

PDB System Administration Guide

Parameter Database

SYSTEM ADM. GUIDE

Copyright

© Ericsson AB 2012 – 2016. All rights reserved. No part of this document may be reproduced in any form without the written permission of the copyright owner.

Disclaimer

The contents of this document are subject to revision without notice due to continued progress in methodology, design, and manufacturing. Ericsson shall have no liability for any error or damage of any kind resulting from the use of this document.

Trademark List

Ericsson

Ericsson is the trademark or registered trademark of Telefonaktiebolaget LM Ericsson. All other product or service names mentioned in this manual are trademarks of their respective companies.



Contents

1	About This Guide	1
1.1	Intended Audience	1
1.2	How This Guide is Organized	1
1.3	Conventions Used in This Guide	2
1.4	Prerequisites	4
1.5	Comments About the Documentation	4
2	Overview	5
2.1	PDB GUI	5
2.2	PDB Server	6
3	User Accounts	6
3.1	Access Control	7
3.2	Configuring GlassFish and the PDB Server to Authenticate Against an LDAP Directory	10
4	Configuring PDB	19
4.1	PDB Settings Parameters	20
5	Auditing PDB	22
6	Configuring System Notifications	26
7	PDB Logging	28
7.1	Checking the Log File Using the GlassFish Administration GUI	29
7.2	Defining Component Log Levels for PDB	30
8	Starting and Stopping PDB	32
8.1	Stopping the GlassFish Application	32
8.2	Starting the GlassFish Application	32
8.3	Restarting the GlassFish Application	32
8.4	Stopping the PDB Application on GlassFish	33
8.5	Starting the PDB Application on GlassFish	33
8.6	Checking the GlassFish Application Server	33
8.7	Starting GlassFish Admin Console	34
9	Backing Up PDB	34



9.1	Automated Backup	35
9.2	Manual Backup	36
10	Restoring PDB	37
10.1	PDB Data Recovery	37
10.2	PDB Software Recovery	38
11	Configuring the PDB CLI	39
	Reference List	40



1 About This Guide

This document provides information on managing the Parameter Database (PDB) and describes the Operation and Maintenance (OAM) tasks that can be performed on the PDB server by system administrators.

1.1 Intended Audience

This document is intended for OAM personnel and system administrators involved in the management of PDB.

Personnel working on Ericsson products or systems must have the training and competence required to perform their work correctly.

1.1.1 Prerequisite Knowledge

In addition to a familiarity with PDB, users of this document should have knowledge and experience of the following:

- GlassFish application server
- Linux
- MySQL databases
- eXtensible Markup Language (XML)

1.2 How This Guide is Organized

This document is organized into the following major sections:

Table 1 Document Organization

Section	Description
About This Guide	Introduces the guide, describing prerequisites, document structure, and the conventions used.
Overview	Provides an overview of PDB administration.
User Accounts	Describes PDB user accounts and access control.



Section	Description
Configuring PDB	Describes PDB settings and the configuration parameters.
Auditing PDB	Describes the PDB audit log and ways to interact with it.
Configuring System Notifications	Provides information about system notifications.
PDB Logging	Describes the PDB log file and includes procedures to interact with it. The PDB log is essential for system behavior and fault analysis.
Starting and Stopping PDB	Provides information on how to start and stop PDB software, including the GlassFish Application Server (AS) and the Admin Console.
Backing Up PDB	Describes automated and manual procedures to backup the PDB database.
Restoring PDB	Describes procedures to restore PDB data and recover the system.
Configuring the PDB CLI	Describes how to configure the PDB CLI as a client of the PDB server.

1.3 Conventions Used in This Guide

Table 2 provides a list of typographic conventions that may be encountered in this document:

Table 2 Typographic Conventions

Convention	Description	Example
Code Examples	Code examples	<code>stat char* months[] ={"Jan", "Feb"}</code>



Convention	Description	Example
Command Variables	You need to supply the values within the <>	<home_directory>
Document and File Names	References to document titles or sections in a document and file names	For more information, refer to the System Administrator Guide. Check the local runlog files (xxx.runlog and xxa.runlog) in the /var/log/xxx directory.
GUI Objects	GUI objects, such as menus, fields, and buttons, dialog boxes, and options	On the File menu, click Exit .
Key Combinations	Key combinations	Press Ctrl+X to delete the selected value. ⁽¹⁾
Output Information	Text displayed by the system	System awaiting input
Parameter/Configuration Values	Parameter values (numbers, true/false, yes/no, and so on)	To use this feature, the parameter must be set to true
System Elements	Command and parameter names, program names, path names, URLs, and directory names	The files are located in E:\Test The files are located in /etc/opt/ericsson/bin. ⁽²⁾



Convention	Description	Example
User Input	In this document when you are required to input content, the input content is displayed using this bold mono-spaced font. The content must be added exactly as shown.	cd \$HOME
Line Break	The arrow symbol (\Rightarrow) can be used when an inappropriate line break has been made. An inappropriate line break occurs when the code lines are too long to fit on the page, and there is no appropriate place for a line break.	<code>cd /opt/msmw-cds-\Rightarrow cxp-<version> (3)</code>

(1) The plus sign (+) indicates that you must press the keys simultaneously.

(2) The use of the forward slash (/) is for Linux and UNIX systems; Windows systems use the backslash (\).

(3) The use of the \Rightarrow symbol (character entity \Rightarrow) at the end of a line has a meaning to the human reader, but if copied and pasted from a CPI document to a command line interpreter the symbol must be cut from the code.

1.4 Prerequisites

PDB has been successfully installed and configured.

For installation, upgrade, and rollback procedures, refer to the Parameter Database (PDB) Installation Instructions (Reference [1]).

1.5 Comments About the Documentation

Ericsson encourages you to provide feedback, comments, or suggestions so that we can improve the documentation to better meet your needs. With your comments, provide the following:

- Document title
- Document number and revision
- Page number

Please send your comments to your local Ericsson Support.



2 Overview

PDB is a JAVA application that runs on top of the GlassFish application server. Configuration data is stored in a MySQL database running on the same server.

System administrators can interact with PDB using specific administration windows inside the PDB GUI or by directly accessing the server over SSH.

2.1 PDB GUI

PDB provides administration windows within the Graphical User Interface (GUI) to manage the parameter database.

After logging in, the PDB home page is displayed. The home page shows the PDB welcome message or system notifications. See Figure 1.

System notifications are messages drafted by a PDB system administrator to communicate important information to users. For more information on how to configure system notifications, refer to Section 6 on page 26.

