edocs®

Administrating Telco Analytics Manager

V4.0
Document ID: TAGN-08-4.0-01
Date Published: 9.10.03

edocs, Inc., One Apple Hill Drive, Suite 301, Natick, MA 01760

The information contained in this document is the confidential and proprietary information of edocs, Inc. and is subject to change without notice.

Preface

In This Section

Using this Manual

Welcome to Administrating Telco Analytics Manager.

This manual covers the different tasks that administrators need to carry out when working with analytical applications built using Telco Analytics Manager.

Before You Get Started

You should be familiar with the following:

- Your application architecture
- The CID and CBU databases
- Designing or working with databases and data warehouses

Who Should Read this Manual

This manual is primarily for administrators of deployed Analytical solutions. However, there are other topics covered in this manual that may interest other members of the project development team.

- Administrators

  You will find the information you need to manage a deployed solution. The information in this manual starts with an overview of the administration tasks. You will also find detailed information about other administration tasks, such as managing notifications and running the CID2CBU and CustDim loaders.

- Developers

  You will find information about running and administrating components of your development environment. You may especially be interested in the way notifications are managed and how the CID2CBU Loader handles the synchronization of changes. The chapter on the CustDim Loader has information about how the administrator uses and troubleshoots the loading of Customer Dimensions. You may also want to consult the Administration Tool Reference for the comprehensive list of administration tools and their commands.

- Project Architect

  You will find information about the different components which may require administration or intervention by administrators. There are some features that may be of interest when designing your solution. For instance, you need to be aware of how the notifications are handled. You may also be interested in the different aspects of the CustDim loader as it is run manually to load the CBU.

- Project Manager

  You will find information about the configuration and processes of the runtime environment you need to take into account when planning the development of your solution. There is important information about the loaders and synchronization of information in the CBU.

How this Manual is Organized

This manual contains the following chapters:

- **Overview of Administrating**

  This chapter covers the basics of managing Account Management solutions:

  - The role of an administrator

- **Managing Notifications**

  This chapter covers managing notifications in the CID:

  - Overview of notifications and their status codes
  - Viewing notifications
  - Cancelling notifications
  - Purging notifications
  - Resubmitting notifications
  - Forcing synchronization

- **Managing Reports**

  This chapter covers managing reports saved as files:

  - Purging corrupt reports
  - Purging reports
  - Purging the cache

- **Working with the CID2CBU Loader**

  This chapter covers working with the CID2CBU Loader.

  It contains information about:

  - Starting the CID2CBU Loader
  - Stopping the CID2CBU Loader.

- **Working with the CustDim Loader**

  This chapter covers working with the CustDim Loader.

  It contains information about:

  - Running the CustDim Loader
  - Troubleshooting

- **Monitoring Telco Analytics Manager**

  This chapter covers monitoring using the logger.

  It contains information about:

  - The logger and its components
  - Overview of using logs
  - Configuring the logger
  - Working with log files

- **Administration Tool Reference**

  This appendix is an administration tool reference guide. It covers the location, configuration and use of the administration tools

  The administration tools covered include:

  - CBU Administration Tool
  - NOTIFYORG Administration Tool
  - CID2CBU Loader Administration Tools
  - CustDim Loader Administration Tools

What Typographical Changes and Symbols Mean

This manual uses the following conventions:

TYPEFACE	MEANING	EXAMPLE
Italics	Manuals, topics or other important items	Refer to *Developing Connectors*.
Small Capitals	Software and Component names	Your application uses a database called the CID.
Fixed Width	File names, commands, paths, and on screen commands	Go to `//home/my file`

Finding the Information You Need

edocs Telco Solutions comes with comprehensive documentation that covers all aspects of building TAM solutions. You should always read the release bulletin for late-breaking information.

Getting Started

If you are new to edocs Telco Solutions, you should start by reading *Introducing Telco Analytics Manager*. This manual contains an overview of the various components along with a list of the available features. It introduces various concepts and components you must be familiar with before moving on to more specific documentation. Once you have finished, you can read the manual that covers different aspects of working with the application. At the beginning of each manual, you will find an introductory chapter that covers concepts and tasks.

Designing Your Solution

While reading *Introducing Telco Analytics Manager*, you can begin to envision how the different components can address your solution's needs.

You can refer to *Developing Analytical Applications* for information about customizing the database, synchronizing data with Telco Service Manager (TSM), loading data from external invoice files, and other design issues. The *CBU Reference Guide* also gives you the information about how the information in your solution is managed and stored. You should also read the section on integrating Telco Analytics Manager with Telco Service Manager in *Developing Analytical Applications*.

Installing Your Analytical Application

You should start by reading the Release Bulletin. For detailed installation and configuring information, refer to *Installing Telco Analytics Manager*. This manual covers installing *Telco Analytics Manager* on one or more computers. It also contains the information you need to configure the different components you install.

Building Analytical Solutions

If you are designing and programming analytical applications, you have several different sources of information. If you are programming the user interface of the solution, you should read *Building Reports*. You can also refer to the *QRA API Specification* and the *QRA Configuration File Reference Documentation* for detailed information about the different components you can use to build reports which serve as the user interface. For configuring the various components, refer to *Installing Telco Analytics Manager* and sections in other documents that specifically deal with the component to be configured.

If you are working on the data warehouse side of TAM and are interested in how the information will be loaded into the data warehouse, you should read *Developing Analytical Applications*. For more information about the design and structure of the CBU, you should refer to the *CBU Reference Guide* along with the *CBU Reference* documentation for your database.

Integrating TAM and TSM

If you are involved in configuring your solution to work with Telco Service Manager, you should read *Introducing Telco Analytics Manager* for an overview of the components and how they interact. You should then read *Developing Analytical Applications* for information about synchronizing data between the Telco Analytics Manager and Telco Service Manager. In this manual, you will also find information about loading data in both the CBU and the Telco Service Manager.

Managing Telco Analytics Manager

If you are responsible for managing Telco Analytics Manager, you should read the *Installing Telco Analytics Manager* for information about configuring various components. *Administrating Telco Analytics Manager* covers what you need to know about managing your solution at runtime.

If You Need Help

Technical support is available to customers who have valid maintenance and support contracts with edocs. Technical support engineers can help you install, configure, and maintain your edocs application.

To reach the U.S. Service Center, located in Natick, MA (Monday through Friday 8:00am to 8:00pm EST):

- Telephone: 508.652.8400
- Toll Free: 877.336.3362
- E-support: support.edocs.com (This requires a one-time online registration)
- E-mail: support@edocs.com

When you report a problem, please be prepared to provide us the following information:

- What is your name and role in your organization?
- What is your company's name?
- What is your phone number and best times to call you?
- What is your e-mail address?
- In which edocs product did a problem occur?
- What is your Operating System version?
- What were you doing when the problem occurred?
- How did the system respond to the error?
- If the system generated a screen message, please send us that screen message.
- If the system wrote information to a log file, please send us that log file.

If the system crashed or hung, please tell us.

Contents

CHAPTER 1

Overview of Administrating

In This Section

About Administrating

Telco Analytics Manager is designed to run with a minimum amount of user intervention or maintenance.

To ensure that your system runs smoothly and meets the availability and performance demands your users expect, you have to perform some maintenance and administrative tasks. Along with this set of tasks, you can also monitor the behavior of your application to make sure it is running without problems. You can use the message logger to pinpoint problems.

Administrating Telco Analytics Manager involves:

- Managing Notifications

  Some changes in the CID require the CBU to be updated. When this happens, a notification event occurs. You need to make sure that these events are handled correctly by the CID2CBU loader.

- Working with the CID2CBU loader

  The CID2CBU loader handles the update of information in the CBU with information in the CID.

- Working with the CustDim Loader

  The information in the CBU contains information from various sources. You use the CustDim loader to load information into the CBU either the first time you set it up or periodically as required.

- Monitoring your Application

  You can use the system logger to monitor and track the behavior of your Analytical application.

CHAPTER 2

Managing Notifications

Managing notifications involves:

- Viewing pending notifications
- Viewing a notification detail
- Cancelling notifications
- Resubmitting notifications
- Purging notifications
- Managing synchronization of Organizations

Telco Analytics Manager comes with administration tools to help you carry out these tasks.

In This Section

About Managing Notifications

Changes to specific objects generate notification events. A notification event means that the information in the CID has changed and that the data in the CBU must be updated.

These events are stored in the CID `NOTIFICATION` table. This table serves as the notification queue. This queue is polled at specified intervals by the CID2CBU loader and processes the entries in FIFO (First in/First Out).

Managing notifications involves:

- Viewing pending notifications
- Viewing a notification detail
- Cancelling notifications
- Resubmitting notifications
- Purging notifications
- Managing synchronization of Organizations

The Telco Analytics Manager comes with administration tools to help you carry out these tasks. For more information about the administration tools and their syntax, refer to the A*dministration Tool Reference* chapter in this manual.

Notification Status Codes

This table describes the different notification status codes and their names.

Notification Status Codes

STATUS CODE	NAME
2	NOT YET NOTIFIED
4	NOTIFICATION DONE
5	NOTIFICATION FAILED
8	NOTIFICATION IN PROGRESS

Viewing Notifications

You can view a list of the failed and in progress notifications in the CID. You can also view the detail of a specific notification.

To view failed notifications

1 Go to `<home_dir>/bin`.

2 Run the CID administration tool. Use the syntax:

```
cidAdminTool admin_notifications <CID> <CID_ADMIN login>
<CID_ADMIN password>  [-quiet] [-datablocksize:####]
```

where `<CID>`:

- Oracle: `<instance alias>`
- DB2: `<database alias>`
- SQL Server: `<database host> [:<port>]` If no port is specified, the tool uses the default SQL server port

The CID administration tool displays a menu.

3 Choose `1) View failed notifications` then press Enter. The CID administration tool asks you to enter the number of days the notifications have been set to FAILED.

4 Do one of the following:

- Enter `0` to display all of the pending notifications
- Enter the number of days the notifications have been set to failed

5 Press Enter. The CID administration tool displays a list of failed notifications.

To view in progress notifications

1 Go to `<home_dir>/bin`.

2 Run the CID administration tool. Use the syntax:

```
 cidAdminTool admin_notifications <CID> <CID_ADMIN login>
<CID_ADMIN password>  [-quiet] [-datablocksize:####]
```

where `<CID>`:

- Oracle: `<instance alias>`
- DB2: `<database alias>`
- SQL Server: `<database host> [:<port>]` If no port is specified, the tool uses the default SQL server port

The CID administration tool displays a menu.

3. Choose 2`) List notifications 'in progress'` then press Enter. The CID administration tool asks you to enter the number of days the notifications have been IN PROGRESS.
4. Do one of the following:
   - Enter `0` to display all of the notifications in progress
   - Enter the number of days the notifications have been set to IN PROGRESS
5. Press Enter. The CID administration tool displays a list of notifications.

To view a notification detail

1. Go to `<home_dir>/bin`.
2. Run the CID administration tool. Use the syntax:

   ```
    cidAdminTool admin_notifications <CID> <CID_ADMIN login>
   <CID_ADMIN password>  [-quiet] [-datablocksize:####]
   ```

   where `<CID>`:

   - Oracle: `<instance alias>`
   - DB2: `<database alias>`
   - SQL Server: `<database host> [:<port>]` If no port is specified, the tool uses the default SQL server port

   The CID administration tool displays a menu.
3. Choose `3) View Notification detail` then press Enter. The CID administration tool asks you to enter the ID of the notification to display.
4. Enter the ID of the notification then press Enter. The CID administration tool displays the following information:
   - Notification ID
   - Creation date
   - Modification date
   - Type
   - Notified object type
   - First name
   - Last name/Business:
   - Status

Cancelling Notifications

You can cancel specific notifications or you can cancel all of the notifications. Cancelling a notification means you change its status to failed. To remove the notification from the CID, you need to purge the failed notifications.

