
HP Universal SLA Manager

Version 4.1



Administration Guide

Edition: 1.0

for Linux and Microsoft Windows Operating Systems

August 2014

© Copyright 2014 Hewlett-Packard Company, L.P.

Legal Notices

Warranty

The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

License Requirement and U.S. Government Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notices

© Copyright 2014 Hewlett-Packard Development Company, L.P.

Trademark Notices

Adobe®, Acrobat® and PostScript® are trademarks of Adobe Systems Incorporated.

HP-UX Release 10.20 and later and HP-UX Release 11.00 and later (in both 32 and 64-bit configurations) on all HP 9000 computers are Open Group UNIX 95 branded products.

Java™ is a trademark of Oracle and/or its affiliates.

Microsoft®, Internet Explorer, Windows®, Windows Server®, and Windows NT® are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Oracle® is a registered U.S. trademark of Oracle Corporation, Redwood City, California.

EnterpriseDB® software and Postgres Plus® database are registered trademarks of EnterpriseDB Corporation in the USA and in several other countries

UNIX® is a registered trademark of The Open Group.

X/Open® is a registered trademark, and the X device is a trademark of X/Open Company Ltd. in the UK and other countries.

Red Hat® is a registered trademark of the Red Hat Company.

Linux® is a registered trademark of Linus Torvalds in the U.S. and other countries.

Contents

Preface	5
Chapter 1 USLAM Architecture.....	9
1.1 USLAM Components	10
1.1.1 USLAM Services	10
1.1.2 USLAM ETL	11
1.1.3 USLAM Reports	11
1.1.4 MYUSLAM Portal.....	12
Chapter 2 USLAM Platform Configuration.....	13
2.1 Introduction	13
2.2 Configuring USLAM Server Properties	13
2.2.1 Configuring USLAM using a property file	13
2.2.2 Configuring USLAM using the JMX Console	14
2.2.3 Mandatory USLAM parameters	16
2.3 USLAM Web UI Configuration	17
2.3.1 Configuring USLAM Web UI from the Platform Administration menu	17
2.4 USLAM Reports Configuration.....	20
2.4.1 Multi-Language Support for Reports.....	20
2.5 USLAM Action Executors Configuration	25
2.5.1 SMTP Action Executor configuration	25
2.5.2 SNMP Action Executor configuration.....	25
Chapter 3 USLAM Platform Administration	26
3.1 Starting USLAM Services	26
3.2 Stopping USLAM Services.....	26
3.3 JMX Administration	26
3.3.1 Logging in to the JMX Console	27
3.3.2 Configuring JMX Console Authorization	27
3.4 Configuring USLAM UI User Authentication	28
3.4.1 Changing Authentication Mode.....	28
3.5 Managing collectors	30
3.6 Managing Action Executors	30
3.7 Starting USLAM ETL.....	32
3.8 Stopping USLAM ETL.....	32
Chapter 4 USLAM Database Configuration	33
4.1 Introduction	33
4.2 Database Instance Distribution.....	33
4.2.1 Configuring USLAM Operational Database	33
4.2.2 Configuring USLAM Datamart/Reports Database	33
4.2.3 Error “TNS: no appropriate service handler found”	34
4.3 Database Cleanup	34
4.3.1 USLAM Services retention policy	35

4.3.2	USLAM Datamart retention policy	35
4.3.3	USLAM Datamart cleanup	36
Chapter 5 USLAM Platform Distribution		38
5.1	Overview	38
5.2	Monolithic Platform	38
5.2.1	Characteristics	39
5.2.2	Deployment Instructions	39
5.3	Distributed Platform	40
5.3.1	Characteristics	41
5.3.2	Deployment Instructions	41
Chapter 6 BI Report Publisher Administration		44
6.1	Introduction	44
6.2	Starting BI Automatic Report Publisher	44
6.2.1	Automatic Report Publisher	44
6.2.2	On-demand Report Publisher	44
6.3	Configuring Report Generation Criteria	45
6.3.1	Setting a configuration parameter.....	45
6.3.2	Main Report Publisher configuration parameters	47
Chapter 7 MyUSLAM Portal Administration		48
7.1	Importing USLAM users	48
Chapter 8 USLAM Licensing		50
8.1	Overview	50
8.2	Installing or Updating a USLAM License	50
8.3	Removing a USLAM License	50
8.4	Checking the USLAM License status	51
8.5	USLAM License management in details	51
Chapter 9 USLAM Logs		52
9.1	Configuring USLAM Services Logs	52
9.1.1	Log4j Configuration File	52
9.1.2	JMX Web Console	53
9.1.3	USLAM Services Log Files	54
9.2	Viewing USLAM ETL Logs.....	54
9.3	Viewing USLAM Reports Log	56
9.3.1	BusinessObjects Enterprise.....	56
9.3.2	USLAM BI Auto Report Publisher	56
9.4	Configuring MyUSLAM Logs.....	57
9.4.1	standalone.xml Configuration File	57
9.4.2	Liferay Control Panel (Advisable approach)	57
Appendix A.....		59
1.	List of ports used by USLAM	59

Preface

This document provides information about the administration tasks related to the various stages of the USLAM lifecycle, explaining the user interface and the several administration features that are offered by the USLAM platform and its integrated components.

This Administration Guide is designed as an aid to the Administrator of the *HP Universal SLA Manager* to ease the administration of the USLAM and its various components and to effectively manage USLAM to achieve better business value.

Intended Audience

This document is aimed at the following personnel:

- HP USLAM Administrator.

Abbreviations and Acronyms

The following table describes the abbreviations and acronyms used in this document.

Abbreviation	Description
BO	SAP Business Object
BO-DS	SAP Business Object Data Services
BIAR	Business Intelligence Archive
CMS	Central Management Server
CI	Configuration Item
ID	Identifier
EDB PPAS	Enterprise DB Postgres Plus Advanced Server
ITIL	Information Technology Infrastructure Library
KPI	Key Performance Indicator
OLA	Operational Level Agreement
SLI	Service Level Indicator
SLA	Service Level Agreement
SLO	Service Level Objective
SLM	Service Level Management
SD	Service Definition
SI	Service Instance
SNMP	Simple Network Management Protocol

SM	Service Manager
TTR	Time To Repair
UPC	Underpinning Contract
USLAM	Universal Service Level Agreement Manager

Associated Documents

A list of existing HP Universal SLA Manager documents is given below for your reference:




Document Title	Relevant Product
HP USLAM Installation and Configuration Guide	Universal SLA Manager
HP USLAM User Guide	Universal SLA Manager

Typographic Conventions

This document uses the following conventions to identify special information:

Convention	Information Type/Example
[] (square brackets)	Interface components requiring user actions e.g. Buttons. Ex: Click [Finish] to complete the Import wizard.
() [round brackets]	Supplementary information Ex: Configuration Item (CI).
Bold type	Fields names, menus, window pane names Ex of menus: Admin → Service Level Management → Repository.
<i>Italic type</i>	Important information and/or concepts. Ex: <i>The output is an .XML file.</i>

Symbols Used in this Guide

Symbols	Information
	Note Draws your attention to additional information about a software function/feature.
	Important Draws your attention to important information regarding the proper usage of a software function/feature.
	Caution Draws your attention to an important warning.

Support

Please visit our HP Software Support Online Web site at www.hp.com/go/hpsoftwaresupport for contact information, and details about HP Software products, services, and support.