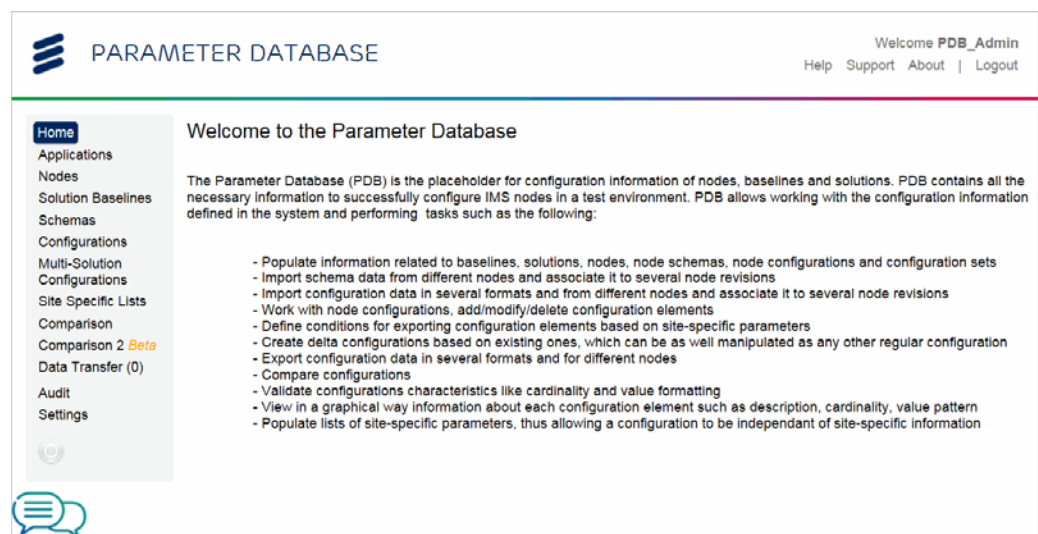


Figure 1 The PDB Home Page

System administrators can access all parts of the PDB GUI using the menu options on the left side of the window.

The following table describes the PDB administrative interfaces:



Table 3 PDB Administrative Interfaces

Interface	Function
Audit	Allows system administrators to review a log of PDB user activities. For more information, see Section 5 on page 22.
Settings	Allows administration of the PDB GUI. For more information, see Section 4 on page 19.

2.2 PDB Server

Most administration tasks are performed directly on the PDB server.

The following table describes the administration tasks:

Table 4 PDB Administrative Interfaces

Task	Description
Configuring System Notifications	Describes how to write system notifications to communicate important information to PDB users.
Working with PDB Logs	Describes the structure of the GlassFish server.log and how to interact with the file.
Starting and Stopping GlassFish and the PDB application	Describes how to stop, start, and restart GlassFish and the PDB application.
Backing Up PDB	Describes how to create backups of the MySQL databases.
Restoring PDB	Describes how to restore backups and recover PDB.

3 User Accounts

Logging in to PDB requires a valid user account. These accounts are provisioned in the IMSREF Centralized User Database (CUDB) or an external LDAP server.



Network connectivity is required for PDB user authentication. If a connection to the authentication server cannot be established, users are denied access to the PDB GUI. The type of user authentication (CUDB or LDAP) is set by a GlassFish authentication realm. The authentication realm must be configured during PDB installation. If PDB has already been installed, it must be reinstalled to make changes.

For CUDB:

- To configure PDB for CUDB authentication, refer to Parameter Database (PDB) Installation Instructions (Reference [1]).
- To provision new users, refer to the IMSREF Centralized User Database (CUDB) System Administration Guide (Reference [3]).

For LDAP:

- To configure PDB for LDAP authentication, refer to Section 3.2 on page 10.
- To provision new users, contact your LDAP administrator.

All PDB user accounts must be provisioned with the necessary access rights. For more information on access control, refer to Section 3.1 on page 7.

3.1 Access Control

Access rights and permissions are granted to user accounts by assigning them to one or more groups. These groups designate the role of member accounts on the system. The permissions granted by group membership are cumulative, thus membership in two-or-more groups grants all of the access rights assigned by each group.

This section describes the mapping between PDB roles and user groups in CUDB. The role mapping must be customized under LDAP authentication. For more information on mapping the names of groups in the LDAP directory to specific roles defined in PDB, refer to Section 3.2.2 on page 13.

Under CUDB authentication, all non-administrator users must be a member of the `pdb_users` group in order to access the PDB GUI. This group grants unrestricted, read-only access to the web interface, allowing users to perform all operations that do not write to the PDB database. In order to work with PDB data, users provisioned in CUDB require additional permissions. These permissions are granted through membership in other PDB groups. For more information on working with user accounts in CUDB, refer to the IMSREF Centralized User Database (CUDB) System Administration Guide (Reference [3]).

The following table describes the PDB roles-to-groups mapping in CUDB.



Table 5 PDB User Groups

Group	Role	Description
pdb_administrators	admin	PDB Administrators have unrestricted access to the PDB GUI.
pdb_application_editor	application_editor	<p>PDB Application Editors are granted full access to the Applications page.⁽¹⁾</p> <p>For more information on the operations available through the Applications page, see Managing Applications in the PDB User Guide (Reference [2]).</p>
pdb_baseline_editor	baseline_editor	<p>PDB Baseline Editors are granted full access to the Baselines page.⁽¹⁾</p> <p>For more information on the operations available through the Baselines page, see Working with Baselines in the PDB User Guide (Reference [2]).</p>
pdb_configuration_editor	configuration_editor	<p>PDB Configuration Editors are granted full access to the Configurations page for all configurations except those tagged as MPVL.⁽¹⁾</p> <p>For more information on the operations available through the Configurations page, see Node Configurations in the PDB User Guide (Reference [2]).</p>
pdb_configuration_set_editor	configuration_set_editor	<p>PDB Configuration Set Editors are granted full access to the Configuration Sets page.⁽¹⁾</p> <p>For more information on the operations available through the Configuration Sets page, see Working with Configuration Sets in the PDB User Guide (Reference [2]).</p>



Table 5 PDB User Groups

Group	Role	Description
pdb_mpvl_configuration_editor	mpvl_configuration_editor	<p>PDB MPVL Configuration Editors are granted full access to the Configurations page for all configurations including those tagged as MPVL.⁽¹⁾</p> <p>For more information on the operations available through the Configurations page, see Node Configurations in the PDB User Guide (Reference [2]).</p>
pdb_mpvl_parameter_editor	mpvl_parameter_editor	<p>PDB MPVL Parameter Editors are granted write access to the Parameters page for all configurations including those tagged as MPVL.⁽¹⁾</p> <p>For more information on the operations available through the Parameters page, see Managing Configuration Data in the PDB User Guide (Reference [2]).</p>
pdb_node_editor	node_editor	<p>PDB Node Editors are granted full access to the Nodes page.⁽¹⁾</p> <p>For more information on the operations available through the Nodes page, see Managing Nodes in the PDB User Guide (Reference [2]).</p>
pdb_parameter_editor	parameter_editor	<p>PDB Parameter Editors are granted write access to the Parameters page for all configurations except those tagged as MPVL.⁽¹⁾</p> <p>For more information on the operations available through the Parameters page, see Managing Configuration Data in the PDB User Guide (Reference [2]).</p>