To cancel a notification

1 Go to `<home_dir>/bin`.

2 Run the CID administration tool. Use the syntax:

```
cidAdminTool admin_notifications <CID> <CID_ADMIN login>
<CID_ADMIN password> [-quiet] [-datablocksize:####]
```

where `<CID>`:

- Oracle: `<instance alias>`
- DB2: `<database alias>`
- SQL Server: `<database host> [:<port>]` If no port is specified, the tool uses the default SQL server port

The CID administration tool displays a menu.

3 Choose `5) Set a notification 'in progress' to 'failed'` then press Enter. The CID administration tool asks you to enter the ID of the notification to cancel by setting its status to failed.

4 Enter the ID of the notification then press Enter.

5 At the prompt, press Y to confirm.

The CID administration tool cancels the notification and displays a confirmation message.

To cancel all notifications

1 Go to `<home_dir>/bin`.

2 Run the CID administration tool. Use the syntax:

```
 cidAdminTool admin_notifications <CID> <CID_ADMIN login>
<CID_ADMIN password> [-quiet] [-datablocksize:####]
```

where `<CID>`:

- Oracle: `<instance alias>`
- DB2: `<database alias>`
- SQL Server: `<database host> [:<port>]` If no port is specified, the tool uses the default SQL server port

The CID administration tool displays a menu.

3. Choose `7) Set all notifications 'in progress' requests to 'failed'` then press Enter. The CID administration tool asks you to enter the number of days the notifications have been in progress that you want to set to `FAILED`.
4. Do one of the following:

   - Enter `0` to process all of the notifications in progress
   - Enter the number of days the requests have been set to in progress

   The CID administration tool displays a list of notifications to cancel.
5. At the prompt, press Y to confirm.

   The CID administration tool cancels the notifications and displays a confirmation message.

Purging Notifications

You can remove notifications from the CID. You can purge notifications that have the following statuses:

- FAILED
- DONE

To purge notifications

1 Go to <home_dir>/bin.

2 Run the CID administration tool. Use the syntax:

```
cidAdminTool purge_notifications <CID> <CID_ADMIN login>
<CID_ADMIN password> <days> [<status>,<status>,...] [-quiet]
[-datablocksize:####]
```

where <CID>:

- Oracle: <instance alias>
- DB2: <database alias>
- SQL Server: <database host> [:<port>] If no port is specified, the tool uses the default SQL server port

The CID administration tool removes all of the notifications that:

- Have the specified statuses
- Are older than the specified number of days

If no statuses are specified, the tool removes FAILED and DONE notifications.

The CID administration tool displays a confirmation message.

Resubmitting Notifications

You can resubmit specific notifications. In general, you do not need to resubmit notifications. However, unforeseen problems or application failures do occur. When this happens, you may need to resubmit some notifications. For instance, if the connection to the CBU is lost, you may have several notifications that are in progress. This means that the notifications have been read but not yet processed.

To resubmit a notification

1 Go to `<home_dir>/bin`.

2 Run the CID administration tool. Use the syntax:

```
 cidAdminTool admin_notifications <CID> <CID_ADMIN login>
<CID_ADMIN password>  [-quiet] [-datablocksize:####]
```

where `<CID>`:

- Oracle: `<instance alias>`
- DB2: `<database alias>`
- SQL Server: `<database host> [:<port>]` If no port is specified, the tool uses the default SQL server port

The CID administration tool displays a menu.

3 Choose 4`) Set a notification 'in progress' to 'not yet notified'` then press Enter. The CID administration tool asks you to enter the ID of the notification to resubmit by setting its status to not yet notified.

4 Enter the ID of the notification then press Enter.

5 At the prompt, press Y to confirm.

The CID administration tool resubmits the notification and displays a confirmation message.

To resubmit all notifications

1 Go to `<home_dir>/bin`.

2 Run the CID administration tool. Use the syntax:

```
 cidAdminTool admin_notifications <CID> <CID_ADMIN login>
<CID_ADMIN password>  [-quiet] [-datablocksize:####]
```

where `<CID>`:

- Oracle: `<instance alias>`
- DB2: `<database alias>`
- SQL Server: `<database host> [:<port>]` If no port is specified, the tool uses the default SQL server port

The CID administration tool displays a menu.

3 Choose 6`) Set all notifications 'in progress' to 'not yet notified'` then press Enter. The CID administration tool asks you to enter the number of days the notifications have been in progress that you want to resubmit by setting the status to not yet notified.

4 Do one of the following:

- Enter 0 to process all of the notifications in progress
- Enter the number of days the notifications have been in progress

5 At the prompt, press Y to confirm.

The CID administration tool resubmits the notifications and displays a confirmation message.

Managing Organization Notifications

You can manage the synchronization of organization data in the CID and CBU. In general, you do not need to manage the synchronization of organization data. However, if an error occurs when processing notifications, you may need to force the resynchronization of some data. For instance, you may have a create organization notification in the queue along with a modify organization notification that adds a contract to this organization. If the create organization notification fails, then the modify organization adding a contract to this organization is meaningless because the organization does not yet exist in the CBU.

An organization has two flags that allow you to manage the notification and synchronization of changes to organization data in the CID. The flags are:

- `NotificationActivated`

  Specifies whether or not changes to the organization generate notification events for the CBU. This option is deactivated by default.
- `NotificationRequested`

  Specifies whether or not a notification event has already been generated for changes to this organization and placed in the notification queue. When true, a notification has been generated and placed in the notification queue. When the CBU has been successfully synchronized with the latest organization change, this flag is reset to false.

You use the `notifyorg` tool to force the synchronization of organization data in the CID and CBU.

This tool does the following:

- Generates a notification event for each organization with a `NotificationRequested` flag set to `1`.
- For each organization with a `NotificationRequested` flag set to `1`, calls the notification logic to determine if an update organization event must be generated.

For more information about the default notification logic and how to write your own, refer to *Developing Analytical Applications - Customizing Notifications.*

Managing the synchronization for organizations involves:

- Resetting organization `NotificationRequested` flags
- Running the `notifyorg` tool to force synchronization of organization data

Before running the `notifyorg` tool to force synchronization, you need to make sure that there are no failed notifications in the queue.

To force the synchronization of an organization

1. Go to `<home_dir>/bin`.
2. Run the CID administration tool. Use the syntax:

   ```
   cidAdminTool admin_notifications <CID> <CID_ADMIN login>
   <CID_ADMIN password>  [-quiet] [-datablocksize:####]
   ```

   where `<CID>`:

   - Oracle: `<instance alias>`
   - DB2: `<database alias>`
   - SQL Server: `<database host> [:<port>]` If no port is specified, the tool uses the default SQL server port

   The CID administration tool displays a menu.
3. Choose `8) Force synchronization of an organization` then press Enter. The CID administration tool asks you to enter the ID of the notification. The `NotificationRequested` flag of the organization of this notification will be set to `true`.
4. Enter the ID of the notification then press Enter.
5. At the prompt, press Y to confirm.
6. The CID administration tool modifies the organization's flag and displays a confirmation message.

To force the synchronization of all organizations

1. Go to `<home_dir>/bin`.
2. Run the CID administration tool. Use the syntax:

   ```
    cidAdminTool admin_notifications <CID> <CID_ADMIN login>
   <CID_ADMIN password>  [-quiet] [-datablocksize:####]
   ```

   where `<CID>`:

   - Oracle: `<instance alias>`
   - DB2: `<database alias>`
   - SQL Server: `<database host> [:<port>]` If no port is specified, the tool uses the default SQL server port

   The CID administration tool displays a menu.
3. Do one of the following:

   - Choose `9) Force synchronization of all organizations with 'failed' notifications`
   - Choose `10) Force synchronization of all organization with notifications 'in progress'`

4 Press Enter. The CID administration tool asks you to enter the number of days the notifications have been set to their current status.

5 Do one of the following:

- Enter `0` to process all of the notifications
- Enter the number of days the notifications have been set to their current status

6 At the prompt, press Y to confirm.

The CID administration tool modifies the organization's flag and displays a confirmation message.

Before you run the `notifyorg` administration tool, make sure you purge all previous notifications with a `failed` or `in progress` statuses.

To synchronize organization information in the CBU

1 Go to `<home_dir>/bin`.

2 Run the `notifyorg` administration tool. Use the syntax:

```
notifyorg [-datablocksize:####]
```

The `notifyorg` administration tool synchronizes the organization information in the CBU. The organizations in the CID with active Notification Requested flags are updated in the CBU. When finished, the tool displays a summary of the organizations it processed.

CHAPTER 3

Managing Reports

In This Section

About Managing Reports

Your Telco Analytics Manager may use the Report Manager to allow users to save reports. If it does, you need to be able to manage these reports. You need to be able to remove files from directories and purge files.

When creating and saving reports, the Report Manager creates temporary files. These files are:

- `.part` files

  These temporary files correspond to report files that are being created.
- `.lock` files

  These temporary files are used to control concurrent use of a specific file.

Under normal circumstances, these temporary files are automatically managed and purged. However, you may need to purge these temporary files.

Managing reports involves:

- Purging corrupted reports
- Purging reports
- Purging the cache

Purging Reports

Purging reports involves:

- Purging corrupted reports
- Purging all reports
- Purging the cache

To purge corrupted reports

1. Go to `<home_dir>/bin`.
2. Run the Report Manager administration tool. Use the syntax:

   ```
   reportmanageradm purgeCorrupted <repositoryCode>
   ```

   where `<repositoryCode>` is the directory declared in the Report Manager configuration file.

   The Report Manager administration tool removes all of the report files that:

   - Are older than the specified validity period
   - Are older than the specified `timetolive`
   - Are temporary files which have not been validated (with a `.part` extension)

   The Report Manager administration tool displays a confirmation message.

To purge reports

1. Go to `<home_dir>/bin`.
2. Run the Report Manager administration tool. Use the syntax:

   ```
   reportmanageradm purgeAllReports <repositoryCode> [<keyCode>]
   ```

   where:

   `<repositoryCode>` is the directory declared in the Report Manager configuration file.

   `<keyCode>` is the optional key code corresponding to a specific directory/user

   The Report Manager administration tool removes all of the report files and removes the directory.

   The Report Manager administration tool displays a confirmation message.

To purge the cache

1. Go to `<home_dir>/bin`.
2. Run the Report Manager administration tool. Use the syntax:

```
reportmanageradm purgeCache <repositoryCode>
```

where:

`<repositoryCode>` is the directory declared in the Report Manager configuration file.

The Report Manager administration tool removes all files and removes the directory.

The Report Manager administration tool displays a confirmation message.

CHAPTER 4

Working with the CID2CBU Loader

In This Section

About Working with the CID2CBU Loader

The CID2CBU Loader is a run-time component to process the notifications in the CID.

As an administrator, you do the following:

- Start the CID2CBU Loader
- Stop the CID2CBU Loader

For information about notifications, refer to *Managing Notifications* in this manual.

Running the CID2CBU Loader

The CID2CBU loader processes the notification events in the CID.

You use a set of administration tools to start and stop the CID2CBU loader. The administration tools are:

- `cid2cbuloader`
- `cid2cbuloaderadm`

These administration tools are located in `<home_dir>/bin`.

These tools use the `cid2cbuloader.properties` configuration file to set the properties of the CID2CBU loader. This file is located in `<home_dir>/config/cid2cbuloader`.

To start the CID2CBU loader

1. Go to `<home_dir>/bin`.
2. Run `cid2cbuloader`.

   The CID2CBU loader loads its configuration file and starts the processes it needs. When finished, it displays a message.

To stop the CID2CBU loader

1. Go to `<home_dir>/bin`.
2. Run `cid2cbuloaderadm`. Use the syntax:

   ```
   cid2cbuloaderadm <host> <port> shutdown
   ```

   When finished, it displays a message.