The Software support area of the Software Web site includes the following:

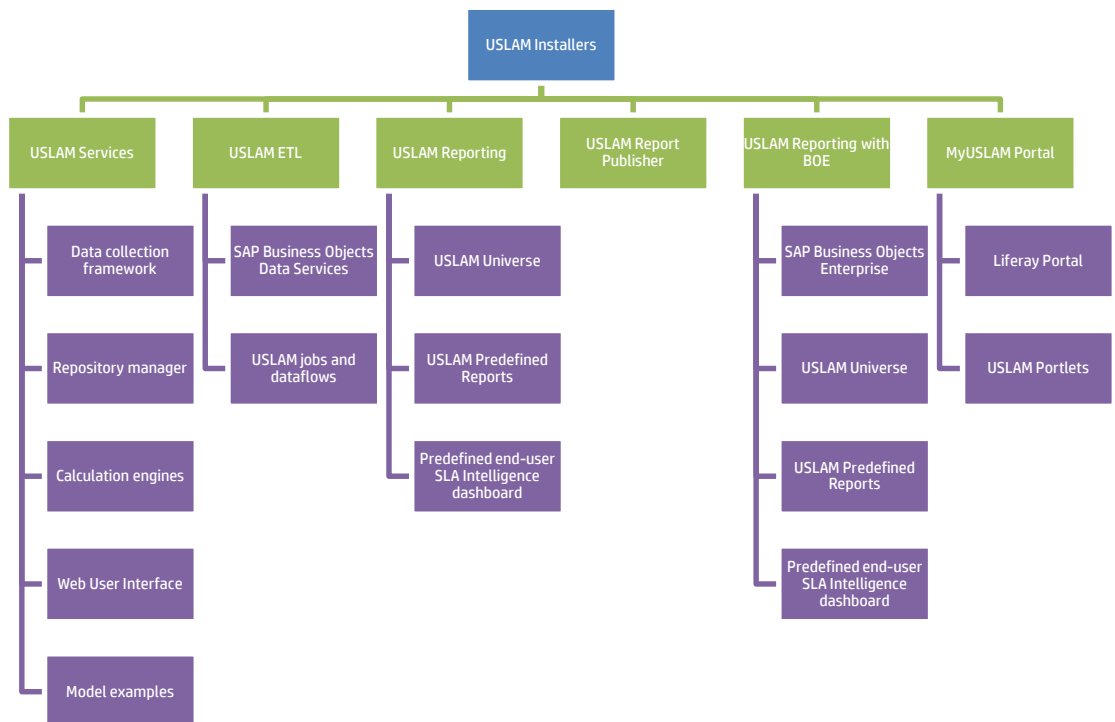
- Downloadable documentation
- Troubleshooting information
- Patches and updates
- Problem reporting
- Training information
- Support program information.

Chapter 1

USLAM Architecture

The HP Universal SLA Manager (USLAM) provides SLA management capabilities for SLA design and creation, SLA instantiations resulting from service orders, SLA near real time monitoring and alerting, and SLA reporting.

All of these tasks are achieved by the USLAM by employing several internal and external modules integrated with USLAM. The USLAM product is delivered as several software kits.



1.1 USLAM Components

The following figure drills-down further to show the various components that are integrated in USLAM, and also shows how they communicate with each other:

The main physical components of the USLAM solution are discussed briefly in the following sections.

1.1.1 USLAM Services

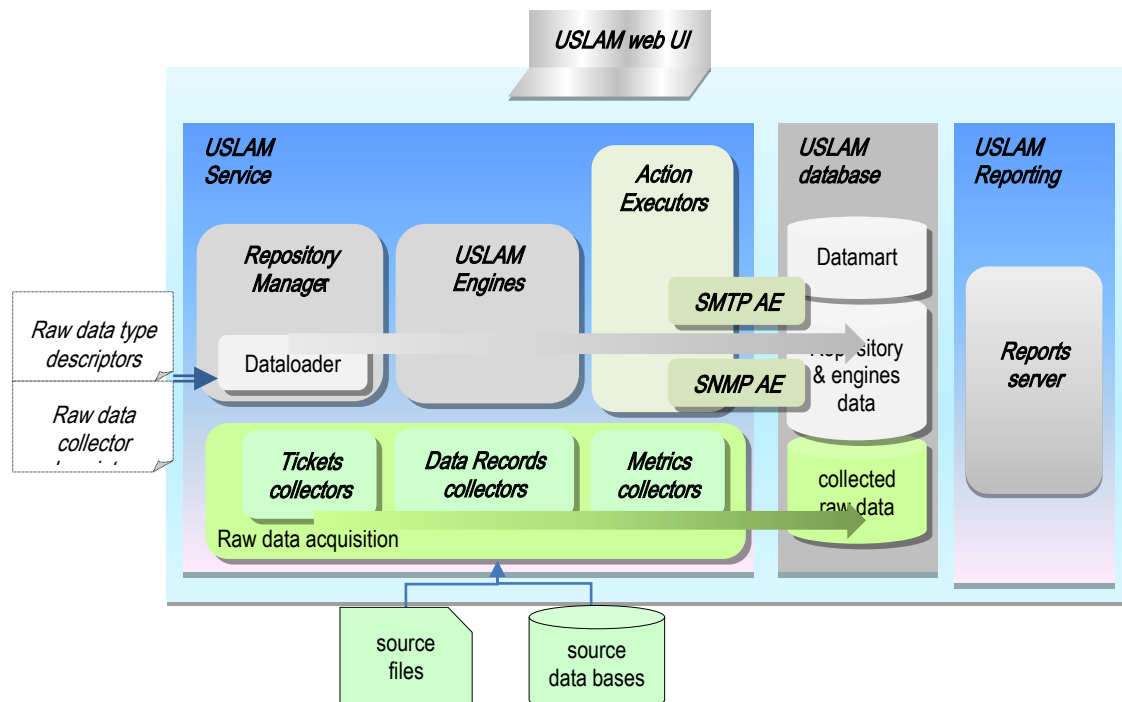
The USLAM Services package contains 5 different modules:

1. Repository Manager
2. Web UI
3. Calculation Engines
4. Collectors
5. Action executors

Repository Manager and the Web UI have to be installed on a same host.

Calculation Engines and Action executors also have to be installed on a same host.

Collectors can be installed on different host.



1.1.1.1 Repository Manager

The SLA Repository Manager is designed to handle the SLA management operations such as create, delete, update, get SLA and underlying objects and to control the SLA lifecycle (activate, terminate, etc.).

The Repository Manager also includes Dataload tools and Web User Interface. It is the starting point from which artifacts can be loaded into the system (Repository manager), managed and monitored (Web UI).

1.1.1.2 USLAM Engines

This is the heart of the solution where all compliance and business impacts calculations take place. The calculation engines are designed to process the collected data and evaluate the corresponding compliance status and business impacts.

1.1.1.3 Collectors

Raw data collectors are designed to collect from external data sources and record data to the USLAM raw data DB.

1.1.1.4 Action executors

Installed with the USLAM engines, the action executors are designed to react on alert created by these Engines and perform predefined actions.

- The SMTP Action Executor sends email on alert detection.
- The SNMP Action Executor sends trap on alert detection.

1.1.1.5 Web User Interface

Installed with the Repository Manager, it contains the USLAM Web User interface used by administrator or operators to monitor, manage and create SLA.

1.1.2 USLAM ETL

The USLAM reporting engine is designed to maintain a SLA datamart where SLA data (e.g. definition, status, compliance, etc.) is stored and summarized, ready for SLA reports building and navigation.

The SLA datamart content is maintained by an ETL (Extract Transform Load) job that can be scheduled up to several times per day. Extracted data come from the SLA registry and the SLI/SLA engine database.

1.1.3 USLAM Reports

The USLAM Reporting package offers a complete Reporting solution. There are 3 kits available:

- HP USLAM Reporting Software: this kit must be installed only if you don't have any installation of Business Object Enterprise server.
- HP USLAM Universe and Standard Reports. this kit must be installed once you have a Business Object Enterprise server installed.
- HP USLAM Reports Publisher: this kit is optional, it is a tool for the automation of report publications.

The reporting package includes a set of predefined reports and predefined reporting dashboards.

It is powered by SAP Business Objects Enterprise XI.



For more information about ETL and USLAM Reports concepts and procedures, refer to the *HP USLAM User Guide*.

1.1.4 MYUSLAM Portal

As an option, USLAM delivers a new end user community portal called 'MyUSLAM'.

This highly customizable portal embeds several USLAM portlets that can be used to build private or public business dashboards, extending business metrics visibility to business managers, end customers and partners.

