Installing Telco Service & Analytics Manager Applications

V4.2
Document ID: TMGN-12-4.2-01
Date Published: 3.11.04

© 1997–2004 edocs® Inc. All rights reserved.

edocs, Inc., One Apple Hill Dr., 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.

This material is protected by U.S. and international copyright laws. edocs and eaPost are registered in the U.S. Patent and Trademark Office.

No part of this publication may be reproduced or transmitted in any form or by any means without the prior written permission of edocs, Inc.

eaSuite, eaDirect, eaPay, eaCare, eaAssist, eaMarket, and eaXchange are trademarks of edocs, Inc.

All other trademark, company, and product names used herein are trademarks of their respective companies.

Printed in the USA.

Preface

In This Section

Using this Manual

Welcome to Installing Edocs Telco Service & Analytics Manager Applications.

This manual covers installing and configuring applications in the edocs Telco Solutions.

Before You Get Started

You should be familiar with the following:

- Administrating UNIX or Windows servers
- Installing and configuring application servers
- Your application architecture
- Programming Java and Java Server pages
- Designing or working with databases
- eXtended Markup Language (XML)

Who Should Read this Manual

This manual is for anybody who needs to install the edocs Telco Service & Analytics Manager Applications or any of its components.

- Administrators

  You will find information several topics that involve the administration of Telco Service & Analytics Manager applications. There is information about preparing the environment before installation. After installing, you need to configure some of the components. There is an entire section dedicated to configuring these components. The section on user authentication should also be of interest.

  Pay attention to the section on monitoring as this section deals with the way you can track the behavior of your solution.

  The steps to deploy your solution are also covered along with a short list of post-installation steps you can run.You may also want to consult the Configuration File Reference for the comprehensive list of configuration files and their location.

- Developers

  This manual contains information you use to install a development environment to build your solution. You should pay close attention to the instructions concerning the installation and configuration of the Telco Service & Analytics Manager application and development tools.

  Although you may not need to configure all of the components which are installed, you must read the section concerning the location of configuration files, the CID and Monitoring your application as these sections deal with components you use during development of any type of solution. You also learn how to deploy the solution you develop. The Configuration File Reference is a comprehensive list of configuration files and their location.

- Project Architect

  You can use the information in this manual to determine the components you need to install on which machines. There are other topics covered in this manual which have an impact on system architectures. The configuring environment variables and configuring authentication are very important during the design and conception phase of a project. You may also want to look at the section covering deployment of the application.

- Project Manager

  You will find information about the installation of the various components that make up Telco Service & Analytics Manager applications. Once you have an idea of the components you need, you should look closely at the information in the chapter about configuring the components. You need to be familiar with configuring components as you need to take into account for both development and production environments. You need to read the section about monitoring Telco Service & Analytics Manager applications.

  You should also have a look at the deployment procedures.

How this Manual is Organized

This manual covers the following:

- **Preparing to Install**

  This chapter covers the preparation of your environment before you install and configure an TSM application.

  It contains information about:

  - The recommended steps for different application servers
  - Installing and configuring the database
  - Installing and configuring the application server
  - Installing and configuring the Web servers

- **Installing Applications**

  This chapter covers installing an application.

  It contains information about:

  - Installing the application
  - Uninstalling the application

- **Installing Development Tools**

  This chapter covers installing development tools.

  It contains information about:

  - Installing the tools
  - Uninstalling the tools

- **Configuring Telco Service & Analytics Manager Applications**

  This chapter covers configuring the Telco Service & Analytics Manager applications after installation.

  It contains information about:

  - Configuring the CID and CBU databases
  - Setting environment variables
  - Setting write permissions for shared directories
  - Managing authentication of users
  - Configuring the Approval Sequencer

- Configuring the CID2CBU loader
- Configuring Notification
- Configuring error logging

- **Deploying Personalization Manager Channels**

  This chapter covers the deployment of Personalization Manager Channels.

  It contains information about deploying the channels on the following supported platforms:

  - BEA WebLogic
  - SunONE
  - IBM WebSphere
  - Oracle 9i Application Server

- **Testing the Installation**

  This chapter covers the quick test to validate the installation.

  It contains information a list of the components items to test and how to test them.

- **Configuration File Reference**

  This appendix is an configuration file reference. It covers the location and use of configuration files as well as other configuration files used for customization.

  It contains information about the following:

  - File name
  - Location
  - Description

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`

Obtaining edocs Software and Documentation

You can download edocs software and documentation directly from Customer Central at https://support.edocs.com. After you log in, click on the Downloads button on the left. When the next page appears, you will see a table displaying all of the available downloads. To search for specific items, select the Version and/or Category and click the Search Downloads button. If you download software, an email from edocs Technical Support will automatically be sent to you (the registered owner) with your license key information.

If you received an edocs product installation CD, load it on your system and navigate from its root directory to the folder where the software installer resides for your operating system. You can run the installer from that location, or you can copy it to your file system and run it from there. The product documentation included with your CD is in the Documentation folder located in the root directory. The license key information for the products on the CD is included with the package materials shipped with the CD.

Finding the Information You Need

The product suite comes with comprehensive documentation set that covers all aspects of building solutions based on the edocs Telco Service & Analytics Manager. You should always read the release bulletin for late-breaking information.

Getting Started

If you are new to the edocs Telco Solutions, you should start by reading *Introducing Telco Service & Analytics Manager Applications*. This manual contains an overview of the various components along with the applications and their 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 which covers different aspects of working with the application. At the beginning of each manual, you will find an introductory chapter which covers concepts and tasks.

Designing Your Solution

While reading *Introducing Telco Service & Analytics Manager Applications*, you should think about how the different components can address your solution's needs.

You can refer to *Developing Telco Service Manager (TSM)* for information about extending the object model, application security, and other design issues. The *CID Reference Guide* also gives you the information about how the information in your solution is managed and stored.

You can refer to *Developing Telco Analytics Manager (TAM)* for information about customizing the database, synchronizing data with 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 TAM with TSM in *Developing Telco Analytics Manager (TAM)*.

You can also read the introduction of *Developing Connectors* for information about integrating your solution.

Installing Telco Service & Analytics Manager Applications

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

You might also refer to *Developing Telco Service & Analytics Manager Applications* and *Developing Connectors* as these manuals contain information on customizing applications and working with other software.

If you are upgrading, be sure to read *Migrating Telco Service & Analytics Manager Applications*.

Building Your Solution

If you are designing and programming your solution, you have several different sources of information. If you are programming the user interface of the solution, you should read *Developing User Interfaces*. You also refer to the *BLM Specification* and *JSPF specification* for detailed information about programming the user interface. For configuring the various components, you refer to *Installing Telco Service & Analytics Manager Applications* and sections in other documents which deal with the component to configure.

If you are designing and programming TAM, you have several different sources of information. If you are programming the user interface of the solution, you should read *Developing Reports*. You 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. For configuring the various components, you refer to *Installing Telco Service & Analytics Manager Applications* and sections in other documents which deal with the component to configure.

If you are working with the business logic of your solution, you should read *Developing Telco Service Manager (TSM)*. You can also refer to the *BLM Reference Guide* for more information about the design and structure of the BLM object model. For information about how this information is stored, you should refer to the *CID Reference Guide* along with the *CID Reference* documentation for your database. In order to develop your application, you most likely will need to install and run the Loopback Connector. This component mimics back-end applications for development purposes. For information about installing and running this component, refer to *Using the Loopback Connector*.

If you are working on the data warehouse side of TAM, you should read *Developing Telco Analytics Manager (TAM)*. 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. You should also read *Developing Telco Analytics Manager (TAM)* for information about synchronizing data between the TAM and *Telco Service Manager (TSM)*. In this manual, you will also find information about loading data in both the CBU and the CID.

For more information about integrating your application, you should read *Building Connectors* to learn how Telco Service & Analytics Manager applications work with different software.

Integrating Your Solution

If you are involved in configuring your solution to work with Operation Support Software (OSS), you should read *Building Connectors*. This manual helps you understand the integration architecture and shows you how to build connectors to connect to today's market-leading OSS software. You can also read *Using the Loopback Connector* for information about a connector built for development purposes. Other manuals you can refer to for information about configuring your application include *Introducing Telco Service & Analytics Manager Applications, Developing Telco Analytics Manager (TAM),* and *Developing Telco Service Manager (TSM)*.

Managing Telco Service & Analytics Manager Applications

If you are responsible for managing Telco Service & Analytics Manager applications, you should read the *Installing Telco Service & Analytics Manager Applications* for information about configuring various components and information about working with different application servers. *Administrating Telco Service & Analytics Manager Applications* covers what you need to know about managing your solution at runtime. For information about OSS systems, you should read *Building Connectors*.

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.

edocs provides global Technical Support services from the following Support Centers:

US Support Center

Natick, MA
Mon-Fri 8:30am – 8:00pm US EST
Telephone: 508-652-8400

Europe Support Center

London, United Kingdom
Mon-Fri 9:00am – 5:00 GMT
Telephone: +44 20 8956 2673

Asia Pac Rim Support Center

Melbourne, Australia
Mon-Fri 9:00am – 5:00pm AU
Telephone: +61 3 9909 7301

Customer Central

https://support.edocs.com

Email Support

mailto: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

Preparing to Install

In This Section

Before Installing

This section covers what you need to do before installing.

Applications use different software packages to run. We suggest installing and configuring these components before installing. One important thing to remember when installing is that you do not have to install all of the components on the same computer. If your system calls for parts of your application to be installed on several computers, you should go through the list of pre-installation tasks for information about preparing the various parts of your environment.

Before you install, you should check the recommended deployment procedures for your application server as the sequence of preparing your environment and installing depends on your application server.

You need to have full administrator permissions to install applications.

The main tasks of preparing your environment include:

- Reviewing the recommended deployment procedures
- Installing and configuring the RDBMS
- Installing and configuring the application server
- Installing and configuring the Web server

If you are upgrading, you should read the migration documentation BEFORE installing this version.

For a list of system requirements, refer to the Release Bulletin on the CD-ROM.

CD-ROM Contents

The CD-ROM contains the following elements:

ELEMENT	DESCRIPTION
release_bulletin.html	Covers issues affecting this version.
contents.html	Describes the contents of the CD-ROM and has links to the various elements and documentation.
install.html	Contains links to the installers for Windows and UNIX systems.
Documentation directory	Contains the entire documentation set.
Third-party directory	Adobe Acrobat Readers JRE Java software extensions
Install directory	Contains the install tools

Recommended Deployment Procedures

Applications work with a wide range of application servers for the JSPF and CSS Engine. Depending on your environment, you might already have some or all of the required products installed and running before you install your application. Preparing your environment depends on which application server you use.

We recommend installing the application it its own directory. This means that you install the product bundles in a directory then configure your application server to use the application at this location. By installing in a separate directory, you make your installation independent of the application server, web server and other software. This also eases managing your application and upgrading software.

For instance, when upgrading your application server from one major version to another, you may have to move application and content files from one location to another. If you install your application in its own directory, you do not have to worry about moving files and directories when upgrading other software and you are familiar with the location of the application directory and its contents.