CHAPTER 5

Working with the CustDim Loader

The CustDim Loader is an ISF connector.

As an administrator, you do the following:

- Start the CustDim Loader
- Monitor CustDim Loader
- Troubleshoot

For more information about working with the ISF and ISF connectors, refer to *Developing Connectors*.

In This Section

About Working with the CustDim Loader

The CustDim Loader is an ISF connector.

As an administrator, you do the following:

- Start the CustDim Loader
- Monitor CustDim Loader
- Troubleshooting

For more information about working with the ISF and ISF connectors, refer to *Developing Connectors*.

Running the CustDim Loader

To start the CustDim Loader

1. Go to `<home_dir>/bin`.
2. Run `custdimloader`. Use the syntax:

   ```
   custdimloader custdimloader
   ```

   The connector is loaded and it starts the processes it needs.

To pause the CustDim Loader

1. Go to `<home_dir>/bin`.
2. Run `custdimloaderadm`. Use the syntax:

   ```
   custdimloaderadm <host> <port> pause
   ```

   This command forces the connector to pause the extraction of new messages from all its inbound queues (invoice file for the splitter, message for the transformer and message for the CID loader and CBU loader)
3. Run `custdimloaderadm` to resume processing. Use the syntax:

   ```
   custdimloaderadm <host> <port> resume
   ```

   This command forces the connector to resume the extraction of new messages from all its inbound queues.

To stop the CustDim loader

1. Go to `<home_dir>/bin`.
2. Run `custdimloaderadm`. Use the syntax:

   ```
   custdimloaderadm <host> <port> shutdown
   ```

   This command waits for the completion of the current input file and all of its associated processing.

Monitoring the CustDim Loader

In order to help you pinpoint problems, the CustDim loader logs events in the configured log output.

Statistical log entries occur:

- When the connector ends its initialization
- When the connector starts processing invoice files
- When the connector loads customer dimensions in the CID
- When the connector loads customer dimensions in the CBU
- When the connector shuts down because of no more data to process or because a shutdown command has been received
- At specified periodic intervals configured in the connector startup command file

They contain statistical information about the processing of the connector, including:

- Number of invoices files to process
- Number of invoices generated per invoice file
- Number of invoices successfully inserted in the CID
- Number of invoices successfully inserted in the CBU

For more information about logs, refer to *Monitoring Telco Analytics Manager* in this manual.

Recovering from Errors

Handling Connector Process Crashes

When running the CustDim Loader, your connector process may crash unexpectedly. When this happens, the current state of the connector is lost and the current processing of an invoice flow is interrupted.

In order to recover from a crash, no specific action is required. Restart your connector and the processing of your invoice flow will start again.

Remember that recovering from a connector process does not damage or corrupt the CBU or CID databases. This means that running the invoice load in CID and CBU is not a problem.

Handling Splitting Errors

When running the CustDim Loader, the invoice file being processed may contain corrupted data. When this happens, errors may occur during the splitting phase.

When such errors occur, the CustDim Loader generates a message in the splitting error queue. This message identifies the file which generated the error.

When you are finished fixing the error, you must reprocess the file.

Handling Transformation Errors

When running the CustDim Loader, the transformation of the invoice format may generate some errors. When this happens, the legacy invoice message is put in the transformation error queue.

You can do the following:

- Correct the problem in the invoice message and resubmit it for transformation by copying the message into the transformation retry queue
- Get a new copy of the invoice file and retry the whole invoice flow

Handling CID Update Errors

When running the CustDim Loader, errors may occur when updating the CID. When this happens, the invoice message is put in the CID Update error queue.

You can do the following:

- Correct the problem in the invoice message and resubmit it for processing by copying the message into the CID Update retry queue
- Get a new invoice file containing the corrected invoice and retry the whole invoice flow

Handling CBU Update Errors

When running the CustDim Loader, errors may occur when updating the CBU. When this happens, the invoice message is put in the CBU Update error queue.

You can do the following:

- Correct the problem in the invoice message and resubmit it for processing by copying the message into the CBU Update retry queue
- Get a new invoice file containing the corrected invoice and retry the whole invoice flow

CHAPTER 6

Monitoring Telco Analytics Manager

In This Section

About Monitoring Applications using Logs

You can use the system logger to create logs while your Telco Analytics Manager is running. These logs are very helpful when you have to monitor system activity and are helpful in solving problems and pinpointing system or architecture bottlenecks.

The common logger features are available for the different product components and each component can have its own specific logger configuration and output. You can log events that occur in the:

- Presentation Layer
- Synchronizer components
- CID2CBU loader
- CustDim Loader

Using logs to monitor your application involves the following:

- The capability to define meaningful alerts based on logs the application generates.
- The assumption that your application works fine until an alert has been raised.
- Your application does not only deal with core components.

This also means that you use a supervision platform or tool to gather information and generate alerts. The enhanced system logger helps you make your application an integral part of a comprehensive monitoring system.

There are also different professionals who supervise applications and who can use these logs to monitor applications:

- **Developers**

  They can define what should be logged and monitored. They also determine the type of alert to generate and define possible corrective actions.

- **Administrators**

  They are responsible for monitoring the application and define and integrate system logs and the monitoring system.

Logger Events

ATTRIBUTE	DESCRIPTION
Date/Time	ISO-8601 date string to identify the time
Thread id	To identify the thread that hosts the event generator
Unique id	To identify the event
Session id	To identify the user session
Type	To classify the impact of event
Severity	To classify the issue level
Module	To identify the event technical source
Code	To classify and describe what occurs
Description	To classify and describe what occurs – linked to code
Debug information	To provide additional technical information

Event Types

EVENT TYPE	DESCRIPTION
INIT	Covers application/module initialization processes
STATE	Covers application/module state changes
EXCEPTION	Covers internal exceptions
SESSION	Covers user session life cycle
REQUEST	Covers request life cycles
MESSAGE	Covers lifecycle in the integration framework
OBJECT	Covers object handling
RESOURCE	Covers component events
DATA	Covers customer data handling in the CID
NONE	Unclassified events

Severity Levels

SEVERITY LEVEL	DESCRIPTION
FATAL	Events that have an impact on the availability of the application
ERROR	Events that cause a given application workflow to not work properly
WARN	Events that may impact the behavior of the application
INFO	Events that record successful basic system actions for supervision
DEBUG	Level 0: no debug information. Level 3: for activation in a production environment within a working day. Level 5: for activation in a production environment for a limited period. Level 7: for activation in a very limited way with one or two concurrent users.

Event Modules

EVENT MODULE	DESCRIPTION
AGT	Any agent – synchronizer, connector, sequencer
BLM	Business Logic Manager
DAL	Data Access Layer
ISF	ISF Framework and all of its sub modules
JFN	JSP Framework
JSP	Java Server Pages
LOG	The logger platform
NIL	No module involved
QRA	Query, Reporting, and Analysis module
UTL	Internal utility components
WFS	Report manager

Event Code/description:

- Events are coded to ensure the description, for a given case, is always the same whatever the event source is.
- A code is associated with a description, which is the actual event message.
- The code is the reference to detect what occurs.

Overview of Using Logs

When using logs to monitor your application, you have to decide on the following:

- The goal for logs
- Which components to monitor
- The need for dynamic configuration
- The information to generate
- The output format

Determining the Goals

You have to determine what exactly you want to do with the logs. Once you decide this, you can easily determine what needs to be done in order to configure and set up the logger.

- If you work with a supervision platform that monitors a text-based source, you should use the circular file (rolling file) driver. We recommend this because even though there are backups, logs are always generated into the main output, with a constant file name.
- If you want to backup and route the generated logs regularly, you should use the time stamped file (daily file) driver. There are several backup capabilities.
- If you want to check what occurs in the application, you should use the standard output.

Determining the Components to Monitor

You can get events from several product layers:

- Presentation Layer

  This layer covers modules that run in the application server.
- Integration Layer
- System components

The components to monitor depend on your application features and architecture. For instance, if your application does not use the order validation feature, you do not use the Approval sequencer. Therefore you do not need to worry about monitoring its logs. The same is true with connectors. If you do not synchronize your backend systems with a connector, there is no need to work with connector logs.

Determining the Need for Dynamic Configuration

You can configure the logger dynamically. This means you can change the logger configuration without restarting the component. Using this feature involves a thread that reloads the logger configuration regularly.

You may need to use this feature in the following environments:

- Production environment

  In this environment, activating the dynamic reload is trade-off between the need to increase the amount of logged information when an issue occurs and having an additional thread running. This also depends on the supervision process and if it covers the capability to change the logger configuration and get more information.

- Application acceptance test / Pre-integration environment

  In these environments, dynamic configuration is helpful to speed up tasks. As issues may occur frequently, using this feature helps get the additional information you need to pinpoint a problem and find a solution quickly.

- Development

  When developing, dynamic configuration is helpful to continuously adapt the log to the components being worked on or tested.

Default Settings:

- The presentation layer logger does not use dynamic configuration.
- Standalone components (synchronizer, connector, approval sequencer) use dynamic configuration with a configuration reload every 5 minutes.

Determining the Level of Information

In order to obtain information you can use, you need to determine the type of information to log.

You use the following event attributes to make sure the logs contain the information you need:

- Event Type

  Basically, you could decide to log all event types, but it really depends on your supervision process.

  For instance, you may not want to focus on init/stop phases. In that case, there is no need to get 'INIT'-types events.

- Event Severity

  The logger configuration offers capability to define the minimum severity level you want to log.

If you select 'ERROR', the log only contains events with severity 'ERROR' or 'FATAL'.

For instance, you log for a monitoring platform and are trying to pinpoint some problems. In this case, you set the minimum severity level to 'WARN'.

Determining the Output Format

The format of your output depends of the way you plan on using the logs. For instance, you most likely use different output formats when working with a system console and a supervision platform.

When determining your format, keep in mind:

- The logger generates one line per event
- You can configure the pattern code
- You can arrange attributes the way you want and you can use any allowed character separator
- You can decide to log all attributes or not.

Configuring the Logger

You can customize the logger for each component. For example, your presentation layer logger can log only logins and critical errors and your Synchronizer logger tracks debug information.

For each component, you use the following configuration files:

- `logger.properties` to set the basic logger properties
- `log4j.properties` to set the dynamic configuration parameters

The location of the configuration files:

COMPONENT	PATH

Presentation Layer	`<home_dir>/classes/nmycfg/util`
	`<home_dir>/<channels>/WEB-INF/classes/nmycfg/util`

Synchronizer	`<home_dir>/config/synchronizers/synchronizer/util`
Approval sequencer	`<home_dir>/config/approvalsequencer/util`
Connector Template	`<home_dir>/config/connectors/loopback/nmycfg/util`
Loopback Connector	`<home_dir>/config/connectors/connectortemplate/nmycfg/util`

Although each component has its own set of configuration files, you configure the components the same way. The instructions in this section apply to configuring the logger for all components.

Setting the Basic Properties

You use the `logger.properties` configuration file to set the following basic properties of each logger:

- Location of the dynamic configuration file

  Path of the `log4j.properties` configuration file that contains information on the events to log and their format.
- Frequency to reload the parameter file

  The time in seconds to reload the `log4j.properties` configuration file.

To set the path of the parameter file

1. Go to the `/util` directory containing the `logger.properties` configuration file.
2. Open `logger.properties`.
3. For the `logger.log4j.properties` setting, enter the name of the `log4j.properties` configuration file.

4 Save your changes.

The two files must be in the same path.
The `log4j.properties` configuration file is found using the `CLASSPATH`.

To set the reload frequency

1 Go to the `/util` directory containing the `logger.properties` configuration file.

2 Open `logger.properties`.

3 Change the `logger.log4j.reloadDelay` setting to one of the following:

- The number of seconds to wait before reloading the `log4j.properties configuration file`
- 0 to deactivate reloading

4 Save your changes.

By default, the logger configuration disables the dynamic reloading of the configuration for Presentation Layer.
This is due to J2EE specifications that recommend not creating threads, except when explicitly required.