Portlets bundled by MyUSLAM are:

1. SLA Status Snapshot
2. SLA Status
3. SLA Item Status
4. Clause Status
5. User Import

Chapter 2

USLAM Platform Configuration

2.1 Introduction

This chapter includes the description of the configuration properties for the various USLAM components. We will also discuss the procedures involved in performing the properties configuration using the configuration properties file and the JMX console.

This chapter also introduces the Database connection properties and their configuration details.

2.2 Configuring USLAM Server Properties

You can configure USLAM global platform properties using two different methods:

2.2.1 Configuring USLAM using a property file

This method consists in creating a property file containing a list of lines formatted as:

```
<property_key>=<property_value>
```

The list of property keys is listed in section 2.2.3 below.

Once created, properties can be loaded into USLAM using the **uslam_config.sh** tool located in the bin directory of USLAM, with the following command:

```
uslam_config.sh -file <property_file>
```

A sample property file is located in *<Install_Dir>/uslam.properties*.



This method must be used only when USLAM is stopped

2.2.2 Configuring USLAM using the JMX Console

This method consists in using the JMX Console to set up USLAM properties.

The JMX (Java Management eXtension) is used by USLAM to provide web-based, modular and dynamic solution for managing and monitoring different USLAM components.

The JBoss Management Console lists all registered services that are active in the application server and that can be accessed through the JMX console. Using the JMX console, you can view wide-ranging information about the active components, allow various configurations and management operations.

There are three types of services offered by USLAM through the JMX console:

- Displaying Health Indicators (uslam.indicators); dedicated for health monitoring of the platform
- Managing the USLAM platform (uslam.management); provides easy access to administer the USLAM components
- Configuring parameters of different USLAM component properties and invoking relevant operations (uslam.parameters).

The following sections include information about configuring the parameters for different USLAM component properties and the relevant operations that are allowed by the JMX console.

For more information about health indicators, refer to *Chapter 5 Platform Health Indicators*.

For more information about working with the USLAM administration options offered by the JMX console, refer to *Chapter 3 Platform Administration*.

For more information about how to access and login to the JMX console, refer to *Chapter 3 Platform Administration*.

Once you log in to the JMX console, the JMX Agent View window displays. An example is shown in the following figure:



All available objects are listed in the pane on the left of the window. You can configure the parameters of the USLAM component properties using objects which name starts with “uslam.parameters”.



This method must be used only when USLAM is started.

2.2.3 Mandatory USLAM parameters

The following parameters must be configured before using the platform. They can be configured either through a property file as described in section 2.2.1 above or using the JMX console as described in section 2.2.2 above.

There are numbers of other parameters for which a default value is provided. You should not change them unless explicitly required by USLAM support.

JMX domain	JMX attribute	Property key	Description	Default value
<i>via JMX</i>		<i>via property file</i>		
uslam.parameters.slaengine	DefaultTimeZone	default_timezone	The default calculation timezone	--
uslam.parameters.repo	DefaultAgreementDatesTimezone	repo.load.slaTimeZone	Default agreement date time zone. Used when the time zone is not specified when setting SLA start dates and end dates in XML files.	GMT

2.3 USLAM Web UI Configuration

Several parameters must be configured before using the USLAM Web User Interface. They can be configured either through a property file as described in section 2.2.1 above or from the UI itself as described in section 2.3.1 below.

The list of mandatory Web UI parameters is:

Property	Property key (for property file)	Description
UslamServersList	webui.uslamservers	<i>Comma separated list of host names on which USLAM Servers are running. RMI port name can be specified if different from the default (1099). Ex: host1.mycorp.com,host2.mycorp.com:1199</i>
DefaultDisplayTimezone	common.display.timezone	<i>The default display timezone</i>
BoServerHostName	webui.report.server.host	<i>Host name of the Business Objects server</i>
BoServerPort	webui.report.server.port	<i>Listening port of the Business Objects server</i>
BoDatePatternFormat	webui.report.date.pattern	<i>The date format pattern used for Business Objects reports</i>

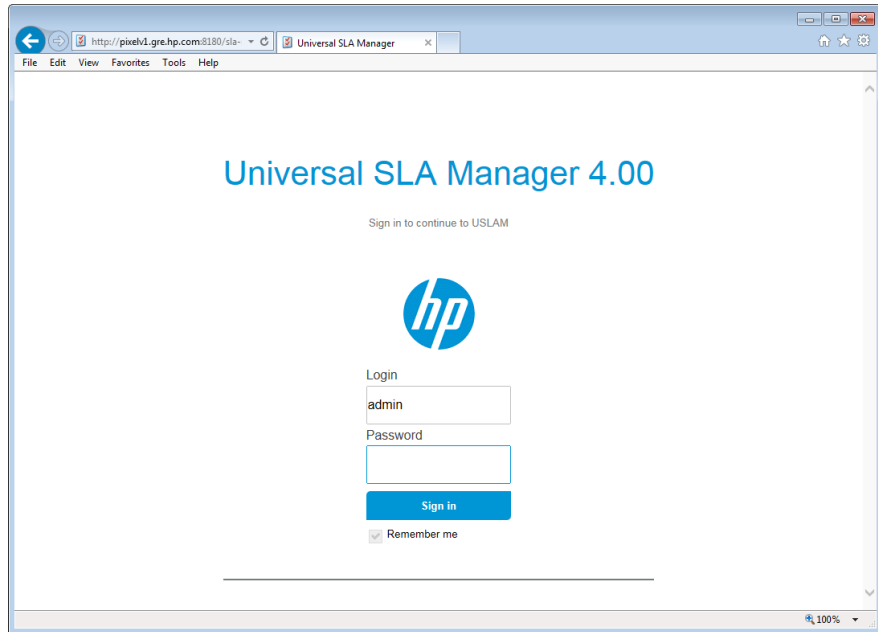
2.3.1 Configuring USLAM Web UI from the Platform Administration menu

The Web UI properties can be set directly from the UI itself. While configuring the USLAM platform, the properties listed above should

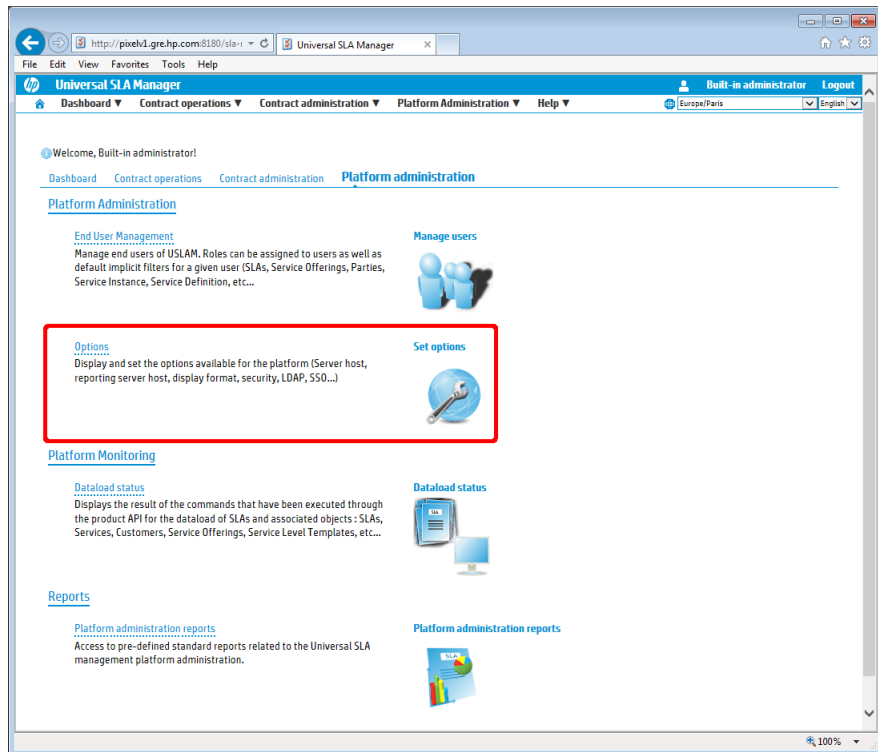
1. Open your web browser and enter the following URL to access the USLAM user interface:

```
http://<server address>:8080/sla-repository
```