Table 5 PDB User Groups

Group	Role	Description
pdb_schema_editor	schema_editor	<p>PDB Schema Editors are granted full access to the Schemas page.⁽¹⁾</p> <p>For more information on the operations available through the Schemas page, see Working with Schemas in the PDB User Guide (Reference [2]).</p>
pdb_site_specific_list_editor	site_specific_list_editor	<p>PDB Site-Specific List Editors are granted full access to the Site Specific Lists page, excluding the Global Variables tab.⁽¹⁾</p> <p>For more information on the operations available through the Site Specific Lists page, see Working with Site-Specific Lists in the PDB User Guide (Reference [2]).</p>
pdb_users	user	<p>PDB Users have unrestricted, read-only access to the PDB GUI. They are able to perform all operations that do not write to the PDB database.</p>
pdb_variable_editor	variable_editor	<p>PDB Variable Editors are granted full access to the Global Variables tab on the Site Specific Lists page.⁽¹⁾</p> <p>For more information on the operations involving Global Variables, see Global Variables in the PDB User Guide (Reference [2]).</p>

(1) All users, except PDB Administrators, must belong to the `pdb_users` group in order to access the PDB GUI.

3.2 Configuring GlassFish and the PDB Server to Authenticate Against an LDAP Directory

PDB can be configured to authenticate against an external LDAP directory. Configuring PDB for LDAP authentication is a two step process that includes the following steps:



- Configuring the GlassFish Authentication Realm
- Mapping PDB Roles to LDAP Groups

3.2.1 Configuring the GlassFish Authentication Realm

To authenticate against an LDAP directory, a GlassFish authentication realm must be configured to use the `LDAPRealm` class. The properties of this class must be configured to match the directory structure of the LDAP directory server.

3.2.1.1 Prerequisites

To perform the realm configuration you need the following:

- Access to the GlassFish Administration GUI
- A valid LAN user ID and password
- SSH access to the PDB server
- The PDB EAR file used to install the PDB application
- In addition, you may also need an LDAP browser to access the LDAP directory.

3.2.1.2 LDAP Properties and Settings

Authenticating against a LDAP server on a given domain requires user credentials for the PDB server and the names of LDAP groups that can be mapped to the different PDB user roles. This information is typically provided by the local Network Administrator.

The following parameter values are assumed for the purpose of this procedure. These values are meant to be used by PDB server.

Name	pdb-auth-realm
Class Name	com.sun.enterprise.security.auth.realm.ldap.LDAPRealm

The following properties are specific to this class:

JASS context	ldapRealm
Directory	ldap://eamcs.ericsson.se:3268
Base DN	DC=eamcs,DC=ericsson,DC=se

**Assign Group**

Leave empty or specify a group to be assigned by default (Such as `pdb_users`). If a group is assigned, then by default the users will have access to PDB as if they belong to that group. PDB usage restrictions will be based on this default group and its role mapping. If Assign group is left empty, users will have access to PDB based on the LAN groups that they belong to and the role mappings that will be defined later in this section.

Additional properties:

search-bind-dn `pdb@eamcs.ericsson.se`

search-bind-password `pdb123`

search-filter `sAMAccountName=%s`

group-search-filter `member=%d`

group-target `cn`

Note: The ldap properties, user ID and password are typically provided by your local Administrator.

3.2.1.3**Modifying PDB Authentication Realm**

A `pdb-auth-realm` is defined during maiden installation and is currently in use by PDB. In order to modify it for LDAP authentication, the existing realm must be deleted and recreated.

Caution!

System Downtime

Modifying the PDB authentication realm requires system downtime.

In general, creating an authentication realm using the GlassFish CLI uses the following syntax:

```
glassfish@pdb> asadmin create-auth-realm --classname
com.sun.enterprise.security.auth.realm.ldap.LDAPRealm --property
"jaas-context=ldapRealm:directory=<Directory>:base-dn=<Base
DN>:search-filter=<search-filter>:search-bind-password=<search-bin
d-password>:group-target=cn:group-search-filter=<group-search-filt
er>:search-bind-dn=<search-bind-dn>" pdb-auth-realm
```

To create an LDAP authentication realm using the GlassFish CLI:



1. Using SSH, connect to the PDB server as root.
2. Switch to the glassfish user.

```
su glassfish
```

3. Delete the pdb-auth-realm:

```
glassfish@pdb> asadmin delete-auth-realm pdb-auth-realm
```

4. Recreate the authentication realm.

The following is a concrete example based on the LDAP properties and settings defined in Section 3.2.1.2 on page 11.

```
glassfish@pdb> asadmin create-auth-realm --classname com.sun.enterprise.security.auth.realm.ldap.LDAPRealm --property "jaas-context=ldapRealm:directory=ldap://eamcs.ericsson.se:3268:base-dn=DC\=eamcs,DC\=ericsson,DC\=se:search-filter=sAMAccountName\=%s:search-bind-password=pdb123:group-target=cn:group-search-filter=member\=%d:search-bind-dn=pdb@eamcs.ericsson.se" pdb-auth-realm
```

5. Switch back to the root user.

```
exit
```

3.2.2 Mapping PDB Roles to LDAP Groups

In PDB, user access rights and permissions are defined by roles. Each role is associated to a number of user groups. By mapping back to specific roles, these groups grant access rights and permissions to the members. For more information on PDB user groups, refer to Section 3.1 on page 7.

When enabling LDAP authentication, the default roles-to-groups mapping must be modified to point to the new LDAP groups.

The mapping between PDB roles and LDAP groups is configured in a deployment descriptor file called sun-web.xml.

Modifications to sun-web.xml only take effect during application deployment. If the PDB application has already been deployed, it must be re-deployed with the modified sun-web.xml before any changes will apply.

The following variables are used to configure the application.

Table 6 PDB Variables

Variable	Use	Provided By	Example
PDB_REVISION	PDB server revision	PDB release notes	2.0-R2B01



Table 6 PDB Variables

Variable	Use	Provided By	Example
PDB_SERVER_DIST_ZIPFILE	PDB server distribution bundle	PDB release notes	pdb-server-dist-2.0-R2B01-pdb-server-dist.zip
PDB_TEMP_PATH	Temporary location of release files	Installer	/tmp/pdb

The `<pdb_revision>` and `<pdb_server_tmp_path>` variables must be defined before they can be used in the subsequent sections of this document.