Specifying the Events to Log

You use the `log4j.properties` configuration file to set the types of events to log.

By specifying different types of events to log, the logs contain only information about the different types of events you want to track.

For each event type, you specify:

- The minimum severity level
- Debug level when required
- The target for the generated events (output)

To specify the types of events to log

1 Go to the `/util` directory containing the `log4j.properties` configuration file.

2 Open `log4j.properties`.

3 Under `EVENTS`, enter the event to log. Use the syntax:

```
<Event Type> = <Severity Level>, [Debug Level], <Output Driver
#1>, <Output Driver #2>,...
```

`<Event Type>` is one of the following event type patterns:

EVENT TYPE	DESCRIPTION
All types	log4j.category.nmy
INIT	log4j.category.nmy.INIT
STATE	log4j.category.nmy.STATE
EXCEPTION	log4j.category.nmy.EXCEPTION
SESSION	log4j.category.nmy.SESSION
REQUEST	log4j.category.nmy.REQUEST
MESSAGE	log4j.category.nmy.MESSAGE
OBJECT	log4j.category.nmy.OBJECT
RESOURCE	log4j.category.nmy.RESOURCE
DATA	log4j.category.nmy.DATA
NONE	log4j.category.nmy.NONE

`<Severity Level>` is the minimum security level you want the event type to be logged with:

SEVERITY LEVEL	LEVELS OF EVENTS LOGGED
FATAL	FATAL
ERROR	FATAL, ERROR
WARN	FATAL, ERROR, WARN
INFO	FATAL, ERROR, WARN, INFO
DEBUG	FATAL, ERROR, WARN, INFO, DEBUG

When the `<Severity Level>` is DEBUG, you can enter the `[Debug Level]`:

NAME	NOTES
0	Minimum debug information
3	Events with debug level 3
5	Events with debug level 3 or 5
7	Events with debug level 3, 5, or 7

`<Output Driver>` is the output driver to use:

DRIVER	CODE
Standard output driver	CON
Rolling file output driver	ROL

DRIVER	CODE
Daily file output driver	DAY

You enter as many lines in the `EVENTS` section as there are event types to track. To disable an event type, enter a line `<Event Type>=INFO,DISABLED`

Configuring the Logger Output

You use the `log4j.properties` configuration file to set the properties of the output drivers.

For each output driver, you specify:

- Target of the driver
- Driver specific properties
- Format of the log

About the Standard Output Driver

This output driver sends log information to your application's standard output (stdout.) This log information is included in all other standard application output.

To configure a standard output driver

1. Go to the `/util` directory containing the `log4j.properties` configuration file.
2. Open `log4j.properties`.
3. Go to the `#STANDARD OUTPUT DRIVER SETTINGS` section.
4. Set `log4j.appender.CON.Target` to one of the following:
   - `System.out` to redirect output to the standard output
   - `System.error` to redirect output to the error output.
5. Set the `log4j.appender.CON.layout.ConversionPattern` to specify the log format.
6. Save your changes.

```
log4j.appender.CON=org.apache.log4j.ConsoleAppender
log4j.appender.CON.Target=System.out
log4j.appender.CON.layout=com.netonomy.util.logger.LoggerLayout
log4j.appender.CON.layout.ConversionPattern=+{DATE_TIME};{EVENT_ID};{SEVERITY};{MODULE};{THREAD_ID};{EVENT_TYPE
};{ERROR_CODE};{DESCRIPTION}
```

About the Rolling File Output Driver

This output driver saves log information in a text file.

When using this driver, you specify the following:

- The name and location of the log file
- Maximum size of the text file.

  When this limit is reached, the logger saves the file as `<log_filename>.1` and starts a new log file. When there is more than one saved logger file, the logger increments the filename of the saved log files. This means that the `<log_filename>.1` is the latest saved log file.
- Number of backup copies.

  Determines the number of backup copies the logger keeps. When this limit is reached, the logger deletes the oldest backup copy.

Make sure you specify the correct file size and the number of backup copies.

If you do not, you may lose some log information as the logger automatically deletes the oldest log files.

To configure a rolling file output driver

1. Go to the `/util` directory containing the `log4j.properties` configuration file.
2. Open `log4j.properties`.
3. Go to the `#ROLLING FILE OUTPUT DRIVER SETTINGS` section.
4. Modify the following settings:

SETTING	DESCRIPTION
log4j.appender.ROL.File	Full path and filename of the log file
log4j.appender.ROL.MaxFileSize	Maximum size of the log file. Use the syntax: <size><unit> Units can be either: KB MB GB For a log file with a maximum size of 4MB, enter: log4j.appender.ROL.MaxFileSize=4MB

5. Save your changes.

Saved copies are named `<log_filename>.<number>` and incremented when the logger creates a new copy.
This means that `<log_filename>.1` is the latest saved log file.

Default Rolling File Output Settings

By default, the Rolling file output driver has the following configuration:

- MaxFileSize: 4 MB
- MaxBackupIndex: 5
- File: depends on the product layer

COMPONENT	ROLLING FILE DRIVER OUTPUT FILE DEFAULT LOCATION
Presentation Layer	`<home_dir>/var/logs/nmy_application.log`
Synchronizer	`<home_dir>/var/logs/nmy_synchronizer.log`
Connector Template	`<home_dir>/var/logs/nmy_connectortemplate.log`
Loopback Connector	`<home_dir>/var/logs/nmy_loopback.log`
Approval Sequencer	`<home_dir>/var/logs/nmy_sequencer.log`

Example of Rolling File Output Driver Settings

In this example, the logger:

- Saves logs in the `nmy_application.log` file
- Keeps a maximum number of 5 backup copies:
  - `nmy_application.log.1`
  - `nmy_application.log.2`
  - `nmy_application.log.3`
  - `nmy_application.log.4`
  - `nmy_application.log.5`
- Has a maximum file size of 4MB

```
log4j.appender.ROL=org.apache.log4j.RollingFileAppender
log4j.appender.ROL.File=!NMY_VAR_DIR!/logs/nmy_application.log
log4j.appender.ROL.MaxFileSize=4MB
log4j.appender.ROL.MaxBackupIndex=5
log4j.appender.ROL.layout=com.netonomy.util.logger.LoggerLayout
log4j.appender.ROL.layout.ConversionPattern=+{DATE_TIME};{EVENT_ID};{SEVERITY};{MODULE};{THREAD_ID};{EVENT_TYPE
};{ERROR_CODE};{DESCRIPTION}
```

About the Time Stamp Log File Output Driver

This output driver saves log information in a text file.

When using this driver, you specify the following:

- The name and location of the log file
- The interval before creating a backup file. When this interval occurs, the logger saves the file as `<log_filename>.<date>` and starts a new log file.

There is no limit to the number of backup copies of the log file.
You must manage the backup log files, as the logger does not automatically delete them.

To configure a time stamp log file driver

1. Go to the `/util` directory containing the `log4j.properties` configuration file.
2. Open `log4j.properties`.
3. Go to the `#DAILY FILE OUTPUT DRIVER SETTINGS` section.
4. Set `log4j.appender.DAY.File` to the full path and log file name.
5. Set `log4j.appender.DAY.DatePattern` to one of the following:

SETTING	DESCRIPTION	BACKUP FILE EXTENSION
`'.'yyyy-MM`	Monthly logs that begin at the start of the month	`<log_filename>.YYYY-MM`
`'.'yyyy-ww`	Weekly logs that start at the beginning of each week	`<log_filename>.YYYY-WW`
`'.'yyyy-MM-dd`	Daily logs that start at midnight	`<log_filename>.YYYY-MM-DD`
`'.'yyyy-MM-dd-h`	Daily logs that start at noon and midnight	`<log_filename>.YYYY.MM-DD-AM/PM`
`'.'yyyy-MM-dd-HH`	Hourly logs that start on the hour	`<log_filename>.YYYY-MM-DD-HH`
`'.'yyyy-MM-dd-HH-mm`	Minute logs that start every minute	`<log_filename>.YYYY-MM-DD-HH-MM`

6. Set the `log4j.appender.CON.layout.ConversionPattern` to specify the log format.
7. Save your changes.

Default Time Stamp Log File Output Settings

By default, the Time Stamp Log File output driver has the following configuration:

- Backup every day at midnight - `DatePattern='.'yyyy-MM`
- File: depends on the product layer

COMPONENT	ROLLING FILE DRIVER OUTPUT FILE DEFAULT LOCATION
Presentation Layer	`<home_dir>/var/logs/nmy_daily_application.log`
Synchronizer	`<home_dir>/var/logs/nmy_daily_synchronizer.log`
Connector Template	`<home_dir>/var/logs/nmy_daily_connectortemplate.log`
Loopback Connector	`<home_dir>/var/logs/nmy_daily_loopback.log`
Approval Sequencer	`<home_dir>/var/logs/nmy_daily_sequencer.log`

In this example, the logger:

- Saves logs in the `nmy_daily_application.log` file
- Daily log that starts at midnight

```
log4j.appender.DAY=org.apache.log4j.DailyRollingFileAppender
log4j.appender.DAY.File=!NMY_VAR_DIR!/logs/nmy_daily_application.log
log4j.appender.DAY.DatePattern='.'yyyy-MM-dd
log4j.appender.DAY.layout=com.netonomy.util.logger.LoggerLayout
log4j.appender.DAY.layout.ConversionPattern=+{DATE_TIME};{EVENT_ID};{SEVERITY};{MODULE};{THREAD_ID};{EVENT_TYPE
};{ERROR_CODE};{DESCRIPTION}
```

About the Output Format

For each driver, you configure the output format. The output format is an arrangement of the event attributes that generates a line in the output.

Each attribute is assigned a pattern you use to define the layout. This means that the format is defined with a string that is a suite of patterns and separators.

Use the syntax:

```
<Driver ConversionPattern setting> = separator
[EVENT_ATTRIBUTE_PATTERN_1] separator
[EVENT_ATTRIBUTE_PATTERN_2]…
```

Each of the format definition elements is optional. You can log the event attributes you want.

You cannot use the following characters as layout separators:

- ! (exclamation point)
- : (colon)
- - (dash)

Use the syntax:

```
log4j.appender.<OUTPUT_DRIVER_CODE>.layout.ConversionPattern=+
{EVENT_OUTPUT_PATTERN1} separator {EVENT_OUTPUT_PATTERN2}...
```

For more information about the attribute patterns, refer to *Output Patterns* in this chapter.

The logger inserts the output as a single line of text in the log file.
If you log DEBUG information, the logger places a block of information in the log between the `<DEBUG_INFO>` and `</DEBUG_INFO>` tags.
Because of this, you should place DEBUG information at the end of your output format. This keeps all of the log information together then lists any associated debug information. Otherwise you may have log information split by a block of debug information.

To specify the output format

1. Go to the `/util` directory containing the `log4j.properties` configuration file.
2. Open `log4j.properties`.

3 Under the `OUTPUT DRIVER` section, set `log4j.appender.<OUTPUT_DRIVER_CODE>.ConversionPattern` to the format to use. Use the output format syntax.