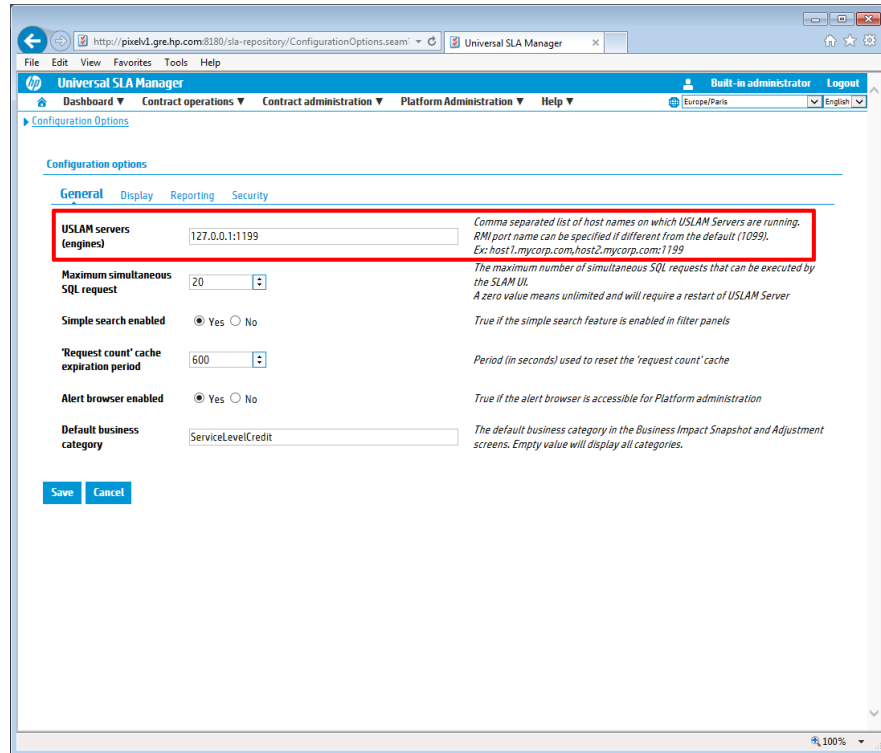
6. Logon to the USLAM UI. Enter the user access credentials of the administrator account.



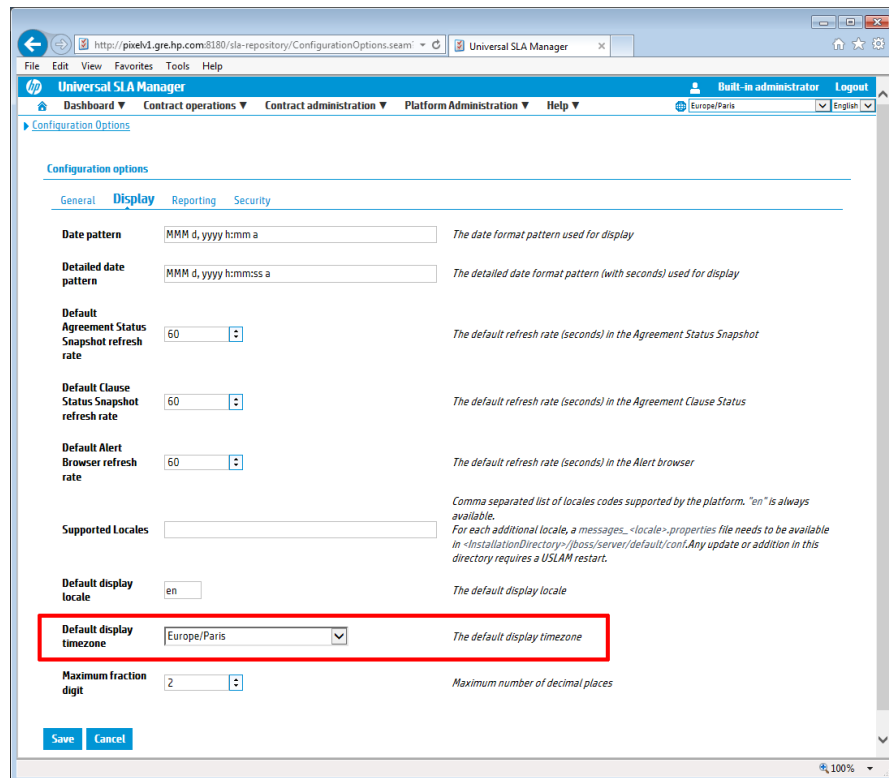
7. Go to the Platform administration → Set options screen