To define variables, execute the following commands while logged into the PDB server as root:

```
root@pdb# export PDB_TEMP_PATH=<pdb_server_tmp_path>
```

```
root@pdb# export PDB_REVISION=<pdb_revision>
```

To configure the PDB application to use LDAP authentication perform the following steps while logged into the PDB server as glassfish:

Unpacking sun-web.xml

1. Copy the PDB_SERVER_DIST_ZIPFILE onto the server and uncompress it to the PDB_TEMP_PATH directory.
2. Extract the `<pdb_gui_war_file>` from the `<pdb_ear_file>`.

```
glassfish@pdb> cd ${PDB_TEMP_PATH}/pdb-server-dist-${PDB_REVISION}
```

```
glassfish@pdb> jar xvf pdb-ear-${PDB_REVISION}.ear  
pdb-gui-${PDB_REVISION}.war
```

3. Extract the sun-web.xml file from the `<pdb_gui_war_file>`.

```
glassfish@pdb> jar xvf pdb-gui-${PDB_REVISION}.war  
WEB-INF/sun-web.xml
```

The file is extracted to a new WEB-INF directory.

Modifying the Roles-to-Groups Mapping

4. Navigate to the WEB-INF directory.

```
glassfish@pdb> cd ${PDB_TEMP_PATH}/pdb-server-dist-${PDB_REVISION}/WEB-INF/
```

5. Make a backup of the sun-web.xml file.

```
glassfish@pdb> cp sun-web.xml sun-web.xml.orig
```



6. Open the sun-web.xml file for editing.

```
glassfish@pdb> vi sun-web.xml
```

sun-web.xml controls the privileges users will have when accessing the PDB application by mapping the names of groups on the LDAP directory to specific roles defined in PDB.

LDAP group names are typically provided by local Network Administrator. These LDAP groups must be mapped to Java EE roles in the deployment descriptor.

PDB has a number of predefined roles (such as admin) that can be mapped to one or more user groups (such as pdb_administrators). For more information on PDB roles and user groups, refer to Section 3.1 on page 7.

sun-web.xml must be edited to add new groups to the different roles. Existing groups should remain; new groups can be added to the same role.

Note: The group name must match the CN attribute of the group on the LDAP directory.

The following example shows a sample configuration where a new <group-name> was added to the admin, user, application-editor and node-editor roles.



```
glassfish@pdb> jar uf pdb-ear-${PDB_REVISION}.ear
pdb-gui-${PDB_REVISION}.war
```

Deploy the PDB Application

10. Deploy the updated .ear file.

Note: If PDB is currently deployed it must be undeployed before performing this step.

Caution!

System Downtime

Redeploying the PDB application requires system downtime.

```
glassfish@pdb> asadmin deploy --name pdb pdb-ear-${PDB_REVISION}.ear
```

The PDB application is now configured to use LDAP authentication.

11. Test PDB LDAP authentication.

Connect to the PDB GUI and attempt to log in using a LAN user ID and password that belongs to one of the LDAP groups mapped in the sun-web.xml file.

`http://<pdb-server-address>:8080/pdb`

3.2.3 Rollback Procedure

Rolling back LDAP authentication and the roles-to-groups mapping consists of:

- Recreating the `pdb-auth-realm`.
- Undeploying the modified .ear distribution file.
- Redeploying the PDB application using the original, unmodified .ear file.

Note: An unmodified distribution file for the PDB server is required to rollback LDAP authentication. If needed, a link to download the PDB distribution can be found in the PDB Release Notes for your version of the product.

- Restarting GlassFish.

This procedure assumes that PDB was authenticating against CUDB before the switch to LDAP authentication. The authentication realm will be recreated to point to CUDB.



Caution!

System Downtime

The rollback procedure requires system downtime.

The rollback procedure uses the same variables defined in Section 3.2.2 on page 13. If you are using the same terminal session as the modification procedure, then these variables are already defined.

To rollback to the original PDB server distribution:

1. Using SSH, connect to the PDB server as root.
2. Switch to the glassfish user.

```
su glassfish
```

3. Delete the pdb-auth-realm:

```
glassfish@pdb> asadmin delete-auth-realm pdb-auth-realm
```

4. Recreate the authentication realm to point to CUDb:

```
glassfish@pdb> asadmin create-auth-realm --classname com.sun.enterprise.security.auth.realm.jdbc.JDBCRealm --property jaas-context=jdbcRealm:datasource-jndi=cudb-data-source:user-table=USERS_VIEW:user-name-column=user_id:password-column=password:group-table=USERS_GROUPS_VIEW:group-name-column=group_id:digest-algorithm=md5 pdb-auth-realm
```

5. Undeploy the existing PDB application.

```
glassfish@pdb> asadmin undeploy pdb
```

6. Deploy an unmodified PDB distribution file.

```
glassfish@pdb> asadmin deploy --name pdb ${PDB_TEMP_PATH}/pdb-server-dist-${PDB_REVISION}/pdb-ear-*.ear
```

7. Switch back to the root user.

```
exit
```

8. Restart GlassFish.

```
service glassfish restart
```

LDAP authentication has been rolled back.



Once you have verified that the PDB server is working properly, temporary or old files and directories can be removed.

4 Configuring PDB

Some behavior of the PDB GUI is controlled by configuration parameters. These parameters are managed through the GUI on the **Settings** window. PDB is provisioned with all supported parameters at installation time. Working with PDB settings involves adjusting the values of these default parameters.

Note: The capacity to add or remove configuration parameters is reserved for future use.

To adjust a PDB setting:

1. In the PDB GUI, select **Settings** from the menu options on the left.

The **Settings** interface is displayed. See Figure 2.

<div> <input type="button" value="Refresh"/> <input type="button" value="New"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/> </div>		
Name	Value	Description
FEEDBACK_URL	mailto:PDLMISREF3@ex1.eamcs.ericsson.se?Subject=My%20Feedback	Feedback link URL
HELP_URL	http://calstore.internal.ericsson.com/alexserv?li=EN/LZN7650048/3R1A	Help link URL
MAINTENANCE_MSG	The site is currently down for maintenance. We will be back soon. Thanks for your patience.	Message to display when the site is down for maintenance.
PREVENT_LOGIN	false	Prevent non-admin users from logging in. Set to true to activate
SUPPORT_URL	https://ericoll.internal.ericsson.com/SITES/IMSTOOL/IMSREF/default.aspx	Support link URL


Figure 2 PDB Settings

The following table describes the different elements forming the **Settings** interface:

Table 7 The Settings Interface

Element	Description
<input type="button" value="New"/>	Adds a new settings parameter.
<input type="button" value="Edit"/>	Opens the selected settings parameter for modification.