4 Save your changes.

EVENT ATTRIBUTE PATTERN	DESCRIPTION	NOTES
{DATE_TIME}	Event date and time	The format complies with ISO 8601 standard
{SEVERITY}	Event severity level	FATAL ERROR WARN INFO DEBUG
{THREAD_ID}	Event thread ID	
{SESSION_ID}	Session ID	Session ID of the user. This is used to identify the user who caused the event to be logged. This ID is the BLM session ID. When there is no BLM session, the value is NONE.
{EVENT_TYPE}	Event type	This is the type of event you can specify
{MODULE}	Event source module	AGT - Any agent – synchronizer, connector, sequencer, CID2CBU loader BLM - Business Logic Manager CUS - Custom event DAL - Data Access Layer ISF - ISF Framework and all of its sub-modules JFN - JFN Framework JSP - Java Server Pages LOG - The logger platform NIL - Unqualified event UTL - Internal utility components
{EVENT_ID}	Event ID	Unique ID for the specific output driver
{ERROR_CODE}	Event error code	
{DESCRIPTION}	Event description	

EVENT ATTRIBUTE PATTERN	DESCRIPTION	NOTES
{DEBUG_INFO}	Event Debug info	Creates a block of information between the <DEBUG_INFO> and </DEBUG_INFO> tags. This should be used at the end of your debug pattern

Default Format Settings

For product layers and drivers, the default output format is:

```
+{DATE_TIME};{EVENT_ID};{SEVERITY};{MODULE};{THREAD_ID};{EVENT_T
YPE};{ERROR_CODE};{DESCRIPTION}
```

For the Presentation Layer, the default output format is:

```
+{DATE_TIME};{EVENT_ID};{SESSION_ID};{SEVERITY};{MODULE};{THREAD
_ID};{EVENT_TYPE};{ERROR_CODE};{DESCRIPTION}
```

Examples of Logger Files

The following examples show the logger configuration files and an example of a logger message.

Example of `logger.properties`

Location of the parameter file	logger.log4j.properties=log4j.properties
Frequency of parameter file reload	logger.log4j.reloadDelay=0

Example of `log4j.properties`

Internal settings
DO NOT MODIFY
	# DO NOT MODIFY
	log4j.categoryFactory=com.netonomy.util.logger.LoggerCategoryFactory
	log4j.rootCategory=INFO,DISABLED

Types of events to log
All of the events are logged in the ROL log file
Exceptions are logged in a DAY log file
Unqualified are displayed in the system console
	# EVENTS
	log4j.category.nmy.INIT=INFO,0,ROL
	log4j.category.nmy.STATE=INFO,0,ROL
	log4j.category.nmy.EXCEPTION=INFO,0,ROL, DAY
	log4j.category.nmy.SESSION=INFO,0,ROL
	log4j.category.nmy.REQUEST=INFO,0,ROL
	log4j.category.nmy.MESSAGE=INFO,0,ROL
	log4j.category.nmy.OBJECT=INFO,0,ROL
	log4j.category.nmy.RESOURCE=INFO,0,ROL
	log4j.category.nmy.DATA=INFO,0,ROL
	log4j.category.nmy.NONE=INFO,0,ROL,CON

Internal settings
DO NOT MODIFY
	# DO NOT MODIFY
	log4j.appender.DISABLED=com.netonomy.util.logger.NullAppender

Standard Output Settings	# STANDARD OUTPUT DRIVER SETTINGS
	log4j.appender.CON=org.apache.log4j.ConsoleAppender
	log4j.appender.CON.Target=System.out
	log4j.appender.CON.layout=com.netonomy.util.logger.LoggerLayout
	log4j.appender.CON.layout.ConversionPattern=+{DATE_TIME};{EVENT_ID};{SESSION_ID};{SEVERITY};{MODULE};{THREAD_ID};{EVENT_TYPE};{ERROR_CODE};{DESCRIPTION}

Rolling FIle Output Settings	# ROLLING FILE OUTPUT DRIVER SETTINGS
	log4j.appender.ROL=org.apache.log4j.RollingFileAppender
	log4j.appender.ROL.File=!NMY_VAR_DIR!/logs/nmy.log
	log4j.appender.ROL.MaxFileSize=4MB
	log4j.appender.ROL.MaxBackupIndex=5
	log4j.appender.ROL.layout=com.netonomy.util.logger.LoggerLayout
	log4j.appender.ROL.layout.ConversionPattern=+{DATE_TIME};{EVENT_ID};{SESSION_ID};{SEVERITY};{MODULE};{THREAD_ID};{EVENT_TYPE};{ERROR_CODE};{DESCRIPTION}

Daily File Output Settings	# DAILY FILE OUTPUT DRIVER SETTINGS
	log4j.appender.DAY=org.apache.log4j.DailyRollingFileAppender
	log4j.appender.DAY.File=!NMY_VAR_DIR!/logs/nmy_daily_log.log
	log4j.appender.DAY.DatePattern='.'yyyy-MM-dd
	log4j.appender.DAY.layout=com.netonomy.util.logger.LoggerLayout
	log4j.appender.DAY.layout.ConversionPattern=+{DATE_TIME};{EVENT_ID};{SESSION_ID};{SEVERITY};{MODULE};{THREAD_ID};{EVENT_TYPE};{ERROR_CODE};{DESCRIPTION}

This example shows the initialization log.

```
+2002-03-06 17:39:23.830;EC6C8259B6ExecuteThread-310;INFO;LOG;ExecuteThread-31;INIT;0012000;Setting logger
configuration file to [path="classes/nmycfg/util/log4j.properties"].


+2002-03-06 17:39:23.850;EC6C8259B6ExecuteThread-311;INFO;LOG;ExecuteThread-31;INIT;0012001;Set dynamic reload
of logger configuration every [frequency=30] seconds.


+2002-03-06 17:39:29.889;EC6C8259B6ExecuteThread-312;INFO;DAL;ExecuteThread-31;INIT;0011001;Initializing
datasource [name="cidDatasource"], [driver="JNDI t3://localhost:7001"], [user="N/A"] succeeded.


+2002-03-06 17:39:52.232;EC6C8259B6ExecuteThread-313;INFO;DAL;ExecuteThread-31;INIT;0010101;Loading component
configuration [name="DAL"] succeeded.


+2002-03-06 17:40:18.760;EC6C8259B6ExecuteThread-314;INFO;BLM;ExecuteThread-31;INIT;0010101;Loading component
configuration [name="BLM"] succeeded.


+2002-03-06 17:40:21.635;EC6C8259B6ExecuteThread-315;INFO;JFN;ExecuteThread-31;INIT;0040001;Loading media
application file [path="<home_dir>/channels/WEB-INF/classes/nmycfg/jfn/MyWeb.xml"] succeeded.


+2002-03-06 17:40:22.255;EC6C8259B6ExecuteThread-316;WARN;BLM;ExecuteThread-31;STATE;0022102;Checking the
validity of the reference data cache. Cache out of date and cleared.
```

This example shows the log when an error occurs when loading a required file.

```
+2003-01-20 18:18:35.261;F2DC8E7ABEExecuteThread: '12' for queue: 'default'63;1060;INFO;BLM;ExecuteThread:
'12' for queue: 'default';INIT;0010101;Loading component configuration [name="nmycfg.blm.config"] succeeded.

+2003-01-20 18:21:21.568;F2DC8E7ABEExecuteThread: '12' for queue: 'default'64;1060;ERROR;DAL;ExecuteThread:
'12' for queue: 'default';DATA;0011404;Executing SQL statement failed [execute type="update"]: [DB error="1"].

+2003-01-20 18:25:40.728;F2DC9FB90CExecuteThread: '12' for queue: 'default'0;NONE;INFO;LOG;ExecuteThread: '12'
for queue: 'default';INIT;0012000;Setting logger configuration file to [path="nmycfg/util/log4j.properties"].

+2003-01-20 18:25:40.768;F2DC9FB90CExecuteThread: '12' for queue: 'default'1;NONE;INFO;LOG;ExecuteThread: '12'
for queue: 'default';INIT;0012002;Dynamic reload of logger configuration is not activated.

+2003-01-20 18:25:48.429;F2DC9FB90CExecuteThread: '12' for queue: 'default'2;NONE;FATAL;DAL;ExecuteThread:
'12' for queue: 'default';INIT;0011002;Initializing datasource [name="jdbc/cidDatasource"], [driver="JNDI "],
[user="N/A"] failed.

+2003-01-20 18:25:48.650;F2DC9FB90CExecuteThread: '12' for queue: 'default'3;NONE;FATAL;DAL;ExecuteThread:
'12' for queue: 'default';INIT;0030105;Instantiating Data Access Layer driver
[instance="nmycfg.dal.instances.instance_route"],
[driver="com.netonomy.dal.drivers.impl.sql.jndi.JNDIDatasourceInstance"] failed.

+2003-01-20 18:25:48.650;F2DC9FB90CExecuteThread: '12' for queue: 'default'4;NONE;FATAL;DAL;ExecuteThread:
'12' for queue: 'default';INIT;0010002;Initializing component [name="DAL Authentication Module"] failed.
```

Using Logs with a Supervision Platform

When using a supervision platform to manage or analyze logs, you can do the following to maximize the performance and quality of information in the logs:

- Adapt the output format for your platform
- Use certain event attributes
- Take advantage of event codes

Adapting the Output Format

Most supervision platforms offer the possibility to parse text files with pattern matching.

This feature is based on the following:

- In the generated file, there is one event per line
- The line structure is known

You can adapt the output format of log files to make pattern matching easier.

As this is also based on the principle that there is one event per line, you should not log `DEBUG_INFO`.

The default output format settings follow these recommendations.

Using Event Attributes

Some event attributes are very helpful to supervise:

- Event Id

  Because the ID is unique, you can use this information as a reference when the alert is to be forwarded to tracking systems.
- Session Id

  This ID allows you to track events which are generated by the same user. This way you can pinpoint and follow the events that are logged. If there is no associated user session id, you can still track the sequence of logged events by using the thread id.

- Event codes

Using Event Codes

When using event codes, the main attribute to check is severity. This is because severity gives you an idea as to the impact of the event on the application. But the real impact depends on your application. As most events are coded, you can use this to your advantage when checking logs.

For event codes:

- Each code is associated with a description (the event description attribute value).
- In some descriptions, there are some fixed parts (the description parameters).

Description parameters refer to the dynamic part of the description. But they are always marked with a fixed and documented tag. This part of the event code is the code interface and will not change even if the description does.

An example:

Your deployment relies on a file system that is extremely sensitive and you want to monitor events that deal with your application interacting with this file system.

There are some event codes that are dedicated to file system interaction.

For instance, the code 2004 is associated with the event description "`Cannot write to file [path="<the path>"].`"

In this case, the section `[path=""]` is fixed.

This means you can:

- Monitor logging of event code '2004'.
- If the event occurs, check the 'path' description parameter.

If the path that appears just after the `<[path=">` tag, you can generate a critical alert for your supervision system and have people work on the issue quickly.

For more information about the standard codes, refer to the *Logger Message Reference Documentation*.

Filtering by Source Component

You can also configure the logger to filter events by component. You can use this feature to help application development and debugging as you focus on event source components.

If you have customized a part of the application and want to check logs for testing or debugging, you can focus on the component you work with, activate full debug information, without getting a huge amount of events from components you are not interested in.

You filter events by specifying event types. In addition to event types, you can filter logged events by using the event component parameter.

You use the syntax:

```
<Event Type>.<Event Source> = <Severity Level>, [Debug Level],
<Output Driver #1>, <Output Driver #2>,...
```

For example, `log4j.category.nmy.INIT.BLM=INFO,0,ROL`

- Reduces the logs for INIT-type events to the ones issued from the BLM.
- Logs events using the Rolling File output, and there is no DEBUG event generated.