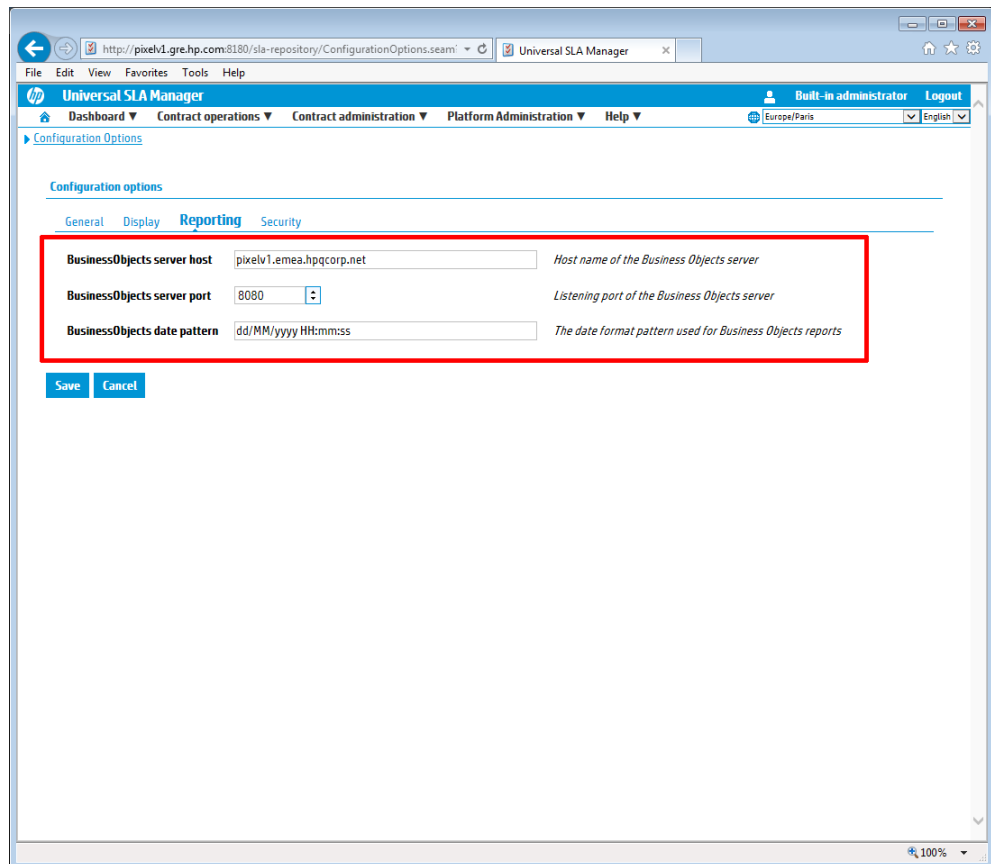
8. In General tab, set USLAM servers



9. In Display tab, set Default display timezone



10. In Reporting tab, set BusinessObjects server host, BusinessObjects server port and BusinessObjects date pattern

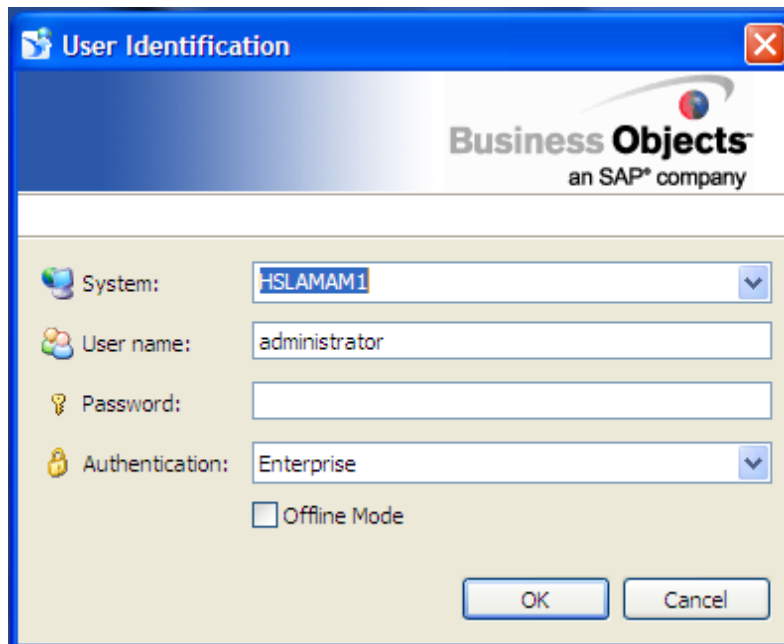


2.4 USLAM Reports Configuration

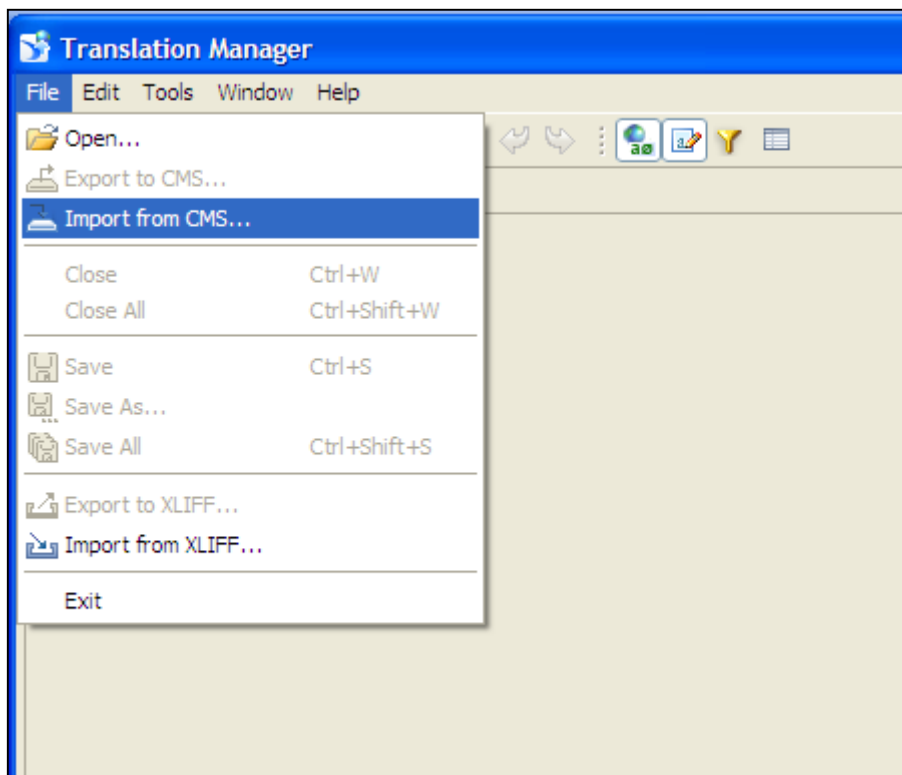
2.4.1 Multi-Language Support for Reports

USLAM reports can be easily translated to different languages. This can be done by using the Business Objects application **Translation Manager**. You can translate Universe objects (on which the reports are based on). In the **Translation Manager** universe, you will be required to add the new language and input the value of the objects associated with it. To perform these tasks, follow the steps given below.

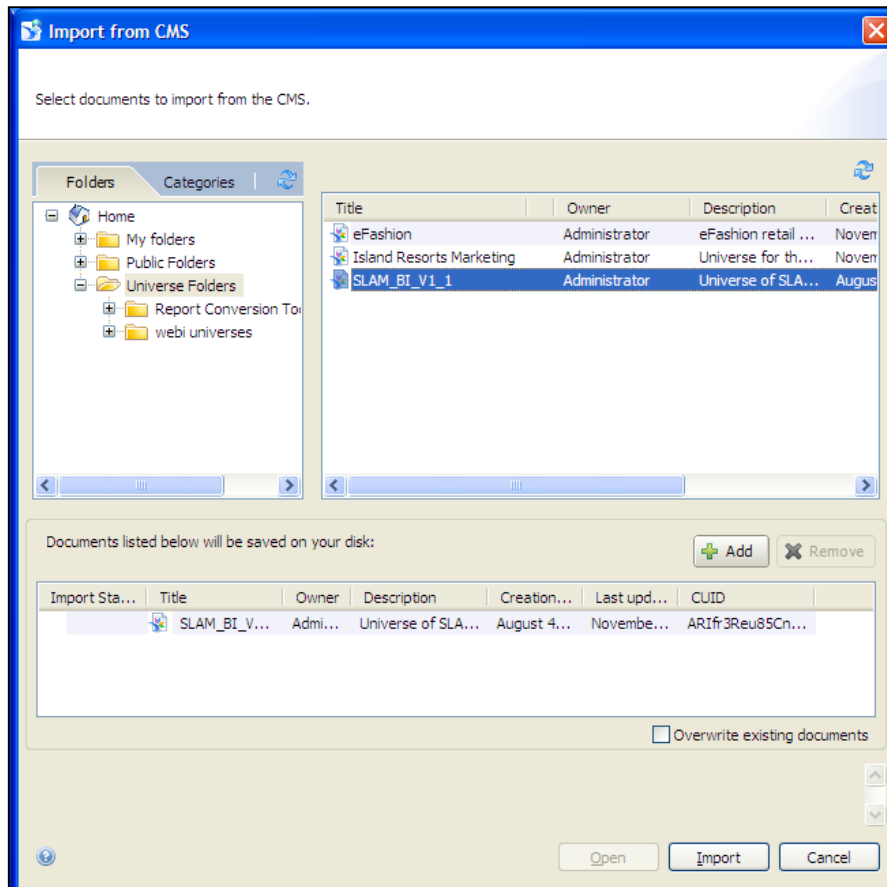
1. Open the **Translation Manager**, from Start Menu → All Programs → BusinessObjects XI 3.1 → BusinessObjects Enterprise → Translation Manager
2. Logon to the BOE CMS. Enter the correct server name, user name and password



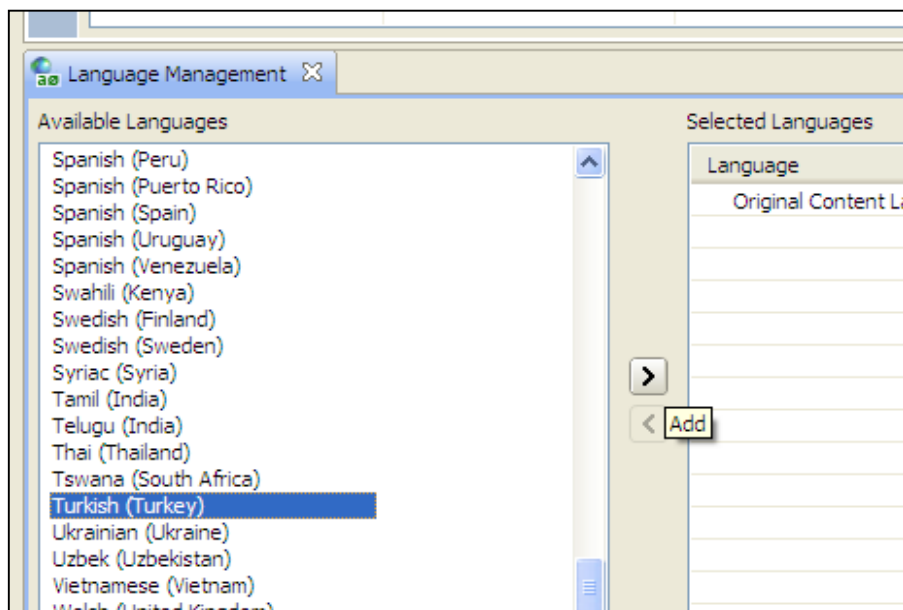
3. Import the universe from CMS, **File** → **Import from CMS**



