanese log files. To support these languages, set the following parameters in the UE Loader **Infranet.properties** file (*BRM_Home***/apps/uel/Infranet.properties**) shown in Table 2-5:

Table 2-5 Parameters in UE Loader File

Infranet.properties File Entry	Description
infranet.uel.event_log_file_lang_code	Specify the language code. See "Language, Country, and Encoding Codes".
infranet.uel.event_log_file_country_code	Specify the country code. UE Loader supports all country codes that apply to each language (English, French, German, and Japanese).

Table 2-5 (Cont.) Parameters in UE Loader File

Infranet.properties File Entry	Description
infranet.uel.event_log_file_variant_code	Specify the variant code.
infranet.uel.event_log_file_encoding	Specify the encoding code. See "Language, Country, and Encoding Codes".
infranet.uel.date_pattern	Specify the date pattern for the locale. See "Specifying the Default Date/Time Format for Timestamps".

Language, Country, and Encoding Codes

Listed below are the language, country, and encoding codes for languages supported by each operating system.

> **Note:**
>
> UE Loader supports all country codes for each language.

NOT_SUPPORTED lists the system encoding support for English.

NOT_SUPPORTED

Operating System	Language Code	Country Codes	Encoding Code
Solaris	en	AU, CA, IE, NZ, ZA, GB, and US	8859-1
Linux	en	AU, CA, IE, NZ, ZA, GB, and US	None
HP-UX IA64	en	AU, CA, IE, NZ, ZA, GB, and US	ISO81
AIX	en	AU, CA, IE, NZ, ZA, GB, and US	ISO8859-1

Table 2-7 lists the system encoding for Japanese.

Table 2-7 System Encoding Support for Japanese

Operating System	Language Code	Country Codes	Encoding Code
Solaris	ja	JP	SJIS
Linux	ja	JP	None
HP-UX IA64	ja	JP	sjis
AIX	ja	JP	IBM-943

Table 2-8 lists the system encoding support for German.

Table 2-8 System Encoding Support for German

Operating System	Language Code	Country Codes	Encoding Code
Solaris	de	AT, DE, LU, and CH	8859-1
Linux	de	AT, DE, LU, and CH	None
HP-UX IA64	de	AT, DE, LU, and CH	ISO81
AIX	de	AT, DE, LU, and CH	ISO8859-1

Table 2-9 lists the system encoding for French.

Table 2-9 System Encoding Support for French

Operating System	Language Code	Country Codes	Encoding Code
Solaris	fr	BE, CA, FR, LU, and CH	8859-1
Linux	fr	BE, CA, FR, LU, and CH	None
HP-UX IA64	fr	BE, CA, FR, LU, and CH	ISO81
AIX	fr	BE, CA, FR, LU, and CH	ISO8859-1

Loading Events

You can load events manually by running UE Loader from a command line or you can schedule UE Loader to run automatically. See:

- Loading Events Manually
- Loading Events Automatically

About the Order of Events

If any event balance impacts are configured to start immediately and never expire, consider the order of events when you load them.

When an event is loaded, BRM checks the validity periods of the account's resource sub-balances. The sub-balance that has a start time earlier than the event's start time is the one that receives the impact. If there is no sub-balance that has an earlier start time, BRM creates a new sub-balance, sets its start time to the event's start time, and impacts that sub-balance. This can cause the creation of multiple sub-balances for the same resource when events are loaded out of order. Having multiple sub-balances for the same resource can impact performance.

To prevent the creation of multiple account sub-balances for resources that start immediately and never expire, be sure to sort the events in ascending order based on their timestamp before running UE Loader.

Loading Events Manually

Use UE Loader to manually load events from data log files into the BRM database. UE Loader is a command-line utility installed with BRM, and its executable file is

in *BRM_Home*/**apps/uel**. For information on UE Loader command-line syntax, see "Universal Event Loader Utility".

For information on loading events from data log files automatically, see "Loading Events Automatically".

Before you begin:

- You must configure UE Loader before you can load events. See "Configuring UE Loader".
- To avoid memory errors, follow the guidelines in "Setting Memory Options".
- An event import template is required to load event data. You must know the exact name of the event import template that was created for the event log file you are loading.
- Before loading events, you must specify the date/time format used in the event log file. You can specify the date/time format for all records in your log file by editing the UE Loader **Infranet.properties** file. See "Specifying the Default Date/Time Format for Timestamps". The date/time format for each record can also be specified on a per field basis when the event import template is created. If the date/time format is not specified on a per field basis, the date/time format specified in the **Infranet.properties** file is used.
- Before loading events, run the UE Loader program on a test database to test the event import template and any custom opcodes that were created for loading your log file events.