Element	Description
	Deletes the selected settings parameter from the database.
Name	The parameter name. Cannot be modified.
Value	The parameter value.
Description	A short description of the parameter. Cannot be modified.

2. Select a parameter to modify from the table. For a detailed description of each parameter, refer to Section 4.1 on page 20.

The **Edit** button becomes available.

3. Click **Edit**.

The selected parameter is opened in edit mode. See Figure 3.

Settings
 

Name ▲	Value	Description
FEEDBACK_URL	mailto:PDLIMSREF3@ex1.eamcs.ericsson.se?Subject=My%20Feedback	Feedback link URL
HELP_URL	<input type="text" value="http://calstore.internal.ericsson.com/alexserv?LI=EN/LZN7650072/14R1A"/>	Help link URL
MAINTENANCE_MSG	The site is currently down for maintenance. We will be back soon. Thanks for your patience.	Message to display when the site is down for maintenance.
PREVENT_LOGIN	false	Prevent non-admin users from logging in. Set to true to activate
SUPPORT_URL	https://ericoll.internal.ericsson.com/SITES/IMSTOOL/IMSREF/default.aspx	Support link URL

Figure 3 Edit Settings

4. Modify the parameter value as required.
5. Click **Apply**.

The updated setting is added to the database.

4.1 PDB Settings Parameters

The following configuration parameters are used by PDB:

Table 8 PDB Settings Parameters

Parameter
FEEDBACK_URL



Parameter
HELP_URL
MAINTENANCE_MSG
PREVENT_LOGIN
SUPPORT_URL

4.1.1 FEEDBACK_URL

Parameter Name	FEEDBACK_URL
Description	This parameter sets a URL for IDEAS FEEDBACK. This link is fetched in a new window when clicking IDEAS FEEDBACK in PDB.
Allowed values	String (URL)
Default value	<URL for IDEAS FEEDBACK>

4.1.2 HELP_URL

Parameter Name	HELP_URL
Description	This parameter sets a URL for the Parameter Database (PDB) Application CPI library. This link is fetched in a new window when clicking Help from the menu options on the left.
Allowed values	String (URL)
Default value	<URL for the current ALEX Library>

4.1.3 MAINTENANCE_MSG

Parameter Name	MAINTENANCE_MSG
Description	This parameter sets a message that will be displayed on the Parameter Database Login page while the system is locked for maintenance. This notice is only displayed while PREVENT_LOGIN is set to true.
Allowed values	String
Default value	"The site is currently down for maintenance. We will be back soon. Thanks for your patience."



4.1.4 PREVENT_LOGIN

Parameter Name

PREVENT_LOGIN

Description

This parameter controls end-user access to PDB. Setting PREVENT_LOGIN to true locks the PDB GUI. Locking the GUI prevents end-users from logging in.

Note: Users that are already connected to PDB when PREVENT_LOGIN is initiated can continue using the system until they log out.

PREVENT_LOGIN allows you to restrict traffic on the PDB server during maintenance activities. While the GUI is locked, the standard **Parameter Database Login** page is disabled. The MAINTENANCE_MSG parameter can be used to set a notice message that is displayed on the login page.

System administrators are never locked out of the PDB GUI. To access the system, administrators can click a link for maintenance access at the bottom of the login page.

Allowed values

Boolean (true/false)

Default value

false

4.1.5 SUPPORT_URL

Parameter Name

SUPPORT_URL

Description

This parameter sets a URL for the official PDB Support and Maintenance web page. This link is fetched in a new window when clicking **Support** in the menu options on the left.

Allowed values

String (URL)

Default value

<URL for PDB Support and Maintenance>

5 Auditing PDB

PDB keeps a detailed log of user activity on the system. This log is available to system administrators through the **Audit** interface in the PDB GUI. Only those



user operations that write to the database are captured by the audit log. Each entry includes the specifics of the operation and the end result. The audit log can be filtered by a number of different search criteria and reviewed online or exported in CSV format.

To review the audit log:

1. In the PDB GUI, select **Audit** from the menu options on the left.

The **Audit** interface is displayed. See Figure 4.

Note: The audit log is empty until you perform a search.


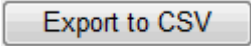
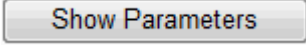
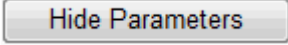
Figure 4 PDB Audit

The following table describes the different elements of the **Audit** interface:

Table 9 The Audit Interface

Element	Description
Start Date	Filter the audit log for operations that took place after the selected date. Click to select a date using the calendar. Dates must have the form <DD/MM/YYYY HH:mm> .
End Date	Filter the audit log for operations that took place before the selected date. Click to select a date using the calendar. Dates must have the form <DD/MM/YYYY HH:mm> .
User (List)	Filter the audit log by PDB user.
Service (List)	Filter the audit log by PDB service.



Element	Description
Operation (List)	Filter the audit log by PDB operation.
Parameters	Filter the audit log for a specified database parameter. Partial matches are accepted.
	Filters the audit log using the selected criteria. If no criteria are selected, the complete log is displayed.
	Exports the audit log in CSV format.
 / 	Controls the display of database parameters in the audit log.

2. Filter the audit log by performing a search using the available fields. See Table 9. All search criteria are optional. Leave these fields blank to display the last 200 entries.

Note: All search results are limited to 200 entries.

3. Click **Apply**.

The audit log is displayed. See Figure 5.



Timestamp ▼	User	Service	Operation	Result
2010-02-03 13:15:58.0	admin	BaselineService	createBaseline	success
2010-02-03 13:15:43.0	admin	BaselineService	createBaselineRevision	success
2010-02-03 13:15:31.0	admin	BaselineService	createBaseline	success
2010-02-03 13:15:16.0	admin	NodeService	updateNode	success
2010-02-03 13:14:55.0	admin	NodeService	createNodeRevision	success
2010-02-03 13:14:42.0	admin	NodeService	createNode	success
2010-02-03 13:14:27.0	admin	NodeService	createNode	success
2010-02-03 13:14:12.0	admin	NodeService	createNode	success
2010-02-03 13:13:57.0	admin	NodeService	createNode	success
2010-02-03 13:13:41.0	admin	NodeService	createNode	success
2010-02-03 13:13:26.0	admin	NodeService	updateApplicationType	success
2010-02-03 13:13:14.0	admin	NodeService	createApplicationType	success
2010-02-03 13:13:03.0	admin	NodeService	createApplicationType	success
2010-02-03 13:12:52.0	admin	NodeService	createApplicationType	success
2010-02-03 13:12:42.0	admin	NodeService	createApplicationType	success
2010-02-03 13:12:32.0	admin	NodeService	createApplicationType	success

Figure 5 Audit Log

The following table describes the different fields of the audit log:

Table 10 Audit Log Fields

Field	Description
Timestamp	Timestamp for the user activity.
User	The user who triggered the activity.
Service	The service associated with the activity.
Operation	The operation associated with the activity.
Result	The result of the activity. Failed results include a tooltip that explains the reason(s) for the failure.
Parameters	Database parameters that were modified by the activity. Note: This field is hidden until you press Show Parameters .