4. The Import from CMS dialog displays



5. Select the relevant universe (i.e. *USLAM_Universe_v4*), and then click [Add]
6. Click [Import]
7. Now you can add the new language to the Translator using the Language Management panel by selecting the language and clicking [Add], as shown in the figure below



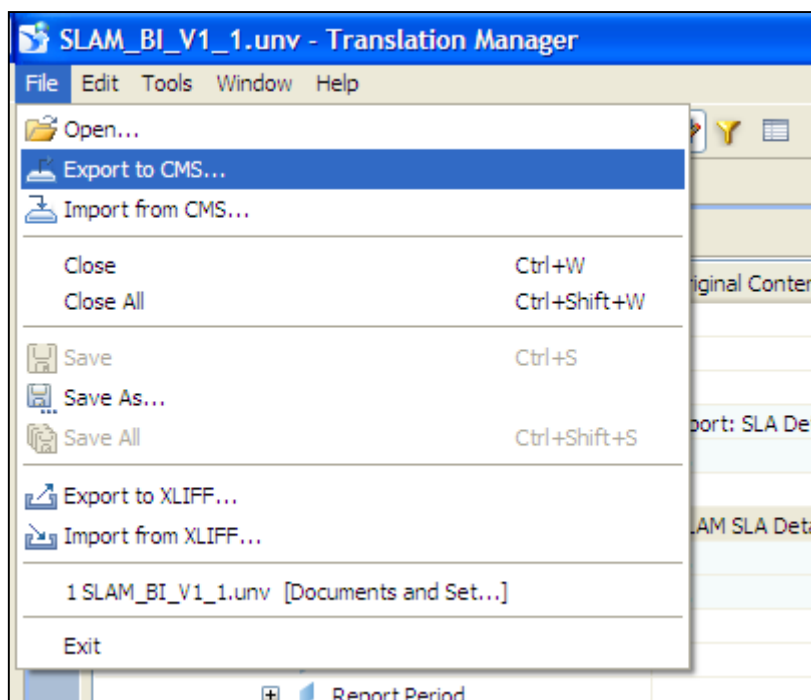
8. Make sure the [ready to use] option is checked for the new language

Language	View	Au...	Tr...	Ready for use
Original Content Language	<input checked="" type="checkbox"/>	<input type="radio"/>	N/A	<input type="checkbox"/>
Turkish (Turkey)	<input checked="" type="checkbox"/>	<input type="radio"/>	56...	<input checked="" type="checkbox"/>

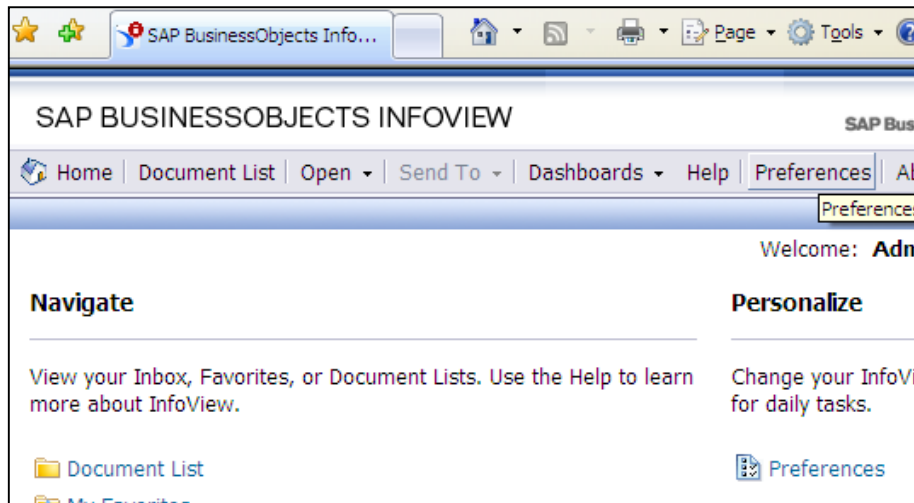
9. In the main window of the universe, the new column appears for the newly added language. Enter the value for the objects of the language

	Original Content Language	Turkish (Turkey)
Report: Customer All SLAs		
Report: Service All SLAs		
Report: SLA Details		
Name	Report: SLA Details	SLA Detayı
Description
uSLAM SLA Details for a Refer		
Name	uSLAM SLA Details for a Refer...	uSLAM Referans Dönemi için SLA Detayı
Description
Format
SLA Properties		
SLA Recurrence		

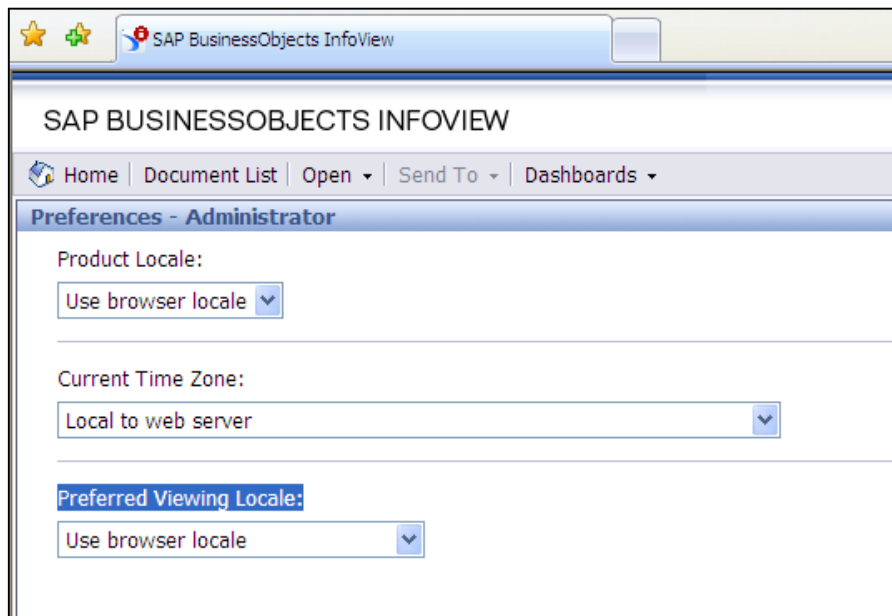
10. Export the universe to CMS, to finish the configuration of the universe



11. Now you will be required to configure the Infoview of the reports. Login to the BusinessObjects Infoview and then click [Preferences]



12. In the [General] section, set the [Preferred Viewing Locale] to [Use browser locale] option



13. This setting will enable the reports to be displayed in the language that is selected as default in the browser (i.e. only if this language was translated in universe translation manager)
14. Click [OK] to save the setup of the Infoview

For more information about the BO Translation Manager, refer to the *Getting started with Translation Manager* section of *Translation Manager* document (http://help.sap.com/businessobject/product_guides/boexir3/en/xi3_translation_manager_en.pdf),

For more information about the BO Infoview locale settings, refer to *Accessing Web Intelligence* chapter of the *Building reports using the Java Report Panel* document (http://help.sap.com/businessobject/product_guides/boexir31/en/xi3-1_web_intelligence_java_report_panel_en.pdf)

2.5 USLAM Action Executors Configuration

In order to be able to run an action executor, USLAM requires that the corresponding protocol is configured properly. The configuration depends on the action executor type, as described below

2.5.1 SMTP Action Executor configuration

The SMTP AE requires that the mail service is configured to identify the mail server to be used.