To load events manually, run UE Loader by using this syntax:

```
uel -t template_name [-v] [-m parse|load] [-test] log_file_name
```

The optional **-v** verbose parameter causes UE Loader to output error messages or debug messages on the console according to the debug level you set in the **Infranet.properties** file.

The **-m** and **-test** modes are used more during testing of event loading. When you are loading events into a production database, you do not need to use these options.

For example, if the template name is *WebServer* and the event log file name is *webserver.log*, use this command to load events:

```
uel -t WebServer webserver.log
```

Notes:

- Do not enter the path of the event log file. You specify the path in the UE Loader **Infranet.properties** file. See "Specifying the Event Log File Location".
- The log file can use any file extension or no extension.

For information on UE Loader command-line syntax, see "Universal Event Loader Utility".

Loading Events Automatically

You can load event log files automatically and manually. Some BRM service integration components provide a batch handler that executes UE Loader to process event log files automatically, while BRM system software provides Batch Controller that controls the handler. You can also write a custom handler. You can load files

of event records automatically on a periodic basis by configuring UE Loader, Batch Controller, and a handler.

To set up automatic event loading:

1. Configure UE Loader. See "Configuring UE Loader".
2. Set up a handler to launch UE Loader.

   See the documentation of the service integration component you use for instructions on setting up the handler for that component.
3. Set up Batch Controller to do these tasks:
   a. Scan a configured directory for record files from an external source.
   b. Move the files to the appropriate handler for processing.

   See also "Controlling Batch Operations" in *BRM System Administrator's Guide*.

Troubleshooting Event Loading

This section explains UE Loader log files and error codes, and provides tips on troubleshooting event loading failures.

Understanding UE Loader Log Files

UE Loader records errors and successes in the appropriate log files in *BRM_Home*/**apps/uel**. Table 2-10 describes the UE Loader log files.

Table 2-10 UE Loader Log Files

Type of File	Log File Name	Description
Cache file	*event_log_file_name*_**cache.xml**	The cache file is an intermediate file generated by UE Loader after it parses the event log file and before it loads events into the database. UE Loader stores the results of event log parsing in the cache file, which includes: • The event records • The customer accounts associated with the events • A record number for each event record The record number appears in the error log so you know which record to fix in the cache file if there are errors during event loading.
Filter log	*event_log_file_name*_**filtered.xml**	The filter log file is generated when events are loaded. It lists records you have specified to archive. Records are archived based on filters you set up in the event import template. See "About Logging Event Records Based on Record Content". **Note:** The event import template specifies whether filtered records are loaded into the BRM database.

Table 2-10 (Cont.) UE Loader Log Files

Type of File	Log File Name	Description
Error log	*event_log_file_name*_**lerr.xml**	The error log file is generated when events are loaded. It reports any loading errors and specifies the records in the event log file that caused the errors.
Success log	*event_log_file_name*_**lsucc.xml**	The success log file is generated when events are loaded. It lists all records that load into the database successfully.
Application log	**uel.pinlog**	The standard application log file contains complete descriptions of the errors generated in the other log files. To log all errors, set the log level to **3**.

When errors occur, you typically check the log files in this order:

1. UE Loader pinlog
2. Error file
3. Success file
4. Cache file

   Check the cache file for NULL account attributes if an account could not be found during event loading.
5. Filter file

   Check the filter file if you have set up filters for event loading to see that events were loaded and logged according to the filter.

> **Note:**
>
> The name that UE Loader generates for its log files includes the extension of the event log file. For example, when the event log file is named **CDR3.txt**, the cache, filter, error, and success files are called:
>
> - **CDR3.txt_cache.xml**
> - **CDR3.txt_filtered.xml**
> - **CDR3.txt_lerr.xml**
> - **CDR3.txt_lsucc.xml**.