Deploying for WebLogic

1. Install your RDBMS or RDBMS client if the database is on a remote machine
2. Create CID instance and required tablespaces
3. Install the required JDK
4. Install WebLogic
5. Configure WebLogic
6. Install the application
7. Configure the application
8. Deploy channels

Deploying for WebSphere

1. Install your RDBMS or RDBMS client if the database is on a remote machine
2. Create CID instance and required tablespaces
3. Install WebSphere Application Server
4. Create and deploy a Web application
5. Create a datasource
6. Install the application

7 Configure the application

8 Deploy channels

Configuring the Databases

Before you install the databases, you need to:

- Install and configure the database for your environment.

  Refer to the product documentation for details on installing and configuring your database.

  If you do not install your database on the same computer as the application server, remember to install your database's client software. Configure the connectivity to the database before installing the application server.

- Configure the database

  Depending on your database, you may have to configure specific settings before installing.

About the Databases

Before you install the Telco Service & Analytics Manager applications and development tools, you need to install and configure your RDBMS.

The applications use the following databases to store and manage information:

- Customer Interaction Datastore (CID)
- Communications and Billed Usage (CBU)

All applications require the CID. But the CBU is a slave database of the CID and not required by all applications. The CBU is used by Telco Analytics Manager (TAM) and the analytical features of customized applications to store billing and account information for analysis.

If your application does not require analytical features, you do not need to install and configure the CBU.

The Presentation Layer MyWeb channel installed by default requires both the CID and the CBU to run correctly. If you do not install and configure the CBU, this channel will not work correctly.

For more information about:

- The databases and their roles, refer to *Introducing Telco Service & Analytics Manager Applications.*

- The structure and organization of the CID, refer to the *CID Reference Guide* and the *CID Reference Documentation* corresponding to your database.

- The structure and organization of the CBU, refer to the *CBU Reference Guide* and the *CBU Reference Documentation* corresponding to your database

---

Configuring the CID

Configuring an Oracle Database for the CID

You need to do the following before installing:

1. Create the CID instance.
2. Create the following tablespaces:
   - `USER_DATA`
   - `USER_INDEX`
   - `REQUEST_DATA`
   - `REQUEST_INDEX`
   - `INVOICE_DATA`
   - `INVOICE_INDEX`
   - `RBS (rollback)`
   - `TEMP (`type: `TEMPORARY)`

   The recommended configuration is one physical disk for each table space. However, if your environment cannot have such a configuration, you should try to host a `DATA` tablespace on one physical disk and its corresponding `INDEX` tablespace on another.
3. Enable function based indexes:
   1. Open the `init.ora` file.
   2. Set the following parameters:

      ```
      QUERY_REWRITE_INTEGRITY=TRUSTED
      QUERY_REWRITE_ENABLED=TRUE
      ```

   For more information about tuning and enhancing the performance of your Oracle database, refer to *Oracle - Designing and Tuning For Performance*.

Configuring a DB2 Database for the CID

You need to do the following before installing:

1. Run `sqllib/java12/usejdbc2` to activate the JDBC 2 driver.
2. Create the CID instance.
3. Create the CID database in the CID instance.
4. Create the following system users:

- cidadmin
- ciduser

In DB2, system user names must be in lowercase, cannot have any underscores, and are limited to 8 characters. However, they are referred to as CID_ADMIN and CID_USER.

5 Add users to the CID database. Do the following:

1. At the system prompt, enter `db2`.
2. Enter `CONNECT TO <CID database name> USER <user of the DB2 instance that hosts CID database>`
3. Grant the necessary rights to `<CID_ADMIN>` and `<CID_USER>`

Example:

```
CONNECT TO CID USER db2inst1

GRANT CREATETAB, CONNECT, IMPLICIT_SCHEMA ON DATABASE
TO USER <CID_ADMIN>

GRANT CONNECT ON DATABASE TO USER <CID_USER>

TERMINATE
```

6 Create the following DMS (Data Managed Space) tablespaces:

- USER_DATA
- USER_INDEX
- REQUEST_DATA
- REQUEST_INDEX
- INVOICE_DATA
- INVOICE_INDEX

These tablespaces require a minimum page size of 16K and a bufferpool of 16K.

7 Grant `CID_ADMIN` full rights to each of the table spaces.

8 Create the following system temporary table spaces:

- TEMP

These tablespaces require a minimum page size of 16K and a bufferpool of 16K.

9 Configure DB2 client to use the correct schema:

1. Open the `db2cli.ini` file
2. Add the following:

```
[CID]

CURRENTSCHEMA = CID_ADMIN
```

You must configure all DB2 clients to use the correct schema.

Configuring a SQL Server Database for the CID

You need to do the following before installing:

1. Go to Microsoft's Web site.
2. Download and install the Microsoft SQL Server 2000 Driver for JDBC.
3. Create the CID database.
4. Create the following file groups and data files:
   - `USER_DATA`
   - `USER_INDEX`
   - `REQUEST_DATA`
   - `REQUEST_INDEX`
   - `INVOICE_DATA`
   - `INVOICE_INDEX`
5. Associate the file groups with their corresponding data files.

Configuring the CBU

Configuring an Oracle Database for the CBU

Configuring your database before installation involves:

- Creating Instances
- Creating Tablespaces

Creating Instances

Use the values in the table below when creating CBU instances:

ORACLE PARAMETERS	VALUE
optimizer_mode	CHOOSE
SGA	Max memory / 2
sort_area_size	Between 256K and 4 MB
block size	8K
query_rewrite_enabled	TRUE
star_transformation_enabled	TRUE

Creating Tablespaces

One of the most important parts of installing and configuring the CBU is allocating enough space for the different tablespaces. The sizes depend on your application and the type of information you load into the CBU. In order to give you an idea of the space you need, we give you a sample of information that a CBU may be required to manage. For the data in the sample, the tablespaces are listed with recommended sizing information.

The CBU contains the following tablespaces:

TABLESPACE NAME	DESCRIPTION
NMY_CBU_REF_DATA	Tablespace dedicated to the reference tables
NMY_CBU_REF_INDX	Tablespace dedicated to the indexes on reference tables
NMY_CBU_USER_DATA	Tablespace dedicated to the user tables (data which come in the CID)
NMY_CBU_USER_INDX	Tablespace dedicated to the indexes on user tables
NMY_CBU_USAGE_DATA	Tablespace dedicated to the usage table
NMY_CBU_USAGE_INDX	Tablespace dedicated to the indexes on usage table
NMY_CBU_INVC_DATA	Tablespace dedicated to the invoice tables
NMY_CBU_INVC_INDX	Tablespace dedicated to the indexes on invoice tables
NMY_CBU_TEMP	Tablespace used as temporary space by the users which query the database

These tablespaces require a minimum page size of 16K and a bufferpool of 16K.

This information is used to determine the sizing recommendations.

SIZING PARAMETER	CODE	SAMPLE
Number of contracts	C	20 000
Number of BUDR/Bill Period per contract	BBP	100
Number of Bill Periods	BP	3

In this sizing recommendation, we assume that there is one billing account per contract.

The following table presents the recommended sizing formula per tablespace and sample values.

TABLESPACE	SIZING FORMULA	AVERAGE UNIT SIZE	SAMPLE
NMY_CBU_REF_DATA	static size	10M	10M
NMY_CBU_REF_INDX	static size	10M	10M
NMY_CBU_USER_DATA	C	5K	100M
NMY_CBU_USAGE_DATA	C * BBP * BP	150 Bytes	~1G
NMY_CBU_INVC_DATA	C * BP	400 Bytes	24M

For index tablespaces, you should use the same sizing recommendations as the corresponding data tablespaces.

The size of the temp tablespace depends on the number of concurrent users and the queries.

Configuring a DB2 Database for the CBU

Configuring your database before installation involves:

- Activating the JDBC Driver
- Creating CBU users
- Creating Tablespaces
- Specifying the schema

Activating the JDBC Driver

1. Run `sqllib/java12/usejdbc2` to activate the JDBC 2 driver.

Creating CBU Users

1. Create the CBU instance.
2. Create the CBU database in the CBU instance.
3. Create the following system users:
   - `cbuadmin`
   - `cbuuser`

   In DB2, system user names must be in lowercase, cannot have any underscores, and are limited to 8 characters. However, they are referred to as CBU_ADMIN and CBU_USER.
4. Add users to the CBU database. Do the following:
   1. At the system prompt, enter `db2`.
   2. Enter `CONNECT TO <CBU database name> USER <user of the DB2 instance that hosts CBU database>`
   3. Grant the necessary rights to `<CBU_ADMIN>` and `<CBU_USER>`

   Example:

```
CONNECT TO CBU USER db2inst1
GRANT CREATETAB, CONNECT, IMPLICIT_SCHEMA ON DATABASE TO USER <CBU_ADMIN>
GRANT CONNECT ON DATABASE TO USER <CBU_USER>
TERMINATE
```

Creating Tablespaces

One of the most important parts of installing and configuring the CBU is allocating enough space for the different tablespaces. The sizes depend on your application and the type of information you load into the CBU. In order to give you an idea of the space you need, we give you a sample of information that a CBU may be required to manage. For the data in the sample, the tablespaces are listed with recommended sizing information.

The CBU contains the following tablespaces:

TABLESPACE NAME	DESCRIPTION
NMY_CBU_REF_DATA	Tablespace dedicated to the reference tables
NMY_CBU_REF_INDX	Tablespace dedicated to the indexes on reference tables
NMY_CBU_USER_DATA	Tablespace dedicated to the user tables (data which come in the CID)
NMY_CBU_USER_INDX	Tablespace dedicated to the indexes on user tables
NMY_CBU_USAGE_DATA	Tablespace dedicated to the usage table
NMY_CBU_USAGE_INDX	Tablespace dedicated to the indexes on usage table
NMY_CBU_INVC_DATA	Tablespace dedicated to the invoice tables
NMY_CBU_INVC_INDX	Tablespace dedicated to the indexes on invoice tables
NMY_CBU_TEMP	Tablespace used as temporary space by the users which query the database

These tablespaces require a minimum page size of 16K and a bufferpool of 16K.

This information is used to determine the sizing recommendations.

SIZING PARAMETER	CODE	SAMPLE
Number of contracts	C	20 000
Number of BUDR/Bill Period per contract	BBP	100
Number of Bill Periods	BP	3

In this sizing recommendation, we assume that there is one billing account per contract.

The following table presents the recommended sizing formula per tablespace and sample values.

TABLESPACE	SIZING FORMULA	AVERAGE UNIT SIZE	SAMPLE
NMY_CBU_REF_DATA	static size	10M	10M
NMY_CBU_REF_INDX	static size	10M	10M
NMY_CBU_USER_DATA	C	5K	100M
NMY_CBU_USAGE_DATA	C * BBP * BP	150 Bytes	~1G
NMY_CBU_INVC_DATA	C * BP	400 Bytes	24M

For index tablespaces, you should use the same sizing recommendations as the corresponding data tablespaces.

The size of the temp tablespace depends on the number of concurrent users and the queries.

Specifying the Schema

Configure DB2 client to use the correct schema:

1. Open the `db2cli.ini` file
2. Add the following:

```
[CBU]
CURRENTSCHEMA = CBU_ADMIN
```

You must configure all DB2 clients to use the correct schema.

Configuring a SQL Server Database for the CBU

Configuring your database before installation involves:

- Installing the JDBC Driver
- Creating file groups

Installing the JDBC Driver

1. Go to Microsoft's Web site.
2. Download and install the Microsoft SQL Server 2000 Driver for JDBC.

Creating File Groups

One of the most important parts of installing and configuring the CBU is allocating enough space for the different tablespaces. The sizes depend on your application and the type of information you load into the CBU. In order to give you an idea of the space you need, we give you a sample of information that a CBU may be required to manage. For the data in the sample, the tablespaces are listed with recommended sizing information.