This configuration is done thanks to a property file located at `<Install_Dir>/jboss/server/default/deploy/mail-service.xml`

In order for the SMTP action executor to be able to send email you should edit this file and set as a minimum the following properties:

Property	Property key
SMTP gateway server	mail.smtp.host
Mail server port	mail.smtp.port



This method must be used only when USLAM is stopped

2.5.2 SNMP Action Executor configuration

The SNMP agent used via the SNMP Action Executor can be configured thanks to the `uslam.properties`. Please refer to section *2.2.1 Configuring USLAM using a property file* for details on property file settings.

Especially, the port used by the SNMP agent can be modified thanks to the property named `uslam.actionexecutor.snmp.port`.

USLAM Platform Administration

This chapter contains information about the general administrative operations that can be performed to manage the USLAM platform. It also provides details about the JMX console administration and its relevant operations.

3.1 Starting USLAM Services

Once you have installed and configured the USLAM Services you can start these services by performing the following steps:

1. After the installation and configuration of USLAM Services, go to `<Install_Dir>/bin` and enter **uslam_start.sh** to start USLAM services
2. It can takes few minutes to be completely started. You can check if the USLAM services are running by executing the following command:
`<Install_Dir>/jboss/bin/twiddle.sh get "jboss.system:type=Server" Started`
3. If the response from this command is: Started=true, then the USLAM services are running.

For more information about starting the USLAM services and to verify that the USLAM service have started successfully, refer to *Chapter 2 Installing and Configuring USLAM Services* in the *HP USLAM Installation and Configuration Guide*.

3.2 Stopping USLAM Services

To stop USLAM Services you will be required to perform the following steps:

1. Browse to the directory where USLAM Services are installed, and browse to:
`<Install_Dir>/bin`
2. Enter **uslam_stop.sh** with the correct parameters to stop USLAM services (uslam_stop -h to get the complete usage).

For more information about stopping the USLAM services and to verify that the USLAM service have stopped successfully, refer to *Chapter 2 Installing and Configuring USLAM Services* in the *HP USLAM Installation and Configuration Guide*.

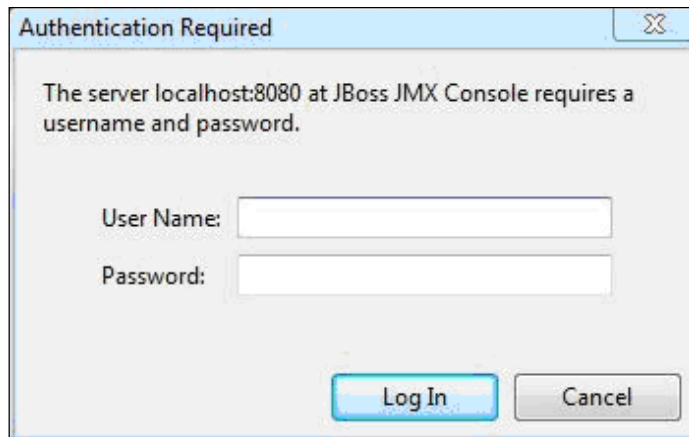
3.3 JMX Administration

This section provides information about the procedures related to the administration of JMX console. It also contains information about how to work with the JMX Console to perform several administration and management tasks for the USLAM User Interface and other USLAM components.

3.3.1 Logging in to the JMX Console

To log in to the JMX console you must have appropriate credential. You can log in to the JMX console by performing the following steps:

1. Open your web browser and enter the following URL:
`http://JbossServerHost:8080/jmx-console/`
2. Press [Enter]. The Authorization Required dialog displays



3. Enter your login credentials and then click [Log In] to log in to the JMX console. The default username/password is **admin/uslim**.

3.3.2 Configuring JMX Console Authorization

The user authorization information (user name, password) can be configured in the file **jmx-console-users.properties**, which is located under the following path:

`<Install_Dir>/jboss/server/default/deploy/conf/props/`

3.4 Configuring USLAM UI User Authentication

USLAM Web UI provides two user authentication modes:

Built-in: In this mode, you have only *Administrator* predefined user, which is granted with the *USLAM Administrator* role.

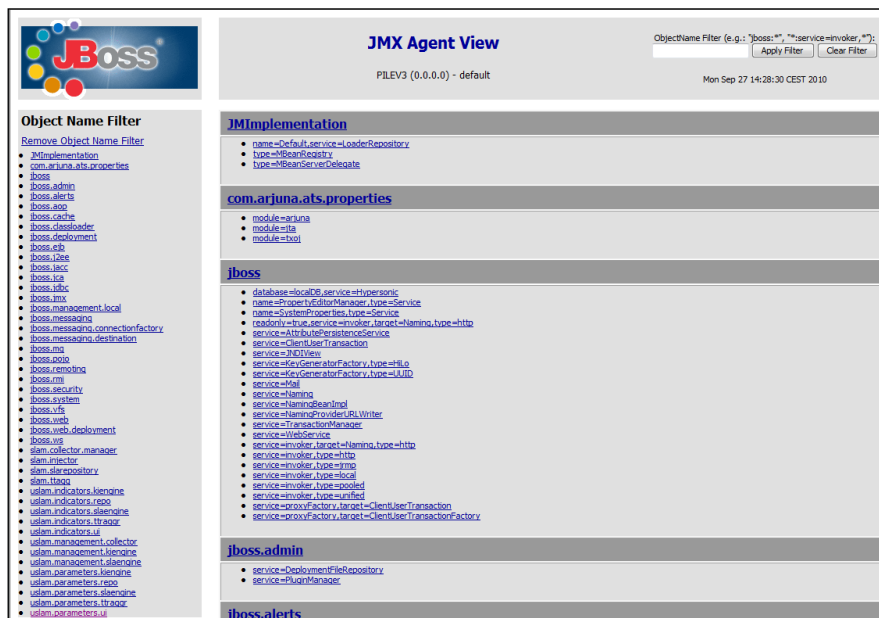
This mode should only be used for testing or demonstration purposes. This is the default mode.

LDAP: Use this mode to connect USLAM Web UI to an external LDAP system. This mode requires additional configuration of the USLAM Web UI as described below.

3.4.1 Changing Authentication Mode

To change the user authentication mode, you will be required to perform the following steps:

1. Once you log in to the JMX console, the JMX Agent View window displays. An example is shown in the following figure:



The screenshot shows the JMX Agent View window for the jboss:*. The left pane displays the Object Name Filter tree, which is expanded to show the path `uslam.parameters.ui`. The right pane displays the JImplementation section, which shows the properties of the selected object. The properties are:

- `name=DefaultServiceLoaderRepository`
- `type=MBeanRegistry`
- `type=MBeanServiceDelegate`

The `com.arjuna.ats.properties` section shows the following properties:

- `module=arjuna`
- `module=ita`
- `module=txos`

The `jboss` section shows the following properties:

- `database=localDB.service=HyperSpace`
- `name=ProjectCollectorManager.type=Service`
- `name=ClusterProperties.type=Service`
- `readOnly=true.service.provider.target=lamina.type=http`
- `service=http://localhost:8080/Service`
- `service=ClientUserTransaction`
- `service=JNDIView`
- `service=KeyGeneratorFactory.type=UUID`
- `service=Mail`
- `service=Name`
- `service=NameBeanImpl`
- `service=NameProviderWriter`
- `service=TransactionManager`
- `service=WebService`
- `service=invoker.target=lamina.type=http`
- `service=invoker.type=http`
- `service=invoker.type=jms`
- `service=invoker.type=sjml`
- `service=invoker.type=spooled`
- `service=invoker.type=unified`
- `service=poolFactory.target=ClientUserTransaction`
- `service=poolFactory.target=ClientUserTransactionFactory`

The `jboss.admin` section shows the following properties:

- `service=DeploymentFileRepository`
- `service=PluginManager`

The `jboss.alerts` section is currently empty.