Understanding UE Loader Error Codes

Table 2-11 describes the UE Loader error codes that appear in the **uel.pinlog**, *event_log_file_name*_**lerr.xml** file, and *event_log_file_name*_**cache.xml** file:

Table 2-11 UE Loader Error Codes

Error Code	Brief Description	Error Description	Possible Required Actions
0	NO_ERROR	All records loaded successfully.	No action needed.
-1	FAILED_HEADER_TRAILER_CHECK	The header record or trailer record check failed.	• If the content does not match the header and trailer check, check for a file content problem. • Use UE Mapper to ensure that the header or trailer check is set up correctly in the event import template.
-2	NOT_ENOUGH_MEMORY	UE Loader ran out of memory.	• Reconfigure the memory available to UE Loader by modifying the **-mx** setting in **uel** script. • Reconfigure the queue size in the **Infranet.properties** file. For details, see "Setting Memory Options".
-3	OUT_OF_DISK_SPACE	UE Loader ran out of disk space while creating log files. For example, there was insufficient disk space in the specified directory to store UE Loader success, error, or filter log files.	Specify a location for log files that contains more disk space. See "Configuring Log File Locations for UE Loader".
-4	TEMPLATE_ERROR	UE Loader detected a problem with the event import template. For example, you used the wrong version of a template (a template that was created with an earlier version of UE Mapper).	To convert an event import template to the latest version, migrate the template in UE Mapper. See the UE Mapper Help for instructions.
-100	BAD_INF_CONNECTION	The BRM system is not available. For example, the CM is down.	• Start BRM if it is down. • Check your connection settings, such as the port number. See "Configuring Connection Information for UE Loader".
1	FIELD_PARSE_ERROR	An error occurred when parsing the fields in the event log file. For example, UE Loader detected no matching string delimiter when parsing the event record data.	1. Open the cache file. 2. Find the record with the error code **1**. 3. Correct the error in the record. 4. Save and close the cache file. 5. Reload the events by using the **-m load** option. See "Reloading Events That Failed to Load".

Table 2-11 (Cont.) UE Loader Error Codes

Error Code	Brief Description	Error Description	Possible Required Actions
2	ACCT_FORMAT_ERROR	When UE Loader constructed the input flist to pass to the opcode that finds the account, it detected an error in the data format. Possible causes: • Incorrect field specification. For example, the login is specified as fields 1 through 5 when the login should be specified as fields 1 through 7. • Incorrect setup of the input flist mapping for finding accounts associated with events.	• If the field specification is incorrect, open the event log file and correct the record. • If the input flist mapping for finding accounts is incorrect, use UE Mapper to correct the mapping in the event import template.
3	LOAD_FORMAT_ERROR	When UE Loader constructed the input flist to pass to the opcode that loads the event data into BRM, it detected an error in the data format. Possible causes are similar to error code **2**.	1. Open the cache file. 2. Find the record with error code **3**. 3. Correct the error in the record. 4. Save and close the cache file. 5. Reload the events by using the **-m load** option. See "Reloading Events That Failed to Load".
4	PREV_PARSE_ERROR	When loading events, a parse error occurred. **Note:** You do not receive this error if you choose to continue loading events when parse errors occur. To configure this, set the **infranet.uel.load_on_parse_errors** entry in the UE Loader **Infranet.properties** file. You can also configure a maximum number of parse errors before UE Loader aborts.	Correct the record in the event log file. **Note:** This type of error cannot be corrected in the cache file (for reloading with the **-m load** option) because the order of events is incorrect. The error must be corrected in the event log file.
5	TOO_MANY_LOAD_ERRORS	The number of load errors exceeded the number of errors you allow. This error is generated when a number of load format errors (code **3**) or load opcode errors (code **101**) occur. **Note:** To set the maximum number of errors, use the **infranet.uel.max_load_errors** entry in the UE Loader **Infranet.properties** file.	See the possible actions for error codes **3** and **101**.
10	COMMAND_LINE_ARG_ERROR	The command line syntax you used to run UE Loader is incorrect.	See the "Universal Event Loader Utility" utility page for the correct syntax.