The event modules you can filter on are:

EVENT COMPONENT	DESCRIPTION
ADM	Administration
AGT	Any agent – synchronizer, connector, sequencer, CID2CBU loader
BLM	Business Logic Manager
CUS	Custom event
DAL	Data Access Layer
ISF	ISF Framework and all of its sub-modules
JFN	JFN Framework
JSP	Java Server Pages
LOG	The logger platform
NIL	Unqualified event
QRA	Query and Report Analysis
UTL	Internal utility components
WFS	Web File System

Examples of Filtering by Component

1 Get all types of event but only ones generated by the BLM – no debug information, rolling file

```
# EVENTS
log4j.category.nmy.INIT.BLM=INFO,0,ROL
log4j.category.nmy.STATE.BLM=INFO,0,ROL
log4j.category.nmy.EXCEPTION.BLM=INFO,0,ROL
log4j.category.nmy.SESSION.BLM=INFO,0,ROL
log4j.category.nmy.REQUEST.BLM=INFO,0,ROL
log4j.category.nmy.MESSAGE.BLM=INFO,0,ROL
log4j.category.nmy.OBJECT.BLM=INFO,0,ROL
log4j.category.nmy.DATA.BLM=INFO,0,ROL
log4j.category.nmy.NONE.BLM=INFO,0,ROL
...
```

2 Get only 'INIT' events, but block the ones generated by the DAL – debug level 3, rolling file

```
# EVENTS
log4j.category.nmy.INIT.BLM=INFO,0,ROL
log4j.category.nmy.INIT.JFN=INFO,0,ROL
log4j.category.nmy.INIT.JSP=INFO,0,ROL
log4j.category.nmy.INIT.AGT=INFO,0,ROL
log4j.category.nmy.INIT.ISF=INFO,0,ROL
log4j.category.nmy.INIT.LOG=INFO,0,ROL
log4j.category.nmy.INIT.UTL=INFO,0,ROL

log4j.category.nmy.STATE.BLM=INFO,DISABLED
log4j.category.nmy.EXCEPTION.BLM=INFO,DISABLED
log4j.category.nmy.SESSION.BLM=INFO,DISABLED
log4j.category.nmy.REQUEST.BLM=INFO,DISABLED
log4j.category.nmy.MESSAGE.BLM=INFO,DISABLED
log4j.category.nmy.OBJECT.BLM=INFO,DISABLED
log4j.category.nmy.DATA.BLM=INFO,DISABLED
log4j.category.nmy.NONE.BLM=INFO,DISABLED
...
```

3 Get all types of event, but block events generated by the JFN and the JSP – no debug information, rolling file

```
# EVENTS
log4j.category.nmy.INIT.BLM=INFO,0,ROL
log4j.category.nmy.INIT.DAL=INFO,0,ROL
log4j.category.nmy.INIT.AGT=INFO,0,ROL
log4j.category.nmy.INIT.ISF=INFO,0,ROL
log4j.category.nmy.INIT.LOG=INFO,0,ROL
log4j.category.nmy.INIT.UTL=INFO,0,ROL

log4j.category.nmy.STATE.BLM=INFO,0,ROL
log4j.category.nmy.STATE.DAL=INFO,0,ROL
log4j.category.nmy.STATE.AGT=INFO,0,ROL
log4j.category.nmy.STATE.ISF=INFO,0,ROL
log4j.category.nmy.STATE.LOG=INFO,0,ROL
log4j.category.nmy.STATE.UTL=INFO,0,ROL

log4j.category.nmy.EXCEPTION.BLM=INFO,0,ROL
log4j.category.nmy.EXCEPTION.DAL=INFO,0,ROL
log4j.category.nmy.EXCEPTION.AGT=INFO,0,ROL
log4j.category.nmy.EXCEPTION.ISF=INFO,0,ROL
log4j.category.nmy.EXCEPTION.LOG=INFO,0,ROL
log4j.category.nmy.EXCEPTION.UTL=INFO,0,ROL
...
```

Note:

- You have to define filtering for all couples (Event Type, Event Module) that you are interested in.
- Blocking JFN and JSP is achieved by not mentioning the modules at all.

APPENDIX A

Administration Tool Reference

In This Section

CBU Administration Tool

You use the `cbuAdminTool` to manage the CBU.

This administration tool is located in `<home_dir>/bin`.

For very large databases, this tool comes with a datablock parameter for commands that manage tables containing large amounts of data (requests, trouble tickets, and so on.) This parameter lets you set the number of records to process before the tool commits changes. This way, you can handle large amounts of data in smaller blocks of information that are easier to manage and help avoid various issues while interacting with the database. You can also override this setting from the command line when administrating the CBU.

However, by limiting the number of items the tool may have to submit several changes to the database. For instance, if your table contains 3,000 records to purge and your default datablock size is 1,000, the tool commits 3 changes. If an error occurs during this process, there is no way to carry out a global rollback of your changes.

Configuring the CBU Administration Tool

After installation, you can change the settings of the CBU administration tool.

You use the `cbu_tools.properties` configuration file to set the properties of the CBU administration tool. This file is located in `<home_dir>/lib/admin/cbu`.

Configuring the CBU Administration tool involves:

- Specifying the JDBC driver
- Specifying the database URL
- Specifying the encoding
- Specifying the data block size

To modify the settings of the CBU administration tool

1. Go to `<home_dir>/lib/admin/cbu`.
2. Open `cbu_tools.properties`.
3. Change the following settings:

SETTING	DESCRIPTION
DRIVER	the class name of the ODBC driver to use
URL	the URL of the CBU database
ENCODING	the encoding to use
DATABLOCKSIZE	Default number of items to handle before submitting the data. The CBU administration tool processes the data in sets corresponding to the specified size until done. This value can be overridden when running the CBU administration tool.

4 Save your changes.

Example of cbu_tool.properties

JDBC driver	`# driver: This is the class name of the jdbc driver (typically oracle.jdbc.driver.OracleDriver)`
	`driver=oracle.jdbc.driver.OracleDriver`

URL of the database	`# url: This is url for the database (typically jdbc:oracle:oci8:@OracleCidName)`
	`url=jdbc:oracle:oci8:@CBU`

Internal settings
DO NOT MODIFY
If you need to modify these settings, you must reinstall Telco Analytics Manager and enter the new values during installation.
	`# SQL Base Path`
	`SqlBasePath=<home_dir>/data/cbu/oracle/`
	`# LOG Path`
	`LogPath=<home_dir>/var/logs/cbu_adm/`
	`# OUTPUT PATH`
	`OutPath=<home_dir>/var/output/cbu_adm/`
	`# VERSION STRING`
	`Version=3.1.0.407`

Encoding	`# ENCODING`
	`Encoding=ISO-8859-1`

Data Block Size	`# DATABLOCKSIZE`
	`DataBlockSize=10000`

cbuAdminTool Syntax

```
cbuAdminTool help | <command> <parameters> [<optional
parameters>] <help>
```

PARAMETERS	DESCRIPTION
help	Displays help for the tool
<command>	Administration command
<parameters>	Specifies the parameters for the command
[<optional parameters>]	Optional parameters
<command> help	Displays help for the command

COMMAND	DESCRIPTION	PARAMETERS
create_cbu_users	Creates the default CBU_ADM and CBU_USR users and their corresponding roles	<CBU> Oracle: <instance alias> - SQL Server: <database host> [:<port>];DatabaseName="<db_name>" If no port is specified, the tool uses the default SQL server port For DB2, the CID users are operating system users and cannot be created using this tool. <login>DBA Login ID <password>DBA Login password <CBU_ADMIN login> login for CBU admin account <CBU_ADMIN password>: password for CBU admin account <CBU_USER login> login for CBU user account <CBU_USER password> password for CBU user account [-quiet] Do not display information
install_cbu	Creates the CBU database for production and populates reference tables	<CBU> - Oracle: <instance alias> - DB2: <database alias> - SQL Server: <database host> [:<port>] If no port is specified, the tool uses the default SQL server port <CBU_ADMIN login> login for CBU admin account <CBU_ADMIN password>: password for CBU admin account <CBU_USER login> login for CBU user account <CBU_USER password> password for CBU user account [-quiet] Do not display information

COMMAND	DESCRIPTION	PARAMETERS
create_cbu_structure	Creates a demo CBU database structure	<CBU> • Oracle: <instance alias> • DB2: <database alias> • SQL Server: <database host> [:<port>] If no port is specified, the tool uses the default SQL server port <login>DBA Login ID <password>DBA Login password <CBU_ADMIN login> login for CBU admin account <CBU_ADMIN password>: password for CBU admin account <CBU_USER login> login for CBU user account <CBU_USER password> password for CBU user account [-quiet] Do not display information

CID Administration Tool

You use the `cidAdminTool` to manage the CID.

This administration tool is located in `<home_dir>/bin`.

For very large databases, this tool comes with a datablock parameter for commands that manage tables containing large amounts of data (requests, trouble tickets, and so on.) This parameter lets you set the number of records to process before the tool commits changes. This way, you can handle large amounts of data in smaller blocks of information that are easier to manage and help avoid various issues while interacting with the database. You can also override this setting from the command line when administrating the CID.

However, by limiting the number of items the tool may have to submit several changes to the database. For instance, if your `REQUEST` table contains 3,000 requests to purge and your default datablock size is 1,000, the tool commits 3 changes. If an error occurs during this process, there is no way to carry out a global rollback of your changes.

Configuring the CID Administration Tool

After installation, you can change the settings of the CID administration tool.

You use the `cid_tools.properties` configuration file to set the properties of the CID administration tool. This file is located in `<home_dir>/lib/admin/cid`.

Configuring the CID Administration tool involves:

- Specifying the JDBC driver
- Specifying the database URL
- Specifying the encoding
- Specifying the data block size

When configuring the CID Administration tool, keep in mind that you can override the data block size when running the tool.

To modify the settings of the CID administration tool

1. Go to `<home_dir>/lib/admin/cid`.
2. Open `cid_tools.properties`.
3. Change the following settings:

SETTING	DESCRIPTION
DRIVER	Class name of the ODBC driver to use
URL	URL of the CID database
ENCODING	Encoding to use
DATABLOCKSIZE	Default number of items to handle before submitting the data. The CID administration tool processes the data in sets corresponding to the specified size until done. This value can be overridden when running the CID administration tool.

4 Save your changes.

Example of cid_tool.properties for Oracle

JDBC driver	# driver: This is the class name of the jdbc driver
	driver=oracle.jdbc.driver.OracleDriverdriver=oracle.jdbc.driver.OracleDriver

URL of the database	# url:This is url for the database (typically jdbc:oracle:oci8:@OracleCidName)
	url=jdbc:oracle:oci8:@CID

Internal settings
DO NOT MODIFY
If you need to modify these settings, you must reinstall and enter the new values during installation.
	# SQL Base Path
	SqlBasePath=<home_dir>/data/oracle/
	# LOG Path
	LogPath=<home_dir>/var/logs/admin/
	# OUTPUT PATH
	OutPath=<home_dir>/var/output/admin/
	# VERSION STRING
	Version=X.X.X.XXX

Encoding	# ENCODING
	Encoding=ISO-8859-1

Data Block Size	# DATABLOCKSIZE
	DataBlockSize=10000

cidAdminTool Syntax

```
cidAdminTool help | <command> <parameters> [<optional
parameters>] <help>
```

PARAMETERS	DESCRIPTION
help	Displays help for the tool
<command>	Administration command
<parameters>	Specifies the parameters for the command
[<optional parameters>]	Optional parameters
<command> help	Displays help for the command

COMMAND	DESCRIPTION	PARAMETERS
admin_notifications	Manages notifications stored in the CID notification queue	<CID> - Oracle: <instance alias> - DB2: <database alias> - SQL Server: <database host> [:<port>] If no port is specified, the tool uses the default SQL server port <CID_ADMIN login> login for the CID admin account <CID_ADMIN password> password for the CID admin account [-quiet] Do not display information [<datablocksize:####>] specifies the number of items to process before committing. Enter 0 to override default settings.
admin_requests	Manages requests stored in the CID request queue	<CID> - Oracle: <instance alias> - DB2: <database alias> - SQL Server: <database host> [:<port>] If no port is specified, the tool uses the default SQL server port <CID_ADMIN login> login for the CID admin account <CID_ADMIN password> password for the CID admin account [-quiet] Do not display information [<datablocksize:####>] specifies the number of items to process before committing. Enter 0 to override default settings.