Note: Output from the audit log, as it appears on screen, can be exported in CSV format.

To export the audit log, click **Export to CSV**.

Your web browser will prompt you to open or save the file.

6 Configuring System Notifications

System notifications are messages posted by PDB system administrators to communicate important information. Once users have logged in to PDB, these messages are displayed on the **Home** page.

System notifications replace the default PDB welcome screen and are presented in a list where multiple notifications can be displayed simultaneously. Figure 6 shows sample notification messages.

Welcome to the Parameter Database		
Notification Message 1		
by the PDB team		2011-07-18 18:00
This is a sample message.		
Notification Message 2		
by the PDB team		2011-07-18 19:34
This is a sample message.		

Figure 6 System Notifications

Each notification is composed of a header and a message body. The header includes the message title, author and a timestamp while the body includes the message text.

While working with PDB, users can review the notifications at any time by returning to the **Home** page.

System notifications are written in a file called notifications.xml, located under /usr/local/glassfish/domains/domain1/config/pdb/ on the PDB server.

The file and directory are not created automatically during PDB installation and must be created manually. Use the following example as a template to create a new file, if required.



```
<?xml version='1.0' encoding='UTF-8' standalone='yes'?>
<notifications>
  <subject>
    <value>
      <value>
        <value>
          <value>
            <value>
              <value>
                <value>
                  <value>
                    <value>
                      <value>
                        <value>
                          <value>
                        </value>
                      </value>
                    </value>
                  </value>
                </value>
              </value>
            </value>
          </value>
        </value>
      </value>
    </value>
  </subject>
  <body>
    <value>
      <value>
        <value>
          <value>
            <value>
              <value>
                <value>
                  <value>
                    <value>
                      <value>
                        <value>
                          <value>
                        </value>
                      </value>
                    </value>
                  </value>
                </value>
              </value>
            </value>
          </value>
        </value>
      </value>
    </value>
  </body>
  <userId>
    <value>
      <value>
        <value>
          <value>
            <value>
              <value>
                <value>
                  <value>
                    <value>
                      <value>
                        <value>
                          <value>
                        </value>
                      </value>
                    </value>
                  </value>
                </value>
              </value>
            </value>
          </value>
        </value>
      </value>
    </value>
  </userId>
  <date>
    <value>
      <value>
        <value>
          <value>
            <value>
              <value>
                <value>
                  <value>
                    <value>
                      <value>
                        <value>
                          <value>
                        </value>
                      </value>
                    </value>
                  </value>
                </value>
              </value>
            </value>
          </value>
        </value>
      </value>
    </value>
  </date>
  <expiryDate>
    <value>
      <value>
        <value>
          <value>
            <value>
              <value>
                <value>
                  <value>
                    <value>
                      <value>
                        <value>
                          <value>
                        </value>
                      </value>
                    </value>
                  </value>
                </value>
              </value>
            </value>
          </value>
        </value>
      </value>
    </value>
  </expiryDate>
</notifications>
```

Example 2 notifications.xml Template

notifications.xml contains the following configuration parameters:

Table 11 notifications.xml Parameters

Parameter	Description	Mandatory
subject	Sets the message title.	Yes
body	Sets the message text.	Yes
userId	Sets the message author.	Yes
date	Sets a start date for the notification.	Yes
expiryDate	Sets an expiration date for the notification	No

Note: XML does not immediately support special characters. To use special characters, including quotes and the ampersand, enclose the text inside a `<![CDATA[]]>` tag.

Caution!

A faulty or incomplete notifications.xml file prevents notifications from being displayed.

To add a new system notification:

1. Use SSH to log in to the PDB server as glassfish.
2. Navigate to `/usr/local/glassfish/domains/domain1/config/pdb/`:
`cd /usr/local/glassfish/domains/domain1/config/pdb/`
3. Make a backup copy of notifications.xml:
`cp notifications.xml notifications.<version>`



4. Open notifications.xml for modification:

```
vi notifications.xml
```

5. Following standard XML format and encapsulation, add a new notification:

```
<notification>
  <subject>Message Title</subject>
  <body>Message Text</body>
  <userId>Author</userId>
  <date><YYYY-MM-DD>T<HH:MM:SS><GMT Offset></date>
</notification>
```

You can modify existing notifications by changing the values.

Note: Notifications that are dated in the future will not be displayed in PDB until the set date has passed.

6. Optionally, set an expiration for the notification message by adding an expiryDate parameter within the notification tags:

```
<expiryDate><YYYY-MM-DD>T<HH:MM:SS><GMT Offset></expiryDate>
```

Note: Expired messages are not displayed in PDB.

7 PDB Logging

PDB logs key events to the server.log file, making it indispensable for monitoring the system and troubleshooting problems.

Note: server.log is the log file for the GlassFish AS. Since GlassFish is not used exclusively by PDB, the server.log file can contain information from other deployed applications that are not relevant to the PDB software.

In a standard installation, server.log is located in the following directory:

```
/usr/local/glassfish/domains/domain1/logs/
```

server.log records information in the following format:

```
[#|timestamp|logging_level|app_server|java_class|thread|message|#]
```

Note: The file uses pipe-separation to delimit information.

The following fields are of interest to the typical system administrator:



2nd field	timestamp
3rd field	logging_level (SEVERE ERROR WARNING INFO FINE FINER FINEST)
7th field	message

The following is an example of a printout from the server.log showing only these fields of interest:

```

...
2009-06-25T16:15:27.155-0400|INFO|...|...|...|commiting transaction
                                com.sun.enterprise.distributedtx.UserTransactionImpl@30351201
2009-06-25T16:15:28.010-0400|INFO|...|...|...|starting transaction
                                com.sun.enterprise.distributedtx.UserTransactionImpl@30351201
2009-06-25T16:15:28.112-0400|INFO|...|...|...|calling export service with:
                                config: b83784c7-7c01-4389-a367-f93ecde25980
                                ssl: 9f5c75e6-3cdd-4469-82fd-47215a6f408e
                                format:LDAP
2009-06-25T16:15:28.989-0400|INFO|...|...|...|commiting transaction
                                com.sun.enterprise.distributedtx.UserTransactionImpl@30351201
...