The CBU contains the following tablespaces:

TABLESPACE NAME	DESCRIPTION
NMY_CBU_REF_DATA	Tablespace dedicated to the reference tables
NMY_CBU_REF_INDX	Tablespace dedicated to the indexes on reference tables
NMY_CBU_USER_DATA	Tablespace dedicated to the user tables (data which come in the CID)
NMY_CBU_USER_INDX	Tablespace dedicated to the indexes on user tables
NMY_CBU_USAGE_DATA	Tablespace dedicated to the usage table
NMY_CBU_USAGE_INDX	Tablespace dedicated to the indexes on usage table
NMY_CBU_INVC_DATA	Tablespace dedicated to the invoice tables
NMY_CBU_INVC_INDX	Tablespace dedicated to the indexes on invoice tables
NMY_CBU_TEMP	Tablespace used as temporary space by the users which query the database

These tablespaces require a minimum page size of 16K and a bufferpool of 16K.

This information is used to determine the sizing recommendations.

SIZING PARAMETER	CODE	SAMPLE
Number of contracts	C	20 000
Number of BUDR/Bill Period per contract	BBP	100
Number of Bill Periods	BP	3

In this sizing recommendation, we assume that there is one billing account per contract.

The following table presents the recommended sizing formula per tablespace and sample values.

TABLESPACE	SIZING FORMULA	AVERAGE UNIT SIZE	SAMPLE
NMY_CBU_REF_DATA	static size	10M	10M
NMY_CBU_REF_INDX	static size	10M	10M
NMY_CBU_USER_DATA	C	5K	100M
NMY_CBU_USAGE_DATA	C * BBP * BP	150 Bytes	~1G
NMY_CBU_INVC_DATA	C * BP	400 Bytes	24M

For index tablespaces, you should use the same sizing recommendations as the corresponding data tablespaces.

The size of the temp tablespace depends on the number of concurrent users and the queries.

Installing and Configuring the Application Server

Refer to the product documentation for details on installing and configuring your application server.

Installing and Configuring the Web Server

Refer to the product documentation for details on installing and configuring your Web server.

If your Web server runs on the same computer as the application server, you first install the Web server, then the application server.

CHAPTER 2

Installing Applications

In This Section

About Installing Telco Service & Analytics Manager

After preparing your environment, you can now install applications and their components. You use the easy-to-use installer to install the complete product suite or just the components you need.

The Telco Service & Analytics Manager Application Suite installer contains the following:

- Telco Service Manager (TSM)
- Telco Analytics Manager (TAM)
- System components

The Telco Service & Analytics Manager applications do not come with an uninstaller because most of the files of your applications are customized files. Therefore, an automatic uninstaller is not the most convenient and practical way of uninstalling.

Installing Telco Service & Analytics Manager Applications

To install applications

1. Run the installer application for your platform that you downloaded from edocs Customer Central or that is on the Telco Service & Analytics Manager CD-ROM. The Installer appears.
   - AIX: *tsm_aix.bin*
   - HPUX: *tsm_hpux.bin*
   - Solaris: *tsm_solaris.bin*
   - Windows: *tsm_windows.exe*
2. Click *Next*. The License Agreement window appears.
3. Read and accept the license agreement then click *Next*. The Location window appears.
4. Enter the home directory then click *Next*. The Installation Type window appears.
5. Choose one of the following then click *Next:*
   - **Full Installation** to install all of the components
   - **Presentation Layer** to install the following:

     Personalization Manager

     CSS Engine
   - **Synchronization Fwk** to install the following:

     SmartLink (ISF)

     Synchronization Layer Message Reference
   - **OSS/BSS Connectors** to install the following:

     Loopback Connector

     Connector Template
   - **TAM Fwk** to install the following:

     Communications and Billed Usage (CBU) database

     Notification Framework

     Customer Dimension Loader and other tools
   - **Database Tools** to install the database administration tools
   - **Custom installation** to select the components to install
6. Follow the on screen instructions to define the following directory and port information:

- Application Root Directory
- Web Root Directory
- Var Directory
- Synchronization Administration Port Number
- Loopback Administration Port Number
- Connector Template Administration Port Number
- Approval Sequencer Administration Port Number
- CID to CBU Loader Administration Port Number (TAM)
- Customer Dimensions Loader Administration Port Number (TAM)
- Report Job Processor Administration Port Number

7 Specify the location of your Java Home directory. Click *Next*.

8 Select the Database Type to use for the Customer Interaction Datastore and click *Next*:

- Oracle
- IBM DB2
- Microsoft SQL Server

9 Specify the location of the Database Client Home Directory and click *Next*.

10 Enter the Customer Interaction Datastore connection parameter information and click *Next*.

11 Enter the Communications Billing and Usage database connection parameter information and click *Next*.

12 The installer displays the Pre-Install Summary Screen. Confirm the settings you have chosen and click *Install*.

13 When finished, the installer displays a message.

To uninstall applications

1 Locate the home directory.

2 Make a backup copy of all customized files.

3 Delete the directory.

CHAPTER 3

Installing Development Tools

In This Section

About Installing Development Tools

The Telco Service & Analytics Manager Application Suite comes with an installer to install the Configuration and Deployment Toolkit.

The Configuration and Deployment Toolkit includes the following development tools:

- Integration Logic Studio (ILS)
- Presentation Logic Studio (PLS)

Configuration and Deployment Toolkit requires:

- at least 256 MB of RAM (512 MB recommended)

- 50 MB of disk space

Installing Development Tools

To install development tools

1. Run the installer *composer_windows.exe* that you downloaded from edocs Customer Central or that is on the Telco Service & Analytics Manager CD-ROM (the tools only run on Windows). The Configuration and Deployment Toolkit Installer appears.
2. Click *Next*. The License Agreement window appears.
3. Read and accept the license agreement then click *Next*. The Location window appears.
4. Enter the home directory then click *Next*. The Installation Type window appears.
5. The Installation Type window appears.
6. Choose one of the following:
   - **All Tools**
   - **Integration Logic Studio**
   - **Presentation Logic Studio**
7. Choose *Next.* The Choose Shortcut window appears.
8. Choose the location of the shortcuts then click *Next*. The Pre-Installation Summary window appears.
9. Review the summary then do one of the following:
   - If the summary is correct, click Install to begin installation
   - If the summary is not correct of if you want to change a setting, click *Previous.*

   When finished, the installer displays a message.

To uninstall development tools

1. Go to `<tool_home>/uninstalldata`.
2. Run `Uninstaller`. The Uninstaller appears.
3. Choose *Uninstall* to remove all of the components. When finished, the uninstaller displays a confirmation message.

If you install the tool on a computer running Windows, you should use the Windows Install/Remove Programs utility. You can find this utility in the Control Panel. Go to *Start>Settings>Control Panel*.

CHAPTER 4

Configuring Telco Service & Analytics Manager Applications

In This Section

About Configuring Telco Service & Analytics Manager

Once you have finished installing, you need to configure some components such as the CID and CBU databases. You also have to configure other software platforms such as the application server by creating a dedicated datasource for each database.

Configuring Telco Service & Analytics Manager applications involves:

- Configuring the CID database
- Configuring the CBU database
- Setting environment variables
- Setting write permissions for shared directories
- Managing authentication of users
- Configuring the Approval Sequencer
- Configuring the CID2CBU loader
- Configuring Notification
- Configuring the Job Processor

About Configuration Files and their Location

During installation, there are two identical sets of configuration files. This is done to respect the requirements of the J2EE Web Application aRchive (WAR) file specifications.

The configuration files are found in:

- `<home_dir>/classes/nmycfg`

  This directory contains the reference set of core configuration files. These files are the default set of configuration files.

- `<home_dir>/channels/WEB-INF/classes/nmycfg`

  This directory contains the Channel configuration files. These files are the configuration files that are deployed when using the J2EE WAR file.

For more information about WAR and configuration files, refer to *Deploying Personalization Manager Channels* in this manual.

Installing and Configuring the CID

After installation, you need to install and configure the CID.

Before you install the CID you need to configure your database.

You use the CID administration tool to:

- Create the CID users and database
- Populate the CID with reference information
- Create the CID and populate it with demo data for the Personalization Manager channel applications

---

If your application uses the search feature, you may need to create additional indexes before populating the CID. For more information about the search feature and creating indexes, refer to the *Optimizing Your Database For Searches* section in *Developing Telco Service Manager (TSM)*.

If you are upgrading your version, you should not create the CID using this tool. For information about upgrading the CID, refer to *Migrating Your Telco Service Manager (TSM)*.

Creating CID Users

The CID needs two users:

- **CID_ADMIN** Administrator to create the CID tables, synonyms, constraints, and to give the access permissions to the CID user.
- **CID_USER** Application user to connect the application to the CID database at run-time.

To create the CID user, you use the CID administration tool to create the CID users. You use the `create_cid_users` option.

If you are using DB2, you have already created these users. For more information about preparing DB2 for the CID, refer to *Preparing to Install - Installing and Configuring the Database*.

To create CID Users

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

```
cidAdminTool create_cid_users <CID> <DBA login> <DBA
password> <CID_ADMIN login> <CID_ADMIN password> <CID_USER
login> <CID_USER password>
```

where <CID>:

- Oracle: <instance alias>
- SQL Server: <database host> [:<port>];DatabaseName=<database 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.

  Example: "localhost;DatabaseName=CID", "server.enterprise:2000;DatabaseName=CID"

For DB2, the CID users are operating system users and cannot be created using this tool. For more information about users and DB2, refer to *Preparing to Install - Installing and Configuring the Database*.

Creating the CID Database

If you want to create the CID and populate it with the required system and application reference data, you use the install_cid option.

When installing the CID on DB2 and SQL Server, the installation creates triggers to enforce restraints on reference legacy IDs.

To create the CID Database

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

   ```
   cidAdminTool install_cid <CID> <CID_ADMIN login> <CID_ADMIN
   password> <CID_USER login> <CID_USER password>
   ```

   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.

When finished, the CID Administration tool displays a message.

To remove the CID

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

   ```
   cidAdminTool drop_demo_cid_structure <CID> <CID_ADMIN login>
   <CID_ADMIN password>
   ```

   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

   When finished, the CID Administration tool displays a message.

Creating the CID Database with Sample Data

To create the CID and populate it with sample data for the Personalization Manager channels, you use the cidAdmin tool.

When working with the demo CID Database, server components can only interact with the database using the `<CID_ADMIN>` account.

To create the CID with system and sample data

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

   ```
   cidAdminTool create_demo_cid_test <CID> <CID_ADMIN login>
   <CID_ADMIN password>
   ```

   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

When finished, the CID Administration tool displays a message.

Installing and Configuring the CBU

After installation, you need to install and configure the CBU.

Before you install the CBU you need to configure your database.

You use the CBU administration tool to:

- Create the CBU users and database
- Create the CBU and populate it with demo data

For DB2, the CBU users are system users and cannot be created using this tool. For more information about users and DB2, refer to *Configuring a DB2 Database for the CBU* in *Preparing to Install*.