COMMAND	DESCRIPTION	PARAMETERS
create_cid_users	Creates the CID_ADM and CID_USR users and their corresponding roles	<CID> • Oracle: <instance alias> • SQL Server: <database host> [:<port>];DatabaseName="<db_name>" If no port is specified, the tool uses the default SQL server port For SQL Server, note that the semicolon is a special character of UNIX and Windows shell. The CID parameter should be enclosed by quotation marks. For DB2, the CID users are system users and cannot be created using this tool. <login> DBA Login ID <password> DBA Login password < new CID_ADMIN login> login for the CID admin account <new CID_ADMIN password> password for the CID admin account <new CID_USER login> login for the CID user account <new CID_USER password> password for the CID user account [<quiet>] Do not display information
create_demo_cid_structure	Creates a CID demo database structure with reference data, but without the demo dataset	<CID> • Oracle: <instance alias> • DB2: <database alias> • SQL Server: <database host> [:<port>] If no port is specified, the tool uses the default SQL server port <CID_ADMIN login> login for the CID admin account <CID_ADMIN password> password for the CID admin account [-quiet] Do not display information

COMMAND	DESCRIPTION	PARAMETERS
create_demo_cid_test	Creates a CID demo database structure with reference data, and populates it with the demo dataset	<CID> • Oracle: <instance alias> • DB2: <database alias> • SQL Server: <database host> [:<port>] If no port is specified, the tool uses the default SQL server port <CID_ADMIN login> login for the CID admin account <CID_ADMIN password> password for the CID admin account [-quiet] Do not display information
drop_demo_cid_structure	Drops a CID demo database structure along with the populated data	<CID> • Oracle: <instance alias> • DB2: <database alias> • SQL Server: <database host> [:<port>] If no port is specified, the tool uses the default SQL server port <CID_ADMIN login> login for the CID admin account <CID_ADMIN password> password for the CID admin account [-quiet] Do not display information
generate_logon_stats	Extracts logon statistics from CID to a file	<CID> • Oracle: <instance alias> • DB2: <database alias> • SQL Server: <database host> [:<port>] If no port is specified, the tool uses the default SQL server port <CID_ADMIN login> login for the CID admin account <CID_ADMIN password> password for the CID admin account <m|q|y> m generate stats for the last completed month q generate stats for the last completed quarter y generate stats for the last completed year [-quiet] Do not display information

COMMAND	DESCRIPTION	PARAMETERS
install_cid	Creates the CID database structure and populates the required reference tables	<CID> • Oracle: <instance alias> • DB2: <database alias> • SQL Server: <database host> [:<port>] If no port is specified, the tool uses the default SQL server port <CID_ADMIN login> login for the CID admin account <CID_ADMIN password> password for the CID admin account <CID_USER login> login for the CID user account <CID_USER password> password for the CID user account [-quiet] Do not display information
purge_requests	Purges the requests stored in the CID request queue	<CID> • Oracle: <instance alias> • DB2: <database alias> • SQL Server: <database host> [:<port>] If no port is specified, the tool uses the default SQL server port <CID_ADMIN login> login for the CID admin account <CID_ADMIN password> password for the CID admin account <days> The age of the requests to keep in days. For example, entering 7 purges all of the requests that were not created during the last week. [<status>,<status>...] The status of the requests to purge [-quiet] Do not display information [<datablocksize:####>] specifies the number of items to process before committing. Enter 0 to override default settings.

COMMAND	DESCRIPTION	PARAMETERS
purge_persistent_action_managers	Purges the persistent action managers (shopping carts) stored in the CID	<CID> • Oracle: <instance alias> • DB2: <database alias> • SQL Server: <database host> [:<port>] If no port is specified, the tool uses the default SQL server port <CID_ADMIN login> login for the CID admin account <CID_ADMIN password> password for the CID admin account <days> The age of the persistent action managers to keep in days. For example, entering 7 purges all of the persistent action managers that were not created during the last week. <category> The category of persistent action managers to purge. By default: • 1 backup shopping carts • 2 contract templates [-quiet] Do not display information [<datablocksize:####>] specifies the number of items to process before committing. Enter 0 to override default settings.
purge_trouble_tickets	Purges the trouble tickets stored in the CID	<CID> • Oracle: <instance alias> • DB2: <database alias> • SQL Server: <database host> [:<port>] If no port is specified, the tool uses the default SQL server port <CID_ADMIN login> login for the CID admin account <CID_ADMIN password> password for the CID admin account <days> The age of the trouble tickets to purge in days [<status>,<status>...] The status of the trouble tickets to purge [-quiet] Do not display information [<datablocksize:####>] specifies the number of items to process before committing. Enter 0 to override default settings.

COMMAND	DESCRIPTION	PARAMETERS
purge_user_events	Purges the user events stored in the CID	<CID> ▪ Oracle: <instance alias> ▪ DB2: <database alias> ▪ SQL Server: <database host> [:<port>] If no port is specified, the tool uses the default SQL server port <CID_ADMIN login> login for the CID admin account <CID_ADMIN password> password for the CID admin account <days> The age of the events to keep in days. For example, entering 7 purges all of the events that were not created during the last week. [<event_code>[,<event_code>...] The code of the type(s) of events to purge. If none entered, all events are purged. The event code corresponds to the the USER_TYPE_EVENT_ID in the USER_TYPE_EVENT table. [-quiet] Do not display information [<datablocksize:####>] specifies the number of items to process before committing. Enter 0 to override default settings.
purge_notifications	Purges the notifications stored in the CID notification queue	<CID> • Oracle: <instance alias> • DB2: <database alias> • SQL Server: <database host> [:<port>] If no port is specified, the tool uses the default SQL server port <CID_ADMIN login> login for the CID admin account <CID_ADMIN password> password for the CID admin account <days> The age of the notifications to keep in days. For example, entering 7 purges all of the notifications that were not created during the last week. [<status>,<status>...] The status of the notifications to purge. If none specified, the tool purges FAILED and DONE notifications. [-quiet] Do not display information [<datablocksize:####>] specifies the number of items to process before committing. Enter 0 to override default settings.

NOTIFYORG Administration Tool

You use the `notifyorg` tool to force the synchronization of organization information in the CBU.

This administration tool is located in `<home_dir>/bin`.

Configuring the NOTIFYORG Administration Tool

After installation, you can change the settings of the notifyorg administration tool.

You use the `notifyorg.properties` configuration file to set the properties of the administration tool. This file is located in `<home_dir>/config/notifyorg/nmycfg/notifyorg`.

Configuring the notifyorg Administration tool involves:

- Specifying the Agent Name and password
- Specifying the data block size

To modify the settings of the NOTIFYORG administration tool

1 Go to `<home_dir>/config/notifyorg/nmycfg/notifyorg`.

2 Open `notifyorg.properties`.

3 Change the following settings:

SETTING	DESCRIPTION
AgentName	The name of the agent to connect to the BLM.
AgentPassword	The corresponding password
DATABLOCKSIZE	Default number of items to handle before submitting the data. The NOTIFYORG administration tool processes the data in sets corresponding to the specified size until done. This value can be overridden when running the administration tool.

4 Save your changes.

Example of notifyorg.properties

Agent Name and password	`AgentName=agent`
	`AgentPassword=agent`

Data Block Size	`# DATABLOCKSIZE`
	`DataBlockSize=1000`

notifyorg Syntax

```
notifyorg [<optional parameters>]
```

PARAMETERS	DESCRIPTION
help	Displays help for the tool
[<optional parameters>]	Optional parameters

COMMAND	DESCRIPTION	PARAMETERS
notifyorg	Processes the synchronization of organizations	[<datablocksize:####>] specifies the number of items to process before committing. Enter 0 to override default settings.

CID2CBU Loader Administration Tools

You use the following administration tools to administrate the CID2CBU Loader:

- `cid2cbuloader`
- `cid2cbuloaderadm`

These administration tools are located in `<home_dir>/bin`.

Configuring the CID2CBU Loader

The CID2CBU loader acts as a polling process that extracts notifications from the CID and updates the CBU database.

Configuring the CID2CBU loader involves:

- Setting the CID2CBU properties
- Optimizing the CID2CBU properties when required
- Setting the CID2CBU database connection properties

Setting the CID2CBU Loader Properties

You can define several CID2CBU loaders.

The CID2CBU loader uses the `cid2cbuloader.properties` configuration file to set its properties. This file is located in `<home_dir>/config/cid2cbuloader`.

For each CID2CBU loader, you configure a `cid2cbuloader.properties` configuration file.

You can configure the following:

- The user name and password to authenticate the CID2CBU loader to the BLM
- The BLM connection retry mechanism
- The CBU Database connection retry mechanism
- The number of notifications to extract from the notification queue
- The stability delay of a notification before extracting it
- The list of notification types-object types to extract
- The average number of notifications selected still to treat (Queue threshold) for one thread before selecting new ones.
- The sleep time between two executions of the notification selection when the first one returns nothing to treat.
- The sleep time between two verifications of the Queue threshold before selecting new notifications
- The running mode of the CID2CBU loader (one shot or not)
- The number of threads to allocate to process notifications
- The configuration directory path
- The administration port

To configure the CID2CBU loader

1. Go to `<home_dir>/config/cid2cbuloader`.
2. Open `cid2cbuloader.properties`.
3. Enter the following:

PARAMETER	DESCRIPTION
CONFIG_DIR	Full path of the directory containing the cid2cbuloader.properties configuration file
ADM_PORT	Port number you use to administer the CID2CBU loader (this parameter is set by the installer)
NAPPING_TIME	Milliseconds between two executions of the notification selection when the first one returns nothing to process
LOOPING_TIME	Milliseconds between two verifications of the queue threshold before selecting new notifications

PARAMETER	DESCRIPTION
Q_THRESHOLD	Average number of notifications selected still to process for one thread before selecting new notifications
BLM_USER	User login to authenticate the CID2CBU loader for the BLM
BLM_PASSWORD	Associated password
BLM_RECONNECT_RETRIES	Number of times the CID2CBU loader tries to reconnect to the BLM
BLM_RECONNECT_DELAY	Milliseconds between two BLM connection retries
CBU_RECONNECT_RETRIES	Number of times the CID2CBU loader tries to reconnect to the CBU
CBU_RECONNECT_DELAY	Milliseconds between two CBU connection retries
NB_NOTIFICATIONS	Number of notifications to extract from the notification queue
NOTIFICATION_STABILITY_DELAY	Milliseconds between a notification creation date-time and current date time to extract it.
NB_THREADS	Number of threads allocated to process notifications
ONE_SHOT	TRUE for one shot mode. This mode is usually for loading an empty CBU. FALSE for agent mode
NOTIFICATION_FILTER	TRUE to process the notification type-object type couple. FALSE to ignore Example: `{"CREATE",MEMBER,true}, {"MODIFY",MEMBER,false}` The CREATE notification event of a MEMBER is processed by the CID2CBU loader. The MODIFY notification event of a MEMBER is ignored.