```

Example 3 Sample Printout from server.log

It is possible to view and revise the server.log file in several ways:

- Using UNIX Commands
- Using the GlassFish Administration GUI

7.1 Checking the Log File Using the GlassFish Administration GUI

The GlassFish administration GUI can be used to display the server.log file.

To view server.log in GlassFish:

1. Using a web browser, open the GlassFish administration GUI that is running on PDB.

http://<PDB_IP_ADDRESS>:4848

Note: The GUI is listening on port 4848.

The login screen appears.

2. Log in using the admin user name and password.

User Name admin

Password adminadmin

The **Common Tasks** window opens.



3. Click **Search Log Files** under Other Tasks.

A new window opens.

4. Click the **Advanced Search** link near the top of the window.
5. Enter the following text in the **Custom Logger** field:

```
com.ericsson.imsref.pdb
```

6. Click **Search**.

The search results appear on screen.

7. Scroll down to review the results.

7.2 Defining Component Log Levels for PDB

The GlassFish administration GUI can be used to define log levels for any component running in the environment. To prevent the server.log file from being filled with non-relevant information, the log levels of the components can be modified.

To modify the log levels:

1. Using a web browser, open the GlassFish administration GUI that is running on PDB.

```
http://<PDB_IP_ADDRESS>:4848
```

Note: The GUI is listening on port 4848.

The login screen appears.

2. Log in using the admin user name and password.

User Name	admin
------------------	-------

Password	adminadmin
-----------------	------------

The **Common Tasks** window opens.

3. Click **Application Server** in the menu tree on the left panel.

Application Server window opens.

4. Click the **Logging** tab near the top of the window.

Logger Settings opens.

5. Click the **Log Levels** tab.



6. Scroll down to the **Additional Properties** table, and click the **Add Property** button.

A new row appears.

7. Define the component log levels.

The following options are available:

- SEVERE
- ERROR
- WARNING
- INFO
- FINE
- FINER
- FINEST

The more sensitive log levels impact system performance, but are necessary when troubleshooting the system.

Note: The values in the following table are recommended in the Parameter Database Installation Instructions, 1/1531-CXP 902 0212, but can be modified as required.

Table 12 Log Level Properties

Name	Value
com.maverick	WARNING
com.sshools	WARNING
com.ericsson.imsref.pdb	FINE

8. Click **Save**.

The modifications are saved.

9. Use SSH to log in to the PDB server as root.

10. Restart the Glassfish AS by executing the following command:

```
service glassfish restart
```

The GlassFish application restarts.



8 Starting and Stopping PDB

These starting and stopping procedures detail how to manage the proper startup and shutdown of PDB and the GlassFish Application Server (AS).

Note: Do not perform any of these procedures unless you know how the system will be affected or you have been instructed to do so by PDB support.

8.1 Stopping the GlassFish Application

During operation of the system it may become necessary to stop the GlassFish Application Server. Stopping GlassFish halts all services provided by PDB and other deployed applications.

To stop GlassFish:

1. Use SSH to log in to the PDB server as root.
2. Execute the following command:

```
service glassfish stop
```

The GlassFish application stops.

8.2 Starting the GlassFish Application

PDB services require that GlassFish is up and running.

To start GlassFish:

1. Use SSH to log in to the PDB server as root.
2. Execute the following command:

```
service glassfish start
```

The GlassFish application starts.

8.3 Restarting the GlassFish Application

When GlassFish hangs or becomes unresponsive, it may need to be restarted. Before restarting GlassFish to correct a fault, contact PDB support for help with troubleshooting.

To restart GlassFish:

1. Use SSH to log in to the PDB server as root.
2. Execute the following command:



```
service glassfish restart
```

The GlassFish application restarts.

8.4 Stopping the PDB Application on GlassFish

GlassFish is not exclusive to PDB and can run other applications. If PDB must be stopped, the software can be brought down directly without interrupting the application server.

The PDB application is stopped using the GlassFish `asadmin` command.

To stop the PDB application:

1. Use SSH to log in to the PDB server as `glassfish`.
2. Execute the following command:

```
asadmin disable <pdb application>
```

The PDB application stops.

8.5 Starting the PDB Application on GlassFish

The PDB application is started using the GlassFish `asadmin` command.

To start the PDB application:

1. Use SSH to log in to the PDB server as `glassfish`.
2. Execute the following command:

```
asadmin enable <pdb application>
```

The PDB application starts.

8.6 Checking the GlassFish Application Server

PDB is an application that runs on top of the GlassFish Application Server. If the PDB software fails to start, verify that GlassFish is running properly.

To verify the status of GlassFish:

1. Use SSH to log in to the PDB server as `root`.
2. Execute the following command:

```
service glassfish status
```

The status of the GlassFish is displayed on screen:

```
domain status:
```



```
domain1 running
```

```
running
```

If GlassFish is not running, attempt to start it by following the instructions in Section 8.2 on page 32. If GlassFish fails to start, contact PDB support.

8.7 Starting GlassFish Admin Console

It is possible to administer GlassFish through the GlassFish Admin Console. The Admin Console allows you to manage users and control resources.

Note: GlassFish must be up and running before you can start the Admin Console.

To launch the Admin Console:

1. Using a web browser, open the Admin Console that is running on PDB.

```
http://<PDB_IP_ADDRESS>:4848
```

Note: The Admin Console is listening on port 4848.

The Admin Console login window opens.

2. Enter a valid user name and password.

For more information, refer to the IMSREF Parameter Database (PDB) Installation Instructions, 1/1531-CXP 902 0212.

3. Click **Login**.

The Admin Console is displayed in your browser window.

For more information on the GlassFish Admin Console, refer to documentation on the Oracle® web site:

<http://www.oracle.com/technetwork/middleware/glassfish/documentation/index.html>

9 Backing Up PDB

PDB data must be backed up to external storage on a regular basis. In the event of data loss, these backups are essential to restoring the system.



To facilitate routine backups, a daily, automated backup of the MySQL databases is configured as part of the PDB installation procedure. For more information on the automated backup, refer to Section 9.1 on page 35.

As outlined in the Parameter Database (PDB) Installation Instructions (Reference [1]), system administrators are required to back up the system prior to making a system upgrade. In addition, it is highly recommended that administrators perform a manual backup before making any significant changes to the system. For more information on the manual backup procedure, refer to Section 9.2 on page 36.

9.1 Automated Backup

By default, an automated backup of the two PDB databases is performed on a daily basis by jobs in the root crontab. These cron jobs call a script file, `daily_bkp.sh`, that performs the backup operations. `daily_bkp.sh` triggers a `mysqldump` on the selected database, compresses the output in gzip format and stores the resulting file in a defined backup location. In PDB, these automated backups are labeled as follows:

- `daily_pdb_<yyyymmdd>.sql.gz`
- `daily_audit_<yyyymmdd>.sql.gz`

Note: If not modified, `daily_bkp.sh` stores the compressed backup files to `/var/backups/pdb/` and `/var/backups/audit/` on the local file system. For enhanced security, it is highly recommended that these backup files be moved to an external storage solution.