Creating CBU Users

The CBU needs the following users to run properly:

- **CBU_ADMIN** The Administrator to create the CBU tables, constraints, and to give the access permissions to the CBU user.
- **CBU_USER** Application user to connect the CID2CBU loader and reporting tool to the CBU database at runtime.

To create the CBU users and their associated roles, you use the CBU administration tool.

To create CBU users

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

   ```
   cbuAdminTool create_cbu_users <CBU> <DBA login> <DBA password> <CBU_ADMIN password> <CBU_USER login> <CBU_USER password>
   ```

   where `<CBU>`:

   - Oracle: `<instance alias>`
   - SQL Server: `<database host> [:<port>];DatabaseName=<database name>`

     If no port is specified, the tool uses the default SQL server port

     Examples: "`localhost;DatabaseName=CID`", "`server.enterprise:2000;DatabaseName=CBU`"

For DB2, refer to *Configuring the CBU - Configuring a DB2 Database for the CBU.*

Creating the CBU Database

To create the CBU users and their associated roles, you use the CBU administration tool. The CBU administration comes with the following creation commands:

- `install_cbu` This command installs the CBU.
- `create_cbu_structure` This command creates the CBU structure for the sample toolkit.

If you want to install the CBU demo, refer to the sample toolkit's `readme.html` file for more information on installing and using the CBU demo. This file is located in `<home_dir>/samples/cbu/demo_kit`.

To create the CBU Database

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

   ```
   cbuAdminTool install_cbu <CBU instance alias> <CBU_ADMIN
   login> <CBU_ADMIN password> <CBU_USER login> <CBU_USER
   password>
   ```

   When finished, the CBU Administration tool displays a message.

To create the demo CBU Database

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

   ```
   cbuAdminTool create_cbu_structure <CBU instance alias>
   <CBU_ADMIN login> <CBU_ADMIN password>
   ```

   When finished, the CBU Administration tool displays a message.

Configuring Environment Variables

This section also covers how to set environment variables for stand alone components and tools that rely on third-party software. Depending on your system and environment, running your application may require setting some dedicated environment variables.

The environment variable values of these components are located in a dedicated `.env` environment setting file. The default values of these variables comes from the values you entered during installation.

Overview of Environment Variables

All tools and stand-alone components are started from scripts that are deployed in `<home_dir>/bin` directory.

This directory contains the following subdirectories:

- `env.cmd` for Windows systems
- `env.sh` for UNIX systems

These directories contains environment setting files for the corresponding operating system. These files also have file extensions corresponding to the operating system (`.cmd` for Windows and `.sh` for UNIX) . The number of files in each directory depends on the components installed by the Installer. The default environment values in these files are entered during installation.

The directories also has the `/custom.env` subdirectory. This subdirectory is for additional setting files you may need to add when customizing your deployment.

With this framework, each component benefits from having common environment settings and can be customized separately if required.

How the Environment Configuration Files Work

When running a tool or component using the launchers in the <home_dir>/bin directory, the launchers:

- Set some variable values
- Read and process the settings in the pre.env file.

  Settings in pre.env file are common to the whole distribution. The values in this settings file come from the values entered during installation.
- Read and process the settings in the post.env.

  This script is specifically designed for your custom settings. It tests the availability, read, and processes settings that are specific to the tool/server to launch.

  Because these settings are read and processed after the common settings, these settings should be:

  - New settings
  - Settings that overwrite the common settings

The launchers do the following:

1. Read the nmy.env file
2. Read component-specific files with names that are required by third-party software.

File Locations

The <home_dir>/bin directory contains the following:

DIRECTORY	CONTENTS	NOTES
/bin		Contains the tools and component launchers
/bin/env.cmd		Root directory of environment setting files for Windows
	nmy.env.cmd	Contains information generated by the installer
	pre.env.cmd	Contains common settings that are read and processed first by all tools/servers
	post.env.cmd	Contains the custom settings that override common settings and the specification of the custom .env.cmd files in the /custom.env directory
/bin/env.cmd/custom.env		Contains the custom environment settings
/bin/env.sh		Root directory of environment setting files for UNIX

	nmy.env.sh	Contains information generated by the installer
	pre.env.sh	Contains common settings that are read and processed first by all tools/servers
	post.env.sh	Contains the custom settings that override common settings and the specification of the custom .env.sh files in the /custom.env directory
/bin/env.sh/custom.env		Contains the custom environment settings

Changing Environment Settings

To change an environment setting, you change the values in `nmy.env`.

Example of changing database access settings

- During the installation workflow, Oracle was selected for both CID and CBU access.
- The installer prompts for the Oracle client home directory information.

If your Oracle client settings change after installation or if an error occurs while entering information, you need to modify the `NMY_CID_ORA_HOME` and `NMY_CBU_ORA_HOME` variable values in `nmy.env` file.

Example of Java runtime settings

If your Java Runtime home directory settings change after installation, you modify the `NMY_JAVA_HOME` variable value in the `nmy.env` file.

Example of .env Environment Setting Files

nmy.env Settings

```
@ECHO OFF
REM ============================================================================
REM Copyright (c) 1999-2003 Netonomy, Inc. All Rights Reserved.
REM ----------------------------------------------------------------------------

SET NMY_JAVA_HOME=c:\jdk1.3

SET NMY_CLASSES_DIR=c:/netonomy/classes
SET NMY_CONFIG_DIR=c:/netonomy/config
SET NMY_LIB_DIR=c:/netonomy/lib
SET NMY_JAR_DIR=c:/netonomy/lib

SET NMY_RDBMS_TYPE=oracle

SET NMY_ORA_HOME=d:\oracle\ora81
SET NMY_MSSQL_JDBC_HOME=
SET NMY_DB2_HOME=

REM ============================================================================
```

Including Custom Code in Environment Settings

If your application uses Java classes that you have developed or other classes that are required by your system architecture, you create an `.env` environment setting file to allow these classes to work with standalone components.

Each component launcher uses a dedicated .env environment setting file for declarations of custom or required variables. This environment setting file is located in the `<home_dir>/bin/env.cmd/custom` or `<home_dir>/bin/env.sh/custom` directory. The file name follows follows the syntax `launcher_name.env.cmd/sh`. For example, to create an `.env` environment setting file for the Synchronizer connector, you create a file called `agentstart.env.cmd` (or `.sh`) and put it in the `/custom` directory.

In general, activating custom code often deals with adding custom Java code to the `CLASSPATH`. You use this `.env` environment setting file in the `/custom` directory to update the `CLASSPATH` variable.

The launchers only take into account these settings when the file is deployed.

Example of Adding a Custom .jar Java Class Archive

The Synchronizer connector launcher is `agentstart.cmd` or `agentstart.sh`. The Synchronizer connector needs to access `custom.jar`.

1. Go to `<home_dir>/bin/env.cmd (.sh)/custom`.
2. Create a new file called `agentstart.env.cmd` (or `agentstart.env.sh`)
3. Add the following to the file:
   - For Windows

     `SET CLASSPATH=<path to file>/custom.jar;%CLASSPATH%`
   - On UNIX platforms, add the line below in the file:

     `CLASSPATH=<path to file>/custom.jar:$CLASSPATH`

The launcher now automatically takes this setting into account while starting.

Example of Overriding an Existing Java Class Archive

You can also use the `.env` environment setting files to override default classes and jar files.

For example, the Synchronizer connector uses the default `pop3.jar` file delivered with other Java classes and archives. If you have obtained a new Mail POP3 provider you want the Synchronizer connector to use, you can use the custom .env environment setting file to override the default POP3 provider.

To override the default jar file with a Java archive called `new_pop3.jar`

1. Go to `<home_dir>/bin/env.cmd (.sh)/custom`.
2. Create a new file named `agentsttart.env.cmd` (or `agentstart.env.sh`)
3. Add the following to the file:
   - For Windows

     `SET CLASSPATH=<path to file>/new_pop3.jar;%CLASSPATH%`
   - On UNIX platforms, add the line below in the file:

     `CLASSPATH=<path to file>/new_pop3.jar:$CLASSPATH`

   The launcher now automatically takes this setting into account while starting.

By using this method, your new jar file `new_pop3.jar` is handled first while resolving Java classes for loading. This is mandatory when you want to override classes.

You can also use this method to deploy maintenance releases or patches in production environments.

Setting Write Permissions for Shared Directories

Because application components may be installed on several different computers, you need to make sure that your application has all of the required access and permissions for shared directories.

You need to make sure that the application can access and has write permission for the following directories:

- `<home_dir>/var` and its subdirectories

These directories are for storing logs, connector message queues, and so on.

When developing, you also need to make sure you have access and write permission for `<home_dir>/share` and its subdirectories

Configuring Authentication

You can use either the CID or LDAP authentication method to grant users access to your application. By default, the CID is used for authentication.

You use the `functionlist.xml` file to configure the authentication method to use and other settings such as the one-way hash function to use to encrypt passwords in the CID. This file is located in `<home_dir>/classes/nmycfg/dal`.

Using LDAP involves:

- Specifying the authentication method to use in the `functionlist.xml` file:
  - JNDI
  - Netscape Directory
- Activating the method to use in the `LDAP.xml` file
- Specifying the servers to use and their port numbers in the `LDAP.xml` file

Specifying Password Encryption

If you use the CID for authentication, you can choose which one-way hash function to use to encrypt passwords.

You can use one of the following:

- MD5
- SHA
- None

To specify the encryption method

1. Go to `<home_dir>/classes/nmycfg/dal`.
2. Open `functionlist.xml`.
3. Go to the `<encryption_scheme>` element.
4. Do one of the following:
   - To use MD5, change the element to:
     `<encryption_scheme>MD5</encryption_scheme>`
   - To use SHA, change the element to:
     `<encryption_scheme>SHA</encryption_scheme>`
   - To not encrypt passwords, change the element to:
     `<encryption_scheme></encryption_scheme>`

5 Save your changes.

Example of MD5 Encryption

```
<XML_CONFIGURATOR __IS_HASH__="true">

...
<default_session_id>0</default_session_id>

<function_list>com.netonomy.dal.api.DalFunctionList</function_list>

<authenticator>com.netonomy.dal.api.authenticator.CidAuthenticator</authenticator>

<encryption_scheme>MD5</encryption_scheme>
<route_session_id>0</route_session_id>
<use_route_table>false</use_route_table>
<use_function_routing>true</use_function_routing>
</XML_CONFIGURATOR>
```

Specifying the Authentication Method

You specify the authentication method in the `functionlist.xml` file.

To specify the authentication method

1 Go to `<home_dir>/classes/nmycfg/dal`.

2 Open `functionlist.xml`

3 Find the `<authenticator>` element.

4 Do one of the following:

- To use the CID, change the `<authenticator>` element to:

  ```
  <authenticator>com.netonomy.dal.api.authenticator.CidAuthen
  ticator</authenticator>
  ```

- To use LDAP via JNDI, change the `<authenticator>` element to:

  ```
  <authenticator>com.netonomy.dal.api.authenticator.ldap.JNDI
  Authenticator</authenticator>
  ```

- To use LDAP via Netscape, change the `<authenticator>` element to:

  ```
  <authenticator>com.netonomy.dal.api.authenticator.ldap.Nets
  capeAuthenticator</authenticator>
  ```

5 Save your changes.

Example of `functionlist.xml`

```
<XML_CONFIGURATOR __IS_HASH__="true">

<!--

#  CID Authentication:  <authenticator>com.netonomy.dal.api.authenticator.CidAuthenticator</authenticator>

#  LDAP Authentication:
<authenticator>com.netonomy.dal.api.authenticator.ldap.JNDIAuthenticator</authenticator>

#  LDAP Authentication:
<authenticator>com.netonomy.dal.api.authenticator.ldap.NetscapeAuthenticator</authenticator>

-->

<default_session_id>0</default_session_id>

<function_list>com.netonomy.dal.api.DalFunctionList</function_list>

<authenticator>com.netonomy.dal.api.authenticator.CidAuthenticator</authenticator>

<encryption_scheme>SHA</encryption_scheme>

<route_session_id>0</route_session_id>

<use_route_table>false</use_route_table>

<use_function_routing>true</use_function_routing>

</XML_CONFIGURATOR>
```

Specifying the Authentication Server

You specify the LDAP authentication servers and their ports in the `LDAP.xml`file.