4 Save your changes.

Example of cid2cbuloader.properties

CID2CBU Loader parameters	`CONFIG_DIR="file:///<home_dir>/config/cid2cbuloader/"`
	`ADM_PORT=3004`
	
	`NAPPING_TIME=1000`
	`LOOPING_TIME=500`
	`NB_THREADS=4`

BLM connection parameters	`BLM_USER="agent"`
	`BLM_PASSWORD="agent"`
	`BLM_RECONNECT_DELAY=60000`
	`BLM_RECONNECT_RETRIES=30`

CBU connection parameters	`CBU_RECONNECT_DELAY=60000`
	`CBU_RECONNECT_RETRIES=30`

Queue parameters	Q_THRESHOLD=2
	NB_NOTIFICATIONS=1000
	NOTIFICATION_STABILITY_DELAY=5000
	NOTIFICATION_FILTER=1

Mode	ONE_SHOT=false
Notification filter for notification type-object type couples	NOTIFICATION_FILTER={{"CREATE",ORGANIZATION,true},{"MODIFY",ORGANIZATION,true},{"UPDATE",ORGANIZATION,true}, \ {"CREATE",MEMBER,true},{"MODIFY",MEMBER,true}, \ {"CREATE",CONTRACT,true},{"MODIFY",CONTRACT,true}, {"CREATE",CONTACT,true},{"MODIFY",CONTACT,true}, \ {"CREATE",BILLINGACCOUNT,true},{"MODIFY",BILLINGACCOUNT,true}, \ {"CREATE",LOGIN,true},{"MODIFY",LOGIN,true}, \ {"CREATE",ORGVIEW,true},{"MODIFY",ORGVIEW,true},{"REMOVE",ORGVIEW,true} }

Configuring the CID2CBU Loader Database Connection

You can define the CID2CBU loader database connection parameters. By default, you enter the CID2CBU connection parameters for the CID and CBU during installation. However, you may need to change the connection parameters after installation.

The CID2CBU loader uses the following configuration files:

- `instance_route.properties` configuration file to access the CID.
- `cbu_instance.properties` configuration file to access the CBU.

These files are located in `<home_dir>/config/cid2cbuloader/nmycfg/dal/instances`.

To configure the database connection

1. Go to `<home_dir>/config/cid2cbuloader/nmycfg/cid2cbu`.
2. Do one of the following:
   - To change the connection to the CID, open `instance_route.properties`.
   - To change the connection to the CBU, open `cbu_instance.properties`
3. Enter the following:
   - `DRIVER`: enter the name of the driver to use
   - `URL`: enter the location of the database
   - `USER:` enter the login
   - `PASSWORD`: enter the password
4. Save your changes.

Optimizing the CID2CBU loader

The default CID2CBU loader settings may not correspond exactly to your system architecture and resources. When not optimized for your environment, the CID2CBU loader may not respond to your performance targets for data synchronization.

To optimize the performances of the CID2CBU loader, you have to change the following settings for the Java Virtual Machine (JVM):

- Number of threads
- Allocated memory

By changing these settings, you ensure that the CID2CBU loader is running at optimal performance for your system.

Optimizing the CID2CBU loader involves:

- Determining the system resources available for the CID2CBU loader
- Specifying the number of threads in the `cid2cbuloader.properties` configuration file
- Specifying the allocated memory in the CID2CBU loader administration tool

To optimize the CID2CBU

1 Determine the number of processors dedicated to running the CID2CBU loader. The number of processors is referred to as `N`.

2 Determine the maximum amount of RAM to use per dedicated processor. The maximum available RAM is referred to as `MR`. By definition, `MR` =`Total RAM/N`, although this may be different for your environment.

3 Specify the number of threads per processor. Do the following:

1. Go to `<home_dir>/config/cid2cbuloader`.
2. Open `cid2cbuloader.properties`.
3. Set the `NB_THREADS` setting to the number of threads the CID2CBU loader uses.

The number of threads should be limited to 1 or 2 per processor.

4 Determine the minimum and maximum memory allocated to the JVM. Use the following formulas to determine the values of the JVM command line arguments:

JVM Arguments:

- `MaxJVMSize = (MR)x(N)`
- `MinJVMSize = MaxJVMSize`

If the MaxJVMSize is greater than the maximum allowed memory for a process, you can run more than one instance of the CID2CBU loader. The maximum amount depends on your system. For instance, the maximum available RAM for Solaris 8 is 4GB. If you run more than one CID2CBU loader, you need to install it and specify a different administration port and configure it by following these instructions.

If the minimum amount of RAM is not equal to the maximum, enter the minimum amount of guaranteed RAM. By entering the amount of guaranteed RAM, you can prevent problems when the process or system are restarted.

New Object Size Arguments:

- MaxNewSize=MaxJVMSize/3
- NewSize=MaxJVMSize/3

5 Enter the JVM command line arguments to the CID2CBU loader administration tool. Do the following:

1. Go to <home_dir>/bin.
2. Open the cid2cbuloader file.
3. Change the following Java command line arguments:

DEFAULT	NEW
-ms64m	-XmsMaxJVMSizem
-mx128m	-XmxMinJVMSizem

4. Add the following Java command line arguments:

```
-XX:MaxNewSize=MaxNewObjectSizem
-XX:NewSize=MaxNewObjectSizem
```

Your command line should look like this:

```
$NMY_JAVA_HOME/bin/java -XmsMaxJVMSizem -XmxMinJVMSizem -
XX:MaxNewSize=MaxNewObjectSizem -XX:NewSize=MaxNewObjectSizem
...
```

cid2cbuloader Syntax

```
cid2cbuloader <connector_name>
```

PARAMETERS	DESCRIPTION
<connector_name>	Name of the connector to start

cid2cbuloaderadm Syntax

```
cid2cbuloaderadm help | <host> <port> <command> [<parameters>]
```

PARAMETERS	DESCRIPTION
help	Displays help for the tool
<host>	Specifies the agent host
<port>	Specifies the agent administration port
<command>	Administration command

COMMAND	DESCRIPTION	PARAMETERS	RETURN VALUES
start	Resumes the execution of the connector. Use this command to restart the connector after a stop command.	none	-200 AGENT START
stop	Stops the connector. The connector no longer processes inbound and outbound messages	none	-120 AGENT STOP
shutdown	Terminates the connector	none	-100 AGENT SHUTDOWN
setparameter	Changes the settings of the connector remotely. Use the following syntax: setparameter <parameter>=<value>	AGENTNAPPING AGENTLOOPING ONE_EXECUTION NBRROW QFILL MAXNUMBERRETRY RETRYDELAY	-510 AGENT SET<parameter name>=<parameter value>
getparameter	Restarts the current value of the parameter	AGENTNAPPING AGENTLOOPING ONE_EXECUTION NBRROW QFILL MAXNUMBERRETRY RETRYDELAY Any parameter defined in the properties files	-610 AGENT GET<parameter name>=<parameter value>
getmode	Displays the current running mode	none	-210 AGENT NORMAL -220 AGENT REDUCE -230 AGENT PAUSE

COMMAND	DESCRIPTION	PARAMETERS	RETURN VALUES
getstatus	Displays the current status	none	130 AGENT NORMAL -140 AGENT REDUCE -150 AGENT PAUSE
force	Forces the connector to run in the specified mode	NORMAL REDUCE RECOVER	-410 AGENT SET MODE=NORMAL -410 AGENT SET MODE=REDUCE -410 AGENT SET MODE=RECOVER For MODE=RECOVER, an additional parameter must be provided: FILTER <FilterName>
info	Displays information about the connector	none	[AGENTNAME:xxx,NBTHREADS:xxx,FILL:xxx,NBROW:xxx,NAPPING:xxx,LOOPING:xxx,VERSION:xxx,BUILD:xxx,OS:xxx]
kill	Forces shutdown	none	none
list	Returns all the profiling information	none	none
purge	Purges the profiling information	none	none
stat	Returns the profiling information of the last profiling element	none	none
version	Returns the version of the connector	none	none

CustDim Loader Administration Tools

You use the following administration tools to administrate the CustDim Loader:

- `custdimloader`
- `custdimadm`

These tools are located in `<home_dir>/bin`.

Configuring the CustDim Loader Database Connection

You can define the CustDim loader database connection parameters. By default, you enter the CustDim connection parameters for the CID and CBU during installation. However, you may need to change the connection parameters after installation.

The CustDim loader uses the following configuration files:

- `instance_route.properties` configuration file to access the CID.
- cbu`_instance.properties` configuration file to access the CBU.

These files are located in `<home_dir>/config/custdim/nmycfg/dal/instances`.

To configure the database connection

1. Go to `<home_dir>/config/CustDim/nmycfg/cid2cbu`.
2. Do one of the following:
   - To change the connection to the CID, open `instance_route.properties`.
   - To change the connection to the CBU, open `cbu_instance.properties`
3. Enter the following:
   - `DRIVER`: enter the name of the driver to use
   - `URL`: enter the location of the database
   - `USER:` enter the login
   - `PASSWORD`: enter the password
4. Save your changes.

custdim Syntax

```
custdimloader CustDim
```

PARAMETERS	DESCRIPTION
CustDim	Name of the connector to start

custdimadm Syntax

```
custdimadm help | <host> <port> <command> [<parameters>]
```

PARAMETERS	DESCRIPTION
help	Displays help for the tool
<host>	Specifies the agent host
<port>	Specifies the agent administration port
<command>	Administration command

COMMAND	DESCRIPTION
start	Start the connector. If no data is found then the connector is automatically stopped.
shutdown	Shuts down the connector. This command waits for the completion of the current input file and all of its associated processing.
pause	Pause the execution of the connector. This command forces the connector to pause the extraction of new messages from all its inbound queues (invoice file for the splitter, message for the transformer and message for the CID loader and CBU loader)
resume	Resume the execution of the connector. This command forces the connector to resume the extraction of new messages from all its inbound queues.
kill	Kill the connector process. This command stops the connector immediately. It does not wait for the completion of all internal components and processing. This may lead to loss of data.

Report Manager Administration Tool

You use the reportmanageradm administration tool to purge the files created by the WFS FileManager.

This tool is located in <home_dir>/bin.

Configuring the Report Manager Administration Tool

You can define different properties of the Report Manager.

The Report Manager uses the options.properties configuration file to set its properties. This file is located in <home_dir>/config/reportmanager/nmycfg/wfs.

You can configure the following:

- The length of time a file can be locked
- Location of the directories

To set the lifespan file locks

1. Go to <home_dir>/config/reportmanager/nmycfg/wfs.
2. Open options.properties.
3. Enter the following:
    - For filemgr.lockfiles.admin.timetolive, enter the lifespan of the lock file in minutes. When the lock file is older than the specified length of time, the administration tool may purge it.
    - For filemgr.lockfiles.ignoreifolderthan, enter the age of the lock file in seconds. When the lock file is older than the specified amount of time, the Report Manager ignores it.
4. Save your changes.

To specify the root directories

1. Go to <home_dir>/config/reportmanager/nmycfg/wfs.
2. Open options.properties.
3. For permanent report files, enter the full path. Use the syntax:

    ```
    filemgr.rootdir.<repositoryCode>=full_path
    ```

    where <repositoryCode> is the code to use in report manager APIs

4. For temporary report files, enter the full path and specify the directory type. Use the syntax:

```
filemgr.rootdir.<repositoryCode>=full_path
filemgr.rootdir.<repositoryCode>.type=cache
```

5. Save your changes.

Example of options.properties

The number of minutes a lock file is allowed to exist	filemgr.lockfiles.admin.timetolive=1440
The number of seconds before the Report Manager ignores the lock file	filemgr.lockfiles.ignoreifolderthan=120

Declaration of cache1	filemgr.rootdir.cache1=C:/mydir/var/wfs/cache
	filemgr.rootdir.cache1.type=cache

Declaration of cache2	filemgr.rootdir.cache2=C:/mydir/cache
	filemgr.rootdir.cache2.type=cache

Declaration of permanent file directories	filemgr.rootdir.permanent1=C:/mydir/var/wfs/permanent
	filemgr.rootdir.permanent2=C:/mydir/var/wfs/permanent

reportmanageradm Syntax

```
reportmanageradm help | <command> <repositoryCode> [<keyCode>]
```

PARAMETERS	DESCRIPTION
help	Displays help for the tool
<command>	Administration command
<repositoryCode>	The file directory specified in options.properties
<keycode>	The Key code of the directory

COMMAND	DESCRIPTION	PARAMETERS
purgeCorrupted	Removes the reports not correctly written	repositoryCode
purgeAllReports	Removes all the reports for a given key	repositoryCode keyCode
purgeCache	Removes all the reports in a cache repository	repositoryCode keyCode

Index