Table 2-11 (Cont.) UE Loader Error Codes

Error Code	Brief Description	Error Description	Possible Required Actions
11	FILE_NOT_FOUND_ERROR	The event log file you specified in the UE Loader command line cannot be found.	• Ensure that the location of the event log file is specified correctly in the **Infranet.properties** file. • Use the correct name for the event log file in the UE Loader command.
100	ACCT_NOT_FOUND	The opcode that finds accounts associated with event data could not find an account. Possible causes: • The account does not exist in BRM. • The account exists in BRM, but the login or service specified is incorrect. • The input flist mapping for finding accounts associated with event data is incorrect.	• Correct the record in the event log file. • Use UE Mapper to ensure that the correct service type associated with the event is specified in the event import template. • If the input flist mapping for finding accounts is incorrect, use UE Mapper to correct the mapping in the event import template.
101	LOAD_OPCODE_ERROR	The opcode that loads event data into BRM could not load the data.	• Open the cache file and correct the record. • If the input flist mapping for loading event data is incorrect, use UE Mapper to correct the mapping in the event import template.
111	The **template61.dtd** file is missing.	The default UE Loader template file (**template61.dtd**) is not present in the directory from which the UE Loader utility is run.	Copy the template file to the directory from which you run UE Loader. The default template, **template61.dtd**, is in *BRM_Home***/apps/uel**.

Troubleshooting Tips

Some common causes of failure to load event records into the BRM database:

- Incorrect connection settings in your UE Loader configuration file.

  See the UE Loader properties file (*BRM_Home***/apps/uel/Infranet.properties**) for instructions on setting each entry.

- Incorrect format of the event data in your event log files.

  UE Loader expects the records in your event log file to use the format defined in the event import template.

- Incorrect setup of the event import template used to import the event data.

  UE Loader reads the event import template to obtain instructions for loading the events, so the event import template must specify the correct mapping for the records in your log file.

- Incorrect time format of the event data.

Specifying the Maximum Load Errors to Process Before Quitting

If you choose to continue loading events when parse errors occur, you can specify the number of errors to allow before UE Loader quits. To set this maximum number of load errors, use the **infranet.uel.max_load_errors** entry in the UE Loader properties file (*BRM_Home***/apps/uel/Infranet.properties**).

Troubleshooting Connectivity Errors

If you have trouble connecting to the database, ensure that you are using the correct host name, and that the port number, database number, and host name are specified correctly in your UE Loader properties file (*BRM_Home***/apps/uel/Infranet.properties**).

Troubleshooting Preprocessing Errors

UE Loader reports two types of preprocessing errors:

- Errors found when reading the event log file.

  These errors include the record number, line number, and error description. These errors are typically the result of UE Loader finding an unexpected type of data, such as unmatched literal string indicators.

- Errors in the custom opcode that identifies the account that generated the event.

Some preprocessing errors can be fixed by editing the event import template. Errors in the template are typically fixed by a developer by using the UE Mapper application in Developer Center.

In some cases, an event log file includes a record that has not been generated correctly by the event log file source. In this case, edit the event log file manually to fix the record before loading it.

Troubleshooting Event Loading Errors

Event loading errors include the record number and contents of the record as read from the intermediate cache file. To troubleshoot loading errors, open the **uel.pinlog** file and read the contents of the error buffer returned by the BRM opcode.

Loading errors are usually the result of the following:

- Errors in the BRM system, such as an offline Data Manager (DM).
- Errors in the event import template. Errors in the template are typically fixed by a developer familiar with BRM programming concepts and techniques.

Testing Event Importing

During testing of event importing, note that:

- UE Loader does not use the time set by the **pin_virtual_time** utility. If the event start and end times are not obtained from the log file, UE Loader uses the system time.
- You may want to load a specific event log file more than once.

If you load events from an event log file successfully, UE Loader generates the log files for it, such as the success log file and the error log file. These log files must be deleted before you can reload events from the same event log file.

Reloading Events That Failed to Load

If events fail to load into the BRM database, UE Loader logs the records in an error log file called event_log_file_name_**lerr.xml**. Errors are also recorded in the event_log_file_name_**cache.xml**.

1. Find the record number of the event record(s) that failed to load by looking in the event_log_file_name_**lerr.xml** file.
2. Find the event records that failed to load in the event_log_file_name_**cache.xml** file by using the record number(s) you found in the error log file.
3. Fix the errors in the event records.
4. Reload the events by running UE Loader with the **-m load** parameter:

   ```
   uel -t template_name -m load event_log_file_name
   ```

UE Loader reads the files generated in the previous run, and ignores previously loaded events.

For more information on the cache file and error file, see "Understanding UE Loader Log Files".