In this file, you specify:

- The LDAP authenticator to use (JNDI or Netscape)
- The name and port of the read and write server
- Pooling information
- Login verification
- LDAP parameters

To specify the authenticator to use

1. Go to `<home_dir>/classes/nmycfg/dal`.
2. Open `LDAP.xml`.
3. Verify that the section that corresponds to your authenticator is not commented out.
4. Verify that all other authenticator sections are commented out.
5. Save you changes.

Example of LDAP.xml configured for JNDI

```
<!-- Section to use for the standard JNDI LDAP driver (with
com.netonomy.dal.api.authenticator.ldap.JNDIAuthenticator) -->

<jndi_ldap_initial_ctx_factory>com.sun.jndi.ldap.LdapCtxFactory</jndi_ldap_initial_ctx_factory>

<jndi_ldap_provider_url>ldap://localhost:389</jndi_ldap_provider_url>

<jndi_ldap_security_authentication>simple</jndi_ldap_security_authentication>

<!-- Section to use for the Netscape LDAP JDK driver (with
com.netonomy.dal.api.authenticator.ldap.NetscapeAuthenticator) -->

<!--

<read_server_name>localhost:389</read_server_name>

<write_server_name>localhost:389</write_server_name>

<read_server_use_pool>true</read_server_use_pool>

<read_server_min_pool>10</read_server_min_pool>

<read_server_max_pool>50</read_server_max_pool>

<write_server_use_pool>true</write_server_use_pool>

<write_server_min_pool>10</write_server_min_pool>

<write_server_max_pool>10</write_server_max_pool>

    -->
```

To specify the authentication server

1. Go to `<home_dir>/classes/nmycfg/dal`.
2. Open `LDAP.xml`.
3. For the read server, enter the name and port number in the `<read_server_name>` element using the following syntax:

   ```
   <read_server_name>Server_Name:Port_Number</read_server_name>
   ```

4. For the write server, enter the name and port number in the `<write_server_name>` element using the following the syntax:

   ```
   <write_server_name>Server_Name:Port_Number</write_server_
   name>
   ```

5. For each server, you can:
   - Enable or disable the use of the connection pool (true/false) using the following element:

     ```
     <read_server_use_pool>true/false</read_server_use_pool>
     ```

   - Enter the minimum or maximum pool values using the following elements:

     ```
     <read_server_min_pool>10</read_server_min_pool>
     <read_server_max_pool>50</read_server_max_pool>
     ```

   - Enter the regular expression for the login in the `<schema_DN_regexp>` element using the following syntax:

     ```
     <schema_DN_regexp>your_login_regexp</schema_DN_regexp>
     ```

6. Enter the following LDAP parameters:

- User Distinguished Name in the <schema_DN_template> element using the following syntax:

  <schema_DN_template>uid=$login,o=customers,o=company.com</schema_DN_template>

- Object class hierarchy relative to the user in the <schema_objectclass> element using the following syntax:

  <schema_objectclass>top, Person</schema_objectclass>

- Object attributes (role and password) in the <schema_password> and <schema_roles> elements using the following syntax:

  <schema_password>userpassword</schema_password>

  <schema_roles>roles</schema_roles>

7 DN and password of the LDAP user with administrative rights in the <schema_DN_regexp> element using the following syntax:

<schema_DN_regexp>your_login_regexp</schema_DN_regexp>

8 Save your changes.

Example of `functionlist.xml`

```
<XML_CONFIGURATOR __IS_HASH__="true">

<!--

#  CID Authentication:  <authenticator>com.netonomy.dal.api.authenticator.CidAuthenticator</authenticator>

#  LDAP Authentication:
<authenticator>com.netonomy.dal.api.authenticator.ldap.JNDIAuthenticator</authenticator>

#  LDAP Authentication:
<authenticator>com.netonomy.dal.api.authenticator.ldap.NetscapeAuthenticator</authenticator>

-->

<default_session_id>0</default_session_id>

<function_list>com.netonomy.dal.api.DalFunctionList</function_list>

<authenticator>com.netonomy.dal.api.authenticator.CidAuthenticator</authenticator>

<encryption_scheme>SHA</encryption_scheme>

<route_session_id>0</route_session_id>

<use_route_table>false</use_route_table>

<use_function_routing>true</use_function_routing>

</XML_CONFIGURATOR>
```

Example of `LDAP.xml`

```
#  Multiple server example:

#         <read_server_name>Server1:portA Server2:portB Server3:portC</read_server_name>

#  Schema DN template examples:

#         <schema_DN_template>cn=$login,o=customers,o=netonomy.com</schema_DN_template>

#         <schema_DN_template>uid=$login, o=netonomy.com</schema_DN_template>

#         <schema_DN_template>uid=$login, objectclass=$param[0], o=netonomy.com</schema_DN_template>
```

Port numbers for 'read' and 'write' servers	<read_server_name>Server:389</read_server_name> <write_server_name>Server:389</write_server_name>

Connection pool values: For each server, you can enable or disable the use of the connection pool (true/false) Enter the minimum or maximum pool values for each server	<read_server_use_pool>false</read_server_use_pool> <read_server_min_pool>10</read_server_min_pool> <read_server_max_pool>50</read_server_max_pool> <write_server_use_pool>false</write_server_use_pool> <write_server_min_pool>10</write_server_min_pool> <write_server_max_pool>10</write_server_max_pool>
Login verification	<schema_DN_regexp>\$(login|[0-9]+)</schema_DN_regexp>
LDAP parameters User Distinguished Name Object class hierarchy relative to user Object attributes (role and password) The DN and password of the LDAP user with administrative rights	<schema_DN_template>uid=$login,o=customers,o=netonomy.com</schema_DN_template> <schema_objectclass>top, netonomyPerson</schema_objectclass> <schema_password>userpassword</schema_password> <schema_roles>roles</schema_roles> <user_netonomy_DN>uid=admin, ou=Administrators, ou=TopologyManagement, o=NetscapeRoot</user_netonomy_DN> <user_netonomy_Password>admin</user_netonomy_Password>

Configuring the Approval Sequencer

The Approval Sequencer calls a static BLM class in charge of evaluating Approval Processes.

You use the `agent.properties` configuration file to set the properties of the Approval Sequencer. This file is located in `<home_dir>/config/approvalsequencer`.

Configuring the Approval Sequencer involves:

- Configuring the standard settings
- Configuring the BLM connection:
  - The interval between calls to the BLM Approval class
  - The maximum number of requests to extract
  - The user name and password to connect to the BLM
- Configuring the database connection retries

Configuring the Standard Settings

You can configure:

- The location of the configuration file path
- The administration port

To specify the basic properties

1. Go to `<home_dir>/config/approvalsequencer`.
2. Open `agent.properties`.
3. Enter the following:
   - `CONFIG_DIR` enter the full path of the directory containing the `agent.properties` configuration file
   - `ADMIN_PORT` enter port number you use to administer the Approval Sequencer.
4. Save your changes.

Configuring the BLM Connection Settings

You can configure the following BLM Connection Settings:

- The interval between calls to the BLM Approval class
- The maximum number of approval processes to extract each time you call the BLM Approval class
- The user name and password to connect to the BLM

1 Go to `<home_dir>/config/approvalsequencer`.

2 Open `agent.properties`.

3 Set `NB_APPROVAL_PROCESSES` to the maximum number of approval processes to extract for processing.

4 Save your changes.

To specify the interval between calls to the BLM

1 Go to `<home_dir>/config/approvalsequencer`.

2 Open `agent.properties`.

3 For `NAPPING_TIME`, enter the time in milliseconds to wait before calling the BLM approval class.

4 Save your changes.

1 Go to `<home_dir>/config/approvalsequencer`.

2 Open `agent.properties`.

3 Enter the following:

- `BLM_USER` enter the user name the Approval Sequencer uses to connect to the BLM
- `BLM_PASSWORD` enter the associated password

4 Save your changes.

Configuring the Connection Retries

If the database connection fails, you can specify the number of times the Approval Sequencer tries to reconnect and the interval between each attempt to reconnect.

1 Go to `<home_dir>/config/approvalsequencer`.

2 Open `agent.properties`.

3 Enter the following:

- DB_RECONNECT_DELAY enter the time in milliseconds to wait between attempts to reconnect to the database
- DB_RECONNECT_RETRIES enter the number of times the Approval Sequencer tries to reconnect to the database.

4 Save your changes.

Example of the Approval Process Properties file

Configuration File path	CONFIG_DIR "file:///<home_dir>/config/approvalsequencer/"
BLM Connection settings	BLM_USER ="agent" BLM_PASSWORD="agent"
The time to wait before calling the BLM Approval class	NAPPING_TIME = 1000
Maximum number of approval processes to retrieve	NB_APPROVAL_PROCESSES = 1000
Database Connection Settings	DB_RECONNECT_DELAY = 60000 DB_RECONNECT_NBRETRIES = 30
Administration Port	ADMPORT = 3003

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
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.

1 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

1. 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
...
```

Configuring Notification

You can configure your application to manage notifications to synchronize data in the CBU.

When a notification event occurs, a notification is placed in the notification queue. As the notifications in the queue may involve the same type of change to an organization, you can specify if you want more than one notification for an action. For instance, for some changes you may only want to take into account only the last modification. If a user changes their contact information four times in a single day, you only need to take into account the last notification in order to synchronize the data in the CBU. However, if a user creates two new contracts, you can configure the synchronization to take both changes into account.

Configuring notifications involves:

- Activating notification
- Activating the `is_repeatable` flag
- Specifying the notification class that handles synchronization

To activate notification

1. Go to `<home_dir>/classes/nmycfg/blm`.
2. Open the `config.xml` configuration file.
3. Locate the `enabled` attribute of the `<BusinessLogic name="notification">` tag.
4. Change the values of this attribute to `true`.
5. Save your changes.

To set the is_repeatable flag

1. Use your database tool to connect to the CID.
2. In the `NOTIFICATION_OBJECT_TYPE` table, find the record containing the notification you want to change. Use the tables below to help with the default values:

NOTIFICATION_TYPE_ID	DESCRIPTION
1	Create notification
2	Modify notification
3	Remove notification
4	Update notification

OBJECT_TYPE_ID	DESCRIPTION
1	Level
3	Member
4	Contract
8	Billing account
10	User
17	Organization view

1. Enter 1 in the `IS_REPEATABLE` column.
2. Save your changes.

To specify the notification class

1. Go to `<home_dir>/classes/nmycfg/blm`.
2. Open the `config.xml` configuration file.
3. Under the `<BusinessLogic name="notification">` tag, create a `<property>` subelement specifying the class to use. Use the syntax:

   ```
   <property name="logic_class"
   value="com.<yourpackage>.netonomy.blm.external.<MyNotificatio
   nLogic>"/>
   ```

   where

   `<MyNotificationLogic>` is the name of your class.
4. Save your changes.

For more information about creating a notification logic class, refer to *Customizing Notification Logic* in *Developing Telco Analytics Manager (TAM)*.

```
<XML_CONFIGURATOR __IS_HASH__="false">
  <BusinessLogic name="AddService">
    <property name="UseRequestedRateplan" value="true"/>
  </BusinessLogic>
  <BusinessLogic name="notification">
    <property name="enabled" value="true"/>
    <property name="logic_class" value="com.<yourpackage>.netonomy.blm.external.MyNotificationLogic"/>
  </BusinessLogic>
</XML_CONFIGURATOR>
```

Configuring the Job Processor

You can define several Job Processors.

The Job Processor uses the `jobprocessor.properties` configuration file to set its properties. This file is located in `<home_dir>/config/jobservices/<Job Processor name>/nmycfg/jobservice`.

For each Job Processor, you configure a `jobprocessor.properties` configuration file.

Configuring the Job Processor involves:

- Specifying the administration port
- Specifying the user name and password to authenticate the Job Processor
- Specifying the connection retry mechanism
- Specifying the number of jobs to to fetch
- Specifying the number of threads to allocate to process jobs
- Specifying the Job class paths
- Specifying the service code filter
- Specifying the priority

To configure the Job Processor

1. Go to `<home_dir>/config/jobservices/<Job Processor name>/nmycfg/jobservice`.
2. Open `jobprocessor.properties`.
3. Enter the following:

PARAMETER	DESCRIPTION
ADM_PORT	Port number you use to administer the Job Processor (this parameter is set by the installer)
BLM_USER	User login to authenticate the Job Processor
BLM_PASSWORD	Associated password
NB_JOBS	Maximum number of jobs to fetch
IDLE_LOOPING_TIME	Milliseconds between checks for new jobs to process
RECONNECT_DELAY	Milliseconds between two connection retries to the required service (CID, CBU, WFS, and so on)
RECONNECT_NBRETRIES	Maximum number of times the Job Processor tries to reconnect to the required service
NB_THREADS	Number of threads allocated to process jobs
JOB_TASK_CLASS	The name of the class implementing the job service
JOB_TASK_CLASSPATH	The path to the class
SERVICE_CODE_FILTER	The service code used to filter the jobs to process. Use * for wildcards, do not set any value if not used

4 Save your changes.

To specify the priority

1 Go to `<home_dir>/config/jobservices/<Job Processor name>/nmycfg/jobservice`.

2 Open `jobprocessor.properties`.

3 Enter the following:

PARAMETER	DESCRIPTION
HIGH_PRIORITY	The numerical value of the processing ratio for high priority messages. Enter zero to ignore all such messages.
NORMAL_PRIORITY	The numerical value of the processing ratio for normal priority messages. Enter zero to ignore all such messages.
LOW_PRIORITY	The numerical value of the processing ratio for low priority messages. Enter zero to ignore all such messages.

4 Save your changes.

Example of jobprocessor.properties

Port Configuration	# Administration TCP port
	ADMPORT = 3006

Connection Parameters	# Login
	BLM_USER = agent
	BLM_PASSWORD = agent
	
	# Number of milliseconds elapsed when the idle job processor will check for new jobs to be processed
	IDLE_LOOPING_TIME = 1000
	
	# Number of milliseconds elapsed when the job processor tries to reconnect to any required resource (should be greater than or equal to database connection pool check period).
	RECONNECT_DELAY = 120000
	
	# Maximum number of reconnect to resource retries
	RECONNECT_NBRETRIES = 30

Number of threads	# Number of job worker threads
	NB_WORKER_THREADS = 1

Implementation classes	# Job implementation class (see the interface com.netonomy.jobservice.api.JobTask)
	JOB_TASK_CLASS = com.netonomy.reportjob.api.ReportJobTask
	JOB_TASK_CLASSPATH = c:/netonomy/lib/nmyreportjob.jar

Filters	# Service code filter (use * for wildcards, do not set any value if not used) SERVICE_CODE_FILTER=

Priority	# Job processing priority assignment
	# Default config (high=0, normal=1, low=0) This configuration processes only NORMAL priority jobs.
	# Typical config for priority management might be high=4, normal=2, low=1
	
	# HIGH priority jobs
	HIGH_PRIORITY = 0
	
	# NORMAL priority jobs
	NORMAL_PRIORITY = 1
	
	# LOW priority jobs
	LOW_PRIORITY = 0

CHAPTER 5

Deploying Personalization Manager Channels

In This Section

About Deploying Channels

After installing and configuring, you deploy the channels as a web application. When you deploy your channel, you are telling the application server where to find your application's JSPs and components. As each application server handles JSPs and other files differently, deploying your channel depends on the version and the editor of your application server.

This section covers deploying your application on different application servers. Deploying channels can be as easy as configuring your application server to look for the JSPs in a directory. Other application servers recommend that you deploy web applications as a J2EE Web Application aRchive (WAR) file.

For detailed information about configuring and deploying web applications, refer to your application server's documentation.

Working with BEA WebLogic

After installing your Telco Service & Analytics Manager application and WebLogic, you need modify your environment and system settings to use WebLogic.

This section deals with:

- BEA WebLogic 6.x
- BEA WebLogic 7.x

For WebLogic 6.x and 7.x

Deploying Channels on WebLogic 6.x and 7.x involves:

- Configuring your environment
- Creating and deploying a WAR file

Configuring Your Environment

Depending on your application server and environment, you may have to carry out certain tasks before you can deploy your channel.

For this application server, preparing your environment involves:

- Creating the connection pool
- Creating the data source for each database

To create a connection pool

Before creating your data source, you must create and configure a connection pool.

Refer to your application server documentation for more information about creating connection pool and activating the database connectivity.

For SQL Server, you must change the default `SelectMethod` in the JDBC connection string properties. The default `SelectMethod` is `direct`. The `SelectMethod` must be set to `cursor`.

To create a CID data source

1. Start the Weblogic Server.
2. Open the Weblogic Server Console.
3. Under *JDBC*, click *Data Sources*. The JDBC Data Sources page appears.

4 Click *Configure a new JDBC Data Source*. The Configure JDBC Data Sources page appears.

5 On the *Configuration* tab, enter the following:

FIELD	VALUE
Name	cidDatasource
JNDI Name	jdbc/cidDatasource

6 Click *Create*. The data source appears on the top of the page.

7 Click the home icon to return to the console home page.

Your WebLogic Server now has a declared data source corresponding to the CID.

To create a CBU data source

1 Start the Weblogic Server.

2 Open the Weblogic Server Console.

3 Under *JDBC*, click *Data Sources*. The JDBC Data Sources page appears.

4 Click *Configure a new JDBC Data Source*. The Configure JDBC Data Sources page appears.

5 On the *Configuration* tab, enter the following:

FIELD	VALUE
Name	`cbuDatasource`
JNDI Name	`jdbc/cbuDatasource`

6 Click *Create*. The data source appears on the top of the page.

7 Click the home icon to return to the console home page.

Your WebLogic Server now has a declared data source corresponding to the CBU.

Creating and Deploying a WAR File

In Java2EE, Sun Microsystems published the specification and tools to create Web Application Archive files (WAR) files. A WAR file is a JAR file containing java class files and other files required by Web applications (utility classes, HTML files, applets, and so on.) When using WAR files, you create a single file containing all of the required files for easy deployment on all Java2EE-compliant application servers.

A Web application can be run from directly from the WAR file or a directory that conforms to the WAR specification.

The `WAR` file for applications contains the following:

- Personalization Manager channel JSPs
- `WEB-INF` directory containing:
  - `web.xml` file describing the application
- `WEB-INF/lib` directory containing jar files
- `WEB-INF/classes` directory containing configuration files

After you generate your WAR file, you install this file using the application server's administration console.

Building a WAR file involves:

- Configuring the `web.xml` file
- Moving configuration files
- Generating the WAR file
- Deploying the WAR file

The java command for building WAR files (`jar.exe`) is located in your application server's copy of the Java Development Kit (JDK)

To configure the web.xml File

1. Go to `<app_dir>/WEB-INF` where `<app_dir>` is the location of your channel files.
2. Open `web.xml`.
3. Do the following:
   - In the `<display-name>` element, enter the name of your application.
   - In the `<description>` element, enter a description of the application.

   The name and description you enter here are for the application server only.

   ---
4. Save your changes.

Example of web.xml

```
<?xml version="1.0" encoding="UTF-8"?>


<!DOCTYPE web-app
    PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
    "http://java.sun.com/dtd/web-app_2_3.dtd">


<web-app>
  <display-name>Welcome to MyWeb</display-name>
  <description>Welcome to MyWeb</description>
</web-app>
```

To prepare your files

During installation, there are two identical sets of application configuration files. This is done to respect the requirements of the J2EE Web Application aRchive (WAR) file specifications and to help you easily deploy your application.

The sets of configuration files are:

- Core configuration files

  These configuration files are in `<home_dir>/classes/nmycfg`.
- Channel configuration files

  These configuration files are in `<app_dir>/WEB-INF/classes/nmycfg` where `<app_dir>` is the location of your channel files. By default, they are installed in `<home_dir>/Channels`.

However, when you develop your application, you may need to modify some of the core configuration files. When deploying your application, you must make sure that your modifications are also in the Channel configuration files. If you do not, your deployed application will not behave as expected.

1. Open the JSPF Configuration file (by default `MyWeb.xml`) and enable URL rewriting. Under the `<config>` element, change the content of the `<property name="GETURL_CALL_ENCODE_URL">` element to true. Example:

   `<property name="GETURL_CALL_ENCODE_URL">true</property>`
2. Copy the entire directory structure (`nmycfg/...`) of core configuration files located under `<home_dir>/classes` to `<app_dir>/WEB-INF/classes`.

Your directory structure is now a deployable directory structure that conforms to the WAR specification. The structure should look like this:

DIRECTORIES			CONTENTS
<Channels>/	common/		Contents of <home_dir>/channels/common
	MyWeb/		Contents of <home_dir>/channels/MyWeb
	WEB-INF/		The web.xml file
		lib/	Copy of the required jar files in <home_dir>/lib
		classes/	Copy of <home_dir>/classes

To generate the WAR file

1. Go to `<home_dir>/Channels`.
2. Generate the WAR file. Use the syntax:

   ```
   jar -cvf <home_dir>/myweb.war .
   ```

   Do not forget the final period at the end of this command.

   This Java command generates a WAR file called `myweb.war` in `<home_dir>`. You can generate this file in another directory if required. You use your application server's administration console to locate and deploy the generated WAR file.

To deploy your WAR file

1. Start the WebLogic Server.
2. Open the WebLogic Server Console.
3. Under Deployments, click Web Applications. The *Web Applications* page appears.
4. Click *Install a new Web Application*. The *Upload and Install an Application* page appears.
5. Click Browse to locate your `myweb.war` file. After locating the file, you return to the *Upload and Install an Application* page.
6. Click *Upload* to begin installation. The *Upload and Install an Application* page displays a message when the installation is finished.

Accessing a Deployed Channel

Before you can access your application, you need to restart your WebLogic server along with any other required components.

After you restart your application components, you can access your application using the following URL:

`http://host:port/<instance_name>/MyWeb/index.jsp`

where

`http://host:port/<instance_name>` corresponds to your WebLogic Server instance

Working with IBM WebSphere

After installing your Telco Service & Analytics Manager application and WebSphere, you need modify your environment and system settings to use WebSphere.

This section deals with:

- WebSphere 4.x
- WebSphere 5.x

For WebSphere 4.x

Deploying Channels on WebSphere 4.x involves:

- Configuring your environment
- Creating and deploying a WAR file
- Configuring the deployed channel

Configuring Your Environment

Depending on your application server and environment, you may have to carry out certain tasks before you can deploy your channel.

For this application server, preparing your environment involves:

- Creating the data source for each database
- Specifying Java memory settings

To create a CID data source

1. Start the WebSphere Administration Server.
2. Start the Administration Console.
3. Choose *Console>Wizards>Create Data Source*. The wizard opens.
4. Enter the following information:

   Name: `cidDatasource`

   Database name: your database name
5. Choose *Next.*
6. Choose *Create a new JDBC Driver.*
7. Enter the following information:
   - *Name:* `cidDatasource`
   - *Implementation class*

For the name of your implementation class, refer to your application server documentation.

8 Choose *Next*. A summary window appears.

9 Choose *Finish*.

10 On the console tree, go to the *Resources>JDBC Providers>cidDatasource/Data Sources* node.

11 On the *General* tab, enter the following:

- *User ID:* your CID user name
- *Password*: the associated password
- Add the required Custom Properties for your database.

  For more information about custom properties, select the *Help* button on the tab.

12 Choose *Apply*.

To create a CBU data source

1 Start the WebSphere Administration Server.

2 Start the Administration Console.

3 Choose *Console>Wizards>Create Data Source*. The wizard opens.

4 Enter the following information:

Name: `cbuDatasource`

Database name: your database name

5 Choose *Next.*

6 Choose *Create a new JDBC Driver.*

7 Enter the following information:

- *Name:* `cbuDatasource`
- *Implementation class*

For the name of your implementation class, refer to your application server documentation.

8 Choose *Finish*.

9 Go to the *Resources>JDBC Providers>cbuDatasource/Data Sources* node on the console tree.

10 On the *General* tab, enter the following:

- *User ID:* your CBU user name
- *Password*: the associated password

To specify Java memory settings

1. Start the WebSphere Administration Server.
2. Start the Administration Console.
3. Go to the *Nodes>Node Name>Application Servers/Default Server* node on the console tree.
4. Change the Command Line Parameter to increase the amount of allocated memory. By default, WebSphere does not allocate enough memory.

   For example, enter `-Xms196m -Xmx196m` to increase the allocated memory to 196 MB.

Creating and Deploying a WAR File

In Java2EE, Sun Microsystems published the specification and tools to create Web Application Archive files (WAR) files. A WAR file is a JAR file containing java class files and other files required by Web applications (utility classes, HTML files, applets, and so on.) When using WAR files, you create a single file containing all of the required files for easy deployment on all Java2EE-compliant application servers.

A Web application can be run from directly from the WAR file or a directory that conforms to the WAR specification.

The `WAR` file for applications contains the following:

- Personalization Manager channel JSPs
- `WEB-INF` directory containing:
  - `web.xml` file describing the application
- `WEB-INF/lib` directory containing jar files
- `WEB-INF/classes` directory containing configuration files

After you generate your WAR file, you install this file using the application server's administration console.

Building a WAR file involves:

- Configuring the `web.xml` file
- Moving configuration files
- Generating the WAR file
- Deploying the WAR file

The java command for building WAR files (`jar.exe`) is located in your application server's copy of the Java Development Kit (JDK)

To configure the web.xml File

1. Go to `<app_dir>/WEB-INF` where `<app_dir>` is the location of your channel files.
2. Open `web.xml`.
3. Do the following:
   - In the `<display-name>` element, enter the name of your application.
   - In the `<description>` element, enter a description of the application.

   The name and description you enter here are for the application server only.
4. Save your changes.

Example of web.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE web-app
    PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
    "http://java.sun.com/dtd/web-app_2_3.dtd">

<web-app>
  <display-name>Welcome to MyWeb</display-name>
  <description>Welcome to MyWeb</description>
</web-app>
```

To prepare your files

During installation, there are two identical sets of application configuration files. This is done to respect the requirements of the J2EE Web Application aRchive (WAR) file specifications and to help you easily deploy your application.

The sets of configuration files are:

- Core configuration files

  These configuration files are in `<home_dir>/classes/nmycfg`.

- Channel configuration files

  These configuration files are in `<app_dir>/WEB-INF/classes/nmycfg` where `<app_dir>` is the location of your channel files. By default, they are installed in `<home_dir>/Channels`.

However, when you develop your application, you may need to modify some of the core configuration files. When deploying your application, you must make sure that your modifications are also in the Channel configuration files. If you do not, your deployed application will not behave as expected.

1. Open the JSPF Configuration file (by default `MyWeb.xml`) and enable URL rewriting. Under the `<config>` element, change the content of the `<property name="GETURL_CALL_ENCODE_URL">` element to true. Example:

   ```
   <property name="GETURL_CALL_ENCODE_URL">true</property>
   ```

2. Copy the entire directory structure (`nmycfg/...`) of core configuration files located under `<home_dir>/classes` to `<app_dir>/WEB-INF/classes`.

   Your directory structure is now a deployable directory structure that conforms to the WAR specification. The structure should look like this:

DIRECTORIES			CONTENTS
<Channels>/	common/		Contents of <home_dir>/channels/common
	MyWeb/		Contents of <home_dir>/channels/MyWeb
	WEB-INF/		The web.xml file
		lib/	Copy of the required jar files in <home_dir>/lib
		classes/	Copy of <home_dir>/classes

To generate the WAR file

1. Go to `<home_dir>/Channels`.
2. Generate the WAR file. Use the syntax:

   ```
   jar -cvf <home_dir>/myweb.war .
   ```

   Do not forget the final period at the end of this command.

   This Java command generates a WAR file called `myweb.war` in `<home_dir>`. You can generate this file in another directory if required. You use your application server's administration console to locate and deploy the generated WAR file.

To deploy the WAR file

1. Start the WebSphere Administration Server.
2. Start the Administration Console.
3. Choose *Console>Wizards>Install Enterprise Application*. The wizard opens.
4. Choose *Install stand-alone module.*
5. Enter the following information:

   Path: `<home_dir>/myweb.war`

   Name: `MyWeb`

   Root: `/<instance_name>`
6. Choose *Next.* Accept the default values and choose *Next* until you exit the wizard.

Configuring Deployed Channels

During installation, the installer creates directories and copies files in standard default locations.

There is also a set of configuration files that have default values that you may have to change for your environment.

After deploying your channels, you need to configure some configuration files. You modify the files found in your `<DEPLOY_DIR>`.

By default, WebSphere deploys the war files in `<WEBSPHERE_HOME>/AppServer/installedApps/MyWeb.ear/myweb.war`.

Configuring deployed channels involves:

- Changing path to log files in `log4j.properties`

To change logger paths

1. Go to `<DEPLOY_DIR>/WEB-INF/classes/nmycfg/util`.
2. Open `log4j.properties`.
3. Do the following:
   - Set `log4j.appender.ROL.File` to `/tmp/nmy_application.log`
   - Set `log4j.appender.DAY.File` to `/tmp/nmy_daily_application.log`
4. Save your changes.

Accessing a Deployed Channel

Before you can access your application, you need to restart your WebSphere server along with any other required components.

After you restart your application components, you can access your application using the following URL:

`http://host:port/<instance_name>/MyWeb/index.jsp`

where

`http://host:port/<instance_name>` corresponds to your WebSphere Server instance

For WebSphere 5.x

Deploying Channels on WebSphere 5.x involves:

- Configuring your environment
- Creating and deploying a WAR file
- Enabling URL Rewriting

Configuring Your Environment

Depending on your application server and environment, you may have to carry out certain tasks before you can deploy your channel.

For this application server, preparing your environment involves:

- Creating a JDBC Driver
- Creating the data source for each database

To create a JDBC Driver

1. Start the WebSphere Administration Server.
2. Start the Administration Console.
3. On the Navigation tree, expand *Resources*.
4. Choose *JDBC Providers*. The *JDBC Providers* page appears.
5. Choose *Scope* then select *Apply*.
6. Choose *New* to create a new JDBC provider. The *New* page appears.
7. For *JDBC Providers*, select the select the type of provider to create.

   For example, select `Oracle JDBC oci8 Driver` to create a data source which connects to an Oracle database using connection pooling.
8. Choose *Apply*. The *Configuration* page appears.
9. Enter the following:
   - Name: name of the Oracle JDBC oci8 Driver: For example, `cidPool`
   - Description: description: ex: `Oracle JDBC oci8 Driver - CID`
   - Classpath: The location of `classes12.zip`. By default, it is `<oracle_home>/jdbc/lib`
   - Leave the default Implementation Classname: `oracle.jdbc.pool.OracleConnectionPoolDataSource`
10. Choose Apply then OK. The following message appears:

    Changes have been made to your local configuration. Choose *Save* to apply changes to the master configuration.
    The server may need to be restarted for these changes to take effect
11. Choose *Save* to save your changes.

To create a CID data source

1. Start the WebSphere Administration Server.
2. Start the Administration Console.
3. On the Navigation tree, expand *Resources*.
4. Choose *JDBC Providers*. The *JDBC Providers* page appears.
5. Select the JDBC provider to be used by your version 4 data source. The configuration panel for the JDBC Provider appears.
6. Scroll down and select *Data Sources* (Version 4) in the *Additional Properties* table.
7. Choose *New* to create a new data source or select an existing one to modify the data source properties.
8. Enter the following:

- Name: the Name of the datasource: For example, `cidDatasource`
- JNDI Name: the JNDI Name for example `jdbc/cidDatasource`
- Description: enter the description.
- Database Name: Enter the Database Name For example: `MyCID`
- Default User ID: Enter the Default User ID. For example: `CID_USER`
- Default Password: Enter the Default Password. For example: `CID_USER`

9 Choose *Apply*.

10 In the *Additional Properties* table, select *Custom Properties.*

11 Configure the URL property

- Choose *URL*
- Enter the Value For example: `jdbc:oracle:oci8:@MyCID`

12 Choose *Apply.*

13 Scroll down and choose *OK.* The following message appears:

```
Changes have been made to your local configuration. Choose to
apply changes to the master configuration.
The server may need to be restarted for these changes to take
effect
```

14 Choose *Save* to save your changes.

To create a CBU data source

1 Start the WebSphere Administration Server.

2 Start the Administration Console.

3 On the Navigation tree, expand *Resources*.

4 Choose *JDBC Providers*. The *JDBC Providers* page appears.

5 Select the JDBC provider to be used by your version 4 data source. The configuration panel for the JDBC Provider appears.