This automated backup procedure is defined during PDB installation and can be modified by making changes to the root crontab. For more information, refer to Section 9.1.1 on page 35.

9.1.1 Working with crontab

Making modifications to the PDB automated backup requires interaction with the root crontab.

To list all of the scheduled activities defined in the root crontab:

1. Use SSH to log in to the PDB server as root.
2. Execute the following command:

```
crontab -l
```

Output similar to the following is displayed:



```
# DO NOT EDIT THIS FILE - edit the master and reinstall.
# (/tmp/crontab.XXX9Xlh0X installed on Thu May 21 23:43:30 2009)
# (Cron version V5.0 -- $Id: crontab.c,v 1.12 2004/01/23 18:56:42 vixie Exp $)
0 0 * * * /usr/local/pdb/daily_bkp.sh pdb
5 0 * * * /usr/local/audit/daily_bkp.sh audit
0 0 * * * find /var/backups/pdb -name "daily_pdb*" -type f -mtime +30 -exec rm {} \;
5 0 * * * find /var/backups/audit -name "daily_audit*" -type f -mtime +30 -exec rm {} \;
```

3. To modify the crontab, execute the following command while logged in as root:

```
crontab -e
```

Note: The editor used to modify the crontab can vary as determined by the VISUAL or EDITOR environment variables.

By default, the root crontab on PDB contains four commands:

```
— 0 0 * * * /usr/local/pdb/daily_bkp.sh pdb
```

Performs an automated backup of the MySQL database every night at midnight by running the daily_bkp.sh script.

```
— 0 0 * * * find /var/backups/pdb -name "daily_pdb*" -type f
-mtime +30 -exec rm {} \;
```

Performs a cleanup procedure as part of the daily cron job, removing any PDB backup files that are older than one month (set by default).

```
— 5 0 * * * /usr/local/audit/daily_bkp.sh audit
```

Performs an automated backup of the audit database every night at five minutes past midnight by running the daily_bkp.sh script.

```
— 5 0 * * * find /var/backups/audit -name "daily_audit*" -type f
-mtime +30 -exec rm {} \;
```

Performs a cleanup procedure as part of the daily cron job, removing any audit backup files that are older than one month (set by default).

Note: To change the number of days that automated backup files are kept in storage, edit the mtime variable as follows:

```
-mtime +X
```

Where X is the number of days that backup files are kept.

For more information on the crontab command, refer to the UNIX man-pages.

9.2 Manual Backup

A manual backup of PDB involves making a usable copy of the pdb and audit databases and storing them on an external backup solution.



To perform a manual backup:

1. Use SSH to log in to the PDB server as a glassfish user.
2. Stop the PDB application:

```
asadmin disable <PDB application>
```

Note: Stopping the application interrupts all PDB activities.

3. Perform a mysqldump on the pdb database:

```
mysqldump -c pdb > <external storage location>/pdb_<version>.sql
```

4. Perform a mysqldump on the audit database:

```
mysqldump -c audit > <external storage location>/audit_<version>.sql
```

5. Start the PDB application:

```
asadmin enable <PDB application>
```

A manual backup of the PDB database is complete.

10 Restoring PDB

Problems with PDB can lead to situations where the normal operation of the system is impaired or completely interrupted. When these situations are the result of data loss or corruption they can be fixed by restoring a recent backup.

Note: Once a backup has been restored, all changes since the moment the backup was taken are lost.

For more information on PDB data recovery, refer to Section 10.1 on page 37.

When problems with PDB have disrupted the underlying software (GlassFish, PDB, MySQL), the system may have to be reinstalled prior to performing data recovery. For more information on PDB software recovery, refer to Section 10.2 on page 38.

10.1 PDB Data Recovery

Restoring PDB data involves recreating the databases with an existing MySQL backup. The `pdb` and `audit` databases are restored separately.



Caution!

The MySQL backup must be compatible with the version of the PDB application that is currently running. Contact PDB support before attempting to restore a backup on top of a different version of the application.

Restoring PDB is a manual procedure. To restore the PDB data:

1. Use SSH to log in to the PDB server as glassfish.

2. Stop the PDB application:

```
asadmin disable <PDB application>
```

Note: Stopping the application interrupts all PDB activities.

3. Navigate to the backup directory:

```
cd <external storage location>
```

4. Decompress the backup file if it has been archived:

```
gunzip <backup file>.gz
```

5. Recreate the affected database using a previously saved MySQL dump, as needed:

- `mysql pdb < <external storage location>/<backup file>.sql`
- `mysql audit < <external storage location>/<backup file>.sql`

6. Start the PDB application:

```
asadmin enable <PDB application>
```

PDB data recovery is complete.

10.2 PDB Software Recovery

In the event that problems with the system have disrupted the underlying software, parts of the system may need to be restaged.

Note: It is important to troubleshoot the system before reinstalling any software. Contact IMSREF System Tool support for help with troubleshooting procedures.

For complete installation procedures, refer to the Parameter Database (PDB) Installation Instructions (Reference [1]).



11

Configuring the PDB CLI

The PDB CLI is a collection of command line tools that allows you to make use of the PDB functionality without having to log into PDB server.

Several tools in the PDB Command Line Interface (CLI) act as clients of the PDB server. To function correctly, these commands require a valid PDB user account and connectivity with a PDB server.

Settings that configure the PDB CLI as a client of the PDB server are defined in the `pdcli.properties` file.

Note: The selected PDB server must be network accessible from the machine where the CLI client has been installed.

`pdcli.properties` is part of the CLI installation package and is located in the same directory. The file contains one parameter, `PDBCLIService` that specifies the WSDL URL that is used to connect to a PDB server. The URL is formatted as follows:

```
PDBCLIService=http://<hostname>:8080/PDBCLIService/CLI?wsdl
```

To change the PDB server that the CLI connects to, perform the following steps on the machine where the CLI client has been installed:

1. On the machine where the PDB CLI has been installed, navigate to the PDB CLI installation directory.
2. Open the `pdcli.properties` file for editing.
3. Modify the PDB hostname in the current WSDL URL to connect to a different server.
4. Save your changes and exit.

CLI commands will now use the updated configuration to connect to the PDB server.



Reference List

- [1] Parameter Database (PDB) Installation Instructions, 1/1531-CXP 902 0212 Uen
- [2] PDB User Guide, 2/1553-CXP 902 0212 Uen
- [3] IMSREF Centralized User Database (CUDB) System Administration Guide, 1/1543-CXP 902 0224 Uen