2. From the left pane in the JMX Agent View window, click `[uslam.parameters.ui]` and then `[service=ConfigParamManager]`. The UI properties display

SecurityAuthenticationMode	RW	java.lang.String	The security authentication mode (either 'Built-in' or 'LDAP')	LDAP
SecurityLDAPServerAddress	RW	java.lang.String	LDAP Server Address	ldap.hp.com
SecurityLDAPServerPort	RW	java.lang.Integer	Listening port of the LDAP server	636
SecurityLDAPBindDN	RW	java.lang.String	LDAP Bind DN (needed if anonymous bind is disallowed)	
SecurityLDAPBindCredentials	RW	java.lang.String	LDAP Bind Credentials (needed if anonymous bind is disallowed)	
SecurityLDAPUserNameAttribute	RW	java.lang.String	LDAP User Name Attribute	uid
SecurityLDAPLoginAttribute	RW	java.lang.String	LDAP user attribute used for the login	uid
SecurityLDAPDisplayNameAttribute	RW	java.lang.String	LDAP user attribute used for the display of the user name	cn
SecurityLDAPBaseDN	RW	java.lang.String	The Base DN for users lookup	o=hp.com
SecurityLDAPUseSSL	RW	boolean	True if LDAP provider uses secure connection	<input checked="" type="radio"/> True <input type="radio"/> False
SecurityLDAPTrustStorePath	RW	java.lang.String	Path to the TrustStore file, if required by the LDAP provider (backslashes must be doubled)	/opt/USLAM/ATT47080.keystore

3. Locate the parameter named **SecurityAuthenticationMode** and change its value to either **Built-in** or **LDAP**

SecurityAuthenticationMode	RW	java.lang.String	The security authentication mode (either 'Built-in' or 'LDAP')	LDAP
-----------------------------------	----	------------------	--	------

3.4.1.1 Configuring LDAP Parameters

When the **SecurityAuthenticationMode** property is set to **LDAP**, you must define a number of parameters in order to describe:

- How to reach the LDAP server
- How your user are stored in your LDAP server
- These parameters, which can be set through the JMX Console, are described below:

Name	Description	Example
SecurityLDAPServerAddress	The IP address of your LDAP Server	ldapmycorp.com
SecurityLDAPServerPort	The listening port of your LDAP Server	389
SecurityLDAPBindDN	LDAP Bind DN (needed if anonymous bind is disallowed)	cn=Manager,dc=slam,dc=com
SecurityLDAPBindCredentials	LDAP Bind Credentials (needed if anonymous bind is disallowed)	secret
SecurityLDAPUserNameAttribute	LDAP User Name Attribute	uid
SecurityLDAPLoginAttribute	LDAP user attribute used for the login	uid
SecurityLDAPDisplayNameAttribute	LDAP user attribute used for the display of the user name	cn
SecurityLDAPBaseDN	The Base DN for users lookup	o=mycorp.com

Name	Description	Example
SecurityLDAPUseSSL	True if LDAP provider uses secure connection	False
SecurityLDAPTrustStorePath	Path to the TrustStore file, if required by the LDAP provider	/opt/USLAM/mycorp.keystore

3.5 Managing collectors

This section provides information about the lifecycle manager for the Data collector framework.

To access the Data Collector Framework JMX Manager, login to the JMX console (refer to *Section 3.3 JMX Administration above*).

Click [uslam.management.collectorManager] to view the list of the relevant JMX services groups in the JMX Agent View window and click [service=CollectorLifecycleManager] to view the lifecycle manager for the CommonCollectorSDK as below.

The screenshot displays the JMX MBean View for the `uslam.management.collectorManager` MBean. The main view shows the MBean's details, including its name, domain, Java class, and description. Below this, there are two tables: one for attributes and one for operations.

Attribute Name	Access	Type	Description	Attribute Value
Status	R	java.lang.String	Show all of the started collectors' statuses	The statuses of the collector instances managed by EventCollectorManager are: AbandonedSDcallsEventCollector is running: true

Operation	Return Type	Description	Parameters
register	java.lang.Boolean	Register the new collector instance into the collector manager. The collector instance is automatically started after successfully registered.	collectorInstanceName java.lang.String (no description) <input type="text"/> [Invoke]
startAllCollectors	void	Start all of the collector instances.	[no parameters] <input type="text"/> [Invoke]
start	void	Start the specific collector instance.	collectorInstanceName java.lang.String (no description) <input type="text"/> [Invoke]
startByCollectorManager	void	Start all of the collector instances managed by the specified manager.	collectorManagerName java.lang.String (no description) <input type="text"/> [Invoke]
stop	void	Stop the specific collector instance.	collectorInstanceName java.lang.String (no description) <input type="text"/> [Invoke]
stopAllCollectors	void	Stop all of the collector instances.	[no parameters] <input type="text"/> [Invoke]
stopByCollectorManager	void	Stop all of the collector instances managed by the specified manager name.	collectorManagerName java.lang.String (no description) <input type="text"/> [Invoke]

The **Status** shows all of the configured collectors' statuses and you can use the **stop/stopAllCollectors** method to stop a specified collector/all of the started collectors, the **start/startAllCollectors** method can be used to start a specified collector/all of the registered collectors.

For more information about the methods see the **Description** column.

3.6 Managing Action Executors

This section provides information about the lifecycle manager for the Action Executors framework.

To access the Action Executors Framework JMX Manager, login to the JMX console (refer to *Section 3.3 JMX Administration above*).

Click [uslam.management.actionexecutor] to view the list of the relevant JMX services groups in the JMX Agent View window and click [service=LifecycleManagement] to view the lifecycle manager for the Action Executor Manager as below.

JMX MBean View

SANNA11 Wed Nov 28 15:20:55 GMT 2012

Name	Domain	Value
	uslam.management.actionexecutor	
service		LifecycleManagement
Java Class		org.jboss.mx.model.mbean.XMBean
Description	ActionExecutorManager is used to manage lifecycle of ActionExecutors, start(), stop(), stopAll(), listAll() etc.	

Attribute Name	Access	Type	Description	Attribute Value
Operation				
start		java.lang.String	Start the specified ActionExecutor. If retrieveBacklog is set, retrieve from alert last processed, else retrieve from latest alert	actionExecutorName java.lang.String (no description) <input type="text"/> retrieveBackdog boolean (no description) <input checked="" type="radio"/> True <input type="radio"/> False <input type="button" value="Invoke"/>
start		java.lang.String	Start the specified ActionExecutor with default non-backdog mode.	actionExecutorName java.lang.String (no description) <input type="text"/> <input type="button" value="Invoke"/>
stop		java.lang.String	Stop the specified action executor instance.	actionExecutorName java.lang.String (no description) <input type="text"/> <input type="button" value="Invoke"/>
listAll		java.lang.String	List all of the currently running action executors	[no parameters] <input type="button" value="Invoke"/>
stopAll		java.lang.String	Stop all of the currently running action executors	[no parameters] <input type="button" value="Invoke"/>
suspend		java.lang.String	Suspend a specified action executor.	actionExecutorName java.lang.String (no description) <input type="text"/> <input type="button" value="Invoke"/>
resume		java.lang.String	Resume a suspended action executor	actionExecutorName java.lang.String (no description) <input type="text"/> <input type="button" value="Invoke"/>

You can use the **stop/stopAll** methods to stop a specified action executor/all of the started action executors, the **start** method can be used to start a specified action executor.

For more information about the methods see the **Description** column.

3.7 Starting USLAM ETL

When installing the USLAM ETL, the ETL servers are automatically started. But, in case you need to start them manually, here is the procedure:

As *hpuslametl* user, run the ETL servers start script:

```
$ cd <USLAM ETL InstallDir>/bin/  
$ ./uslam_etl_start.sh
```

This command will start the IPS, BODS, and Tomcat ETL servers.

Note: this command is related to ETL servers only, it does not start the ETL jobs themselves. For this, please refer to the *HP USLAM Installation and configuration guide - Executing ETL Jobs*.

3.8 Stopping USLAM ETL

As