6 Scroll down and select *Data Sources* (Version 4) in the *Additional Properties* table.

7 Choose *New* to create a new data source or select an existing one to modify the data source properties.

8 Enter the following:

- Name: the Name of the datasource: For example, `cbuDatasource`
- JNDI Name: the JNDI Name for example `jdbc/cbuDatasource`
- Description: enter the description.
- Database Name: Enter the Database Name For example: `MyCBU`
- Default User ID: Enter the Default User ID. For example: `CBU_USER`

- Default Password: Enter the Default Password. For example: `CBU_USER`

9 Choose *Apply*.

10 In the *Additional Properties* table, select *Custom Properties.*

11 Configure the URL property

- Choose *URL*
- Enter the Value For example: `jdbc:oracle:oci8:@MyCBU`

12 Choose *Apply.*

13 Scroll down and choose *OK.* The following message appears:

```
Changes have been made to your local configuration. Choose to
apply changes to the master configuration.
The server may need to be restarted for these changes to take
effect
```

14 Choose *Save* to save your changes.

Creating and Deploying a WAR File

In Java2EE, Sun Microsystems published the specification and tools to create Web Application Archive files (WAR) files. A WAR file is a JAR file containing java class files and other files required by Web applications (utility classes, HTML files, applets, and so on.) When using WAR files, you create a single file containing all of the required files for easy deployment on all Java2EE-compliant application servers.

A Web application can be run from directly from the WAR file or a directory that conforms to the WAR specification.

The `WAR` file for applications contains the following:

- Personalization Manager channel JSPs
- `WEB-INF` directory containing:
  - `web.xml` file describing the application

- `WEB-INF/lib` directory containing jar files
- `WEB-INF/classes` directory containing configuration files

After you generate your WAR file, you install this file using the application server's administration console.

Building a WAR file involves:

- Configuring the `web.xml` file
- Moving configuration files
- Generating the WAR file
- Deploying the WAR file

The java command for building WAR files (`jar.exe`) is located in your application server's copy of the Java Development Kit (JDK)

To configure the web.xml File

1. Go to `<app_dir>/WEB-INF` where `<app_dir>` is the location of your channel files.
2. Open `web.xml`.
3. Do the following:
   - In the `<display-name>` element, enter the name of your application.
   - In the `<description>` element, enter a description of the application.

   The name and description you enter here are for the application server only.

   ---
4. Save your changes.

Example of web.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE web-app
    PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
    "http://java.sun.com/dtd/web-app_2_3.dtd">

<web-app>
  <display-name>Welcome to MyWeb</display-name>
  <description>Welcome to MyWeb</description>
</web-app>
```

To prepare your files

During installation, there are two identical sets of application configuration files. This is done to respect the requirements of the J2EE Web Application aRchive (WAR) file specifications and to help you easily deploy your application.

The sets of configuration files are:

- Core configuration files

  These configuration files are in `<home_dir>/classes/nmycfg`.

- Channel configuration files

  These configuration files are in `<app_dir>/WEB-INF/classes/nmycfg` where `<app_dir>` is the location of your channel files. By default, they are installed in `<home_dir>/Channels`.

However, when you develop your application, you may need to modify some of the core configuration files. When deploying your application, you must make sure that your modifications are also in the Channel configuration files. If you do not, your deployed application will not behave as expected.

1. Open the JSPF Configuration file (by default `MyWeb.xml`) and enable URL rewriting. Under the `<config>` element, change the content of the `<property name="GETURL_CALL_ENCODE_URL">` element to true. Example:

   ```
   <property name="GETURL_CALL_ENCODE_URL">true</property>
   ```

2. Copy the entire directory structure (`nmycfg/...`) of core configuration files located under `<home_dir>/classes` to `<app_dir>/WEB-INF/classes`.

   Your directory structure is now a deployable directory structure that conforms to the WAR specification. The structure should look like this:

DIRECTORIES			CONTENTS
<Channels>/	common/		Contents of <home_dir>/channels/common
	MyWeb/		Contents of <home_dir>/channels/MyWeb
	WEB-INF/		The web.xml file
		lib/	Copy of the required jar files in <home_dir>/lib
		classes/	Copy of <home_dir>/classes

To generate the WAR file

1. Go to `<home_dir>/Channels`.
2. Generate the WAR file. Use the syntax:

   ```
   jar -cvf <home_dir>/myweb.war .
   ```

   Do not forget the final period at the end of this command.

   This Java command generates a WAR file called `myweb.war` in `<home_dir>`. You can generate this file in another directory if required. You use your application server's administration console to locate and deploy the generated WAR file.

To deploy your WAR file

1. Start the WebSphere Administration Server.
2. Start the Administration Console.
3. On the Navigation tree, expand *Resources*.
4. Choose *Install New Application*. The Preparing for the application installation page appears.
5. Enter the following:
   - Server path: `<home_dir>/myweb.war`
   - Context Root: /<instance_name>
6. Choose *Next*. The AppDeployment Options page appears.
7. Enter the following:
   - Directory to Install Application: Enter the directory. For example, `/opt/WebSphere5/AppServer/installedApps/tonga/MyNeto32`
   - Application Name: Enter the name of the application
8. Choose *Next*.
9. Choose *Next*.
10. In *Map modules to application servers*, specify the application server where you want to install modules contained in your application.
11. Choose *Apply*.
12. Choose *Next*.
13. Choose Finish. The administration page displkays a message.
14. Choose *Save* your workspace changes to the master configuration
15. Update Web Server Plug-in. Do the following:
    - Expand *Environment* on the navigation tree
    - Choose *Update Web Server Plug-in*

- Choose O*K* button to update the plugin configuration file.

16 Choose *Save* to save changes to the master configuration.

Enabling URL Rewriting

You need to enable URL Rewriting for this version of WebSphere.

To enable URL rewriting

1 Start the WebSphere Administration Server.

2 Start the Administration Console.

3 On the Navigation tree, expand *Servers*.

4 Choose *Application Servers*. The *Application Servers* page appears.

5 Choose the link corresponding to your server name

6 In the *Additional Properties* table, select *Web Container*. The *Web Container* page appears.

7 In the Additional Properties table, select *Session Management*. The *Session Management* page appears.

8 Clear *Enable Cookies* and select *Enable URL Rewriting*.

9 Choose *Apply*.

10 Choose Save to save your workspace changes to the master configuration.

Accessing a Deployed Channel

Before you can access your application, you need to restart your WebSphere server along with any other required components.

After you restart your application components, you can access your application using the following URL:

`http://host:port/<instance_name>/MyWeb/index.jsp`

where

`http://host:port/<instance_name>` corresponds to your WebSphere Server instance

CHAPTER 6

Testing the Installation

In This Section

Installation Quick Test

Here is a list of quick tests you can run in order to verify that your system is set up and configured correctly:

COMPONENT TO TEST	DESCRIPTION
Java	to make sure you have the required versions of java installed
`http:/MyServer:port/MyWeb/Index.jsp`	to make sure you can access the Web server and application server
`http:/MyServer:port/MyWeb/login.jsp`	to test the application's database connection
`cidAdminTool`	to test your database connection
agentstart	to make sure the Synchronizer starts and runs correctly

APPENDIX A

Configuration File Reference

In This Section

About the Configuration Files

You use several different configuration and customization files to set up and modify your application's behavior.

The files are divided into the following categories:

CATEGORY	DESCRIPTION
Configuration	These files contain the default configuration. You can modify the information in these files to modify the default configuration to meet your deployment needs. Some of the configuration files also contain information that you enter during installation.
Customization	These files contain settings that you change to customize your application. You can modify the information in these files to modify the default configuration to meet your specific application needs. Some of these files may be empty or contain sample information.

Configuration Files

FILE NAME	LOCATION	DESCRIPTION
cbu_adapter.properties	config/cid2cbuloader/nmycfg/cid2cbu	Specifies the BLM mapper properties
cid_tools.properties	lib/admin/cid	Specifies the properties of the cidAdminTool
cid2cbuloader.properties	config/cid2cbuloader	Specifies the properties of CID2CBU loader
config.xml	classes/nmycfg/blm channels/WEB-INF/classes/nmycfg/blm	Specifies the display of rate plans
core_english.properties	classes/nmycfg/errors channels/WEB-INF/classes/nmycfg/errors	Specifies the messages of BLM error in English
core_french.properties	classes/nmycfg/errors channels/WEB-INF/classes/nmycfg/errors	Specifies the messages of BLM error in French
functionlist.xml	classes/nmycfg/dal channels/WEB-INF/classes/nmycfg/dal	Specifies the security settings and encoding of the Database connection
instance_route.properties	channels/WEB-INF/classes/nmycfg/dal/instances config/approvalsequencer/nmycfg/dal/instances config/cid2cbuloader/nmycfg/dal/instances config/notifyorg/nmycfg/dal/instances config/synchronizers/synchronizer/nmycfg/dal/instances	Specifies the properties of the database connection
jfnApplication.properties	channels/WEB-INF/classes/nmycfg/jfn	Specifies the media used by the JSPF
job_tools.properties	lib/admin/jobs	Specifies the properties of the jobAdminTool
jsp_parameters.xml	classes/nmycfg/util/formatter channels/WEB-INF/classes/nmycfg/util/formatter	Specifies the time, date and decimal format for the JSPs
LDAP.xml	classes/nmycfg/dal channels/WEB-INF/classes/nmycfg/dal	Specifies the LDAP configuration
logger.properties log4j.properties	classes/nmycfg/util channels/WEB-INF/classes/nmycfg/util config/connectors/connectortemplate/nmycfg/util config/connectors/loopback/nmycfg/uti config/synchronizers/synchronizer/nmycfg/util config/approvalsequencer/nmycfg/util config/cid2cbuloader/nmycfg/util config/notifyorg/nmycfg/util	Specifies the system logger properties

notifyorg.properties	config/notify_org/nmycfg/notifyorg	Specifies the properties of the NOTIFYORG tool
options.properties	config/reportmanager/nmycfg/wfs channels/WEB-INF/classes/nmycfg/wfs	Specifies the properties of the savedreportAdminTool
persistence_tools.properties	/lib/admin/persistence	Specifies the properties of the messagecacheAdminTool
policy.properties	classes/nmycfg/blm/util channels/WEB-INF/classes/nmycfg/blm/util	Specifies the properties of BLM objects for reference data reloading
translator.properties	classes/nmycfg/util channels/WEB-INF/classes/nmycfg/util	Specifies the location of the language files

Customization Files

FILE NAME	LOCATION	DESCRIPTION
containers_customization.xml	classes/nmycfg/dal channels/WEB-INF/classes/nmycfg/dal	Specifies the DAL container routing properties
core_containers.xml	classes/nmycfg/dal channels/WEB-INF/classes/nmycfg/dal	Specifies the configuration of access to the DAL
core_queries.xml	classes/nmycfg/dal/instances channels/WEB-INF/classes/nmycfg/dal/instances	Specifies the DAL queries
external_custom.xml	classes/nmycfg/blm channels/WEB-INF/classes/nmycfg/blm	Specifies the BLM external classes and their custom extensions
instances.properties	classes/nmycfg/dal/instances channels/WEB-INF/classes/nmycfg/dal/instances	Specifies the location of the DAL instance connection configuration file
MyWeb.xml	channels/common channels/MyWeb	Specifies the presentation layer application properties, string localization, and workflows.
objectID_aliases.xml	classes/nmycfg/dal channels/WEB-INF/classes/nmycfg/dal	Specifies the DAL aliases for objects
security.xml	classes/nmycfg/blm channels/WEB-INF/classes/nmycfg/blm	Specifies access rights to BLM functions

Index