Improving UE Loader Performance

You can improve UE Loader performance by adjusting the number of UE Loader threads and processes and by modifying entries in the UE Loader **Infranet.properties** file.

> **Note:**
>
> Optimal performance configuration for UE Loader depends on your system configuration such as the number and speed of CPUs and the size of your hard drive and cache memory. To find the UE Loader settings that work best for your system, you must test different combinations. For example, vary the number of threads and processes and test with different UE Mapper templates, record distribution, and queue sizes.

You can improve UE Loader performance by doing one or more of the following:

- Increase the number of threads running in UE Loader.

  Increase the number of threads in the **infranet.uei.max_threads** entry in the **Infranet.properties** file. For more information, see "Setting the Number of Threads for Performance".

> **Note:**
>
> If the loading order of records is important, for example, when loading session events, use multiple threads *only* if you configure UE Mapper to load CDRs grouped by account ID.

- Increase the number of UE Loader processes.

  You can run several UE Loader processes at the same time and load a file with each process. Do this by splitting the input file into several files before loading.

  Even better performance can be achieved by running several UE Loader processes using multiple threads in each process.

- Limit the number of records loaded into memory.

  Increase the queue size in the **infranet.uel.queue_size** entry in the **Infranet.properties** file. The **queue_size** entry specifies the number of records to load simultaneously. The queue size is shared by multiple threads, which have concurrent access to the queue. For more information, see "Setting Memory Options".

- Configure UE Loader to load CDRs grouped by account ID.

  Select **Group events by account identifier** when setting up your template in UE Mapper. For more information, start Developer Center and see UE Mapper Help.

- Load CDRs in the order they occur when using multiple threads.

  Select **Group events by account identifier** in UE Mapper and ensure that all CDRs from the same account are loaded in order (ordered by time).

Retrieving Data about Events You Load

BRM stores information about batches of events that you load in a **batch/gel** object. This object contains information such as the event log file name and the date and time loaded.

3
Universal Event Loader Utility

This chapter describes the Oracle Billing and Revenue Management Universal Event (UE) Loader utility.

Universal Event Loader

Use the UE Loader to load event records from event log files into the BRM database. For information about loading events, see "About Rating Events Created by External Sources" and "Loading Events from External Sources".

> **Note:**
>
> To connect to the database, the UE Loader needs a configuration file in the directory from which you run the utility. See the discussion of configuration files in *System Administrator's Guide*.

UE Loader runs in three modes:

- **Preprocessing mode** (**-m parse**). In the preprocessing mode, UE Loader parses the event log file and writes data to an intermediate file, including which accounts to load events for.
- **Event loading mode** (**-m load**). In event loading mode, UE Loader uses the data in the intermediate file (created in preprocessing mode) to load events into the BRM database.
- **Combined mode**. Use combined mode to run both preprocessing and event loading modes without stopping. This is the default mode.

UE Loader records all errors that occur in these modes in the log file *BRM_Home***/apps/uel/***event_log_file_name*_**lerr.xml**. It also records all event load successes in the log file *BRM_Home***/apps/uel/***event_log_file_name*_**lsucc.xml**.

For details, see "Troubleshooting Event Loading".

Location

*BRM_Home***/apps/uel**

Syntax

```
uel -t template_name [-v] [-m parse | load] [-test] event_log_file_name
```

Parameters

-t template_name
The name of the event import template.

-v
Sends error messages or debug messages to the console according to the debug level you set in the **Infranet.properties** file.

-m parse | load
The mode of operation. You specify the mode of operation only when testing the import template. When you load events in a production database, you do not need to specify the mode of operation.

- Use **parse** to read the event log file and store the results in an intermediate cache file instead of in the BRM database.
- Use **load** to load the results contained in the intermediate cache file that was created by using the **parse** entry. The UE Loader loads the events into the BRM database.
- To parse the event log file and load events in one operation, do not enter the **-m** parameter. This is the default mode in a production database.

-test
Runs the program, but loads events in calculate-only mode. To test how events are loaded, read the output flist.

event_log_file_name
The file name of the event log file from which you are loading events.

> **Note:**
>
> Do not enter the file path of the event log file. You specify the file path in the properties file. See "Specifying the Event Log File Location".

Results

For information about troubleshooting UE Loader, see "Troubleshooting Event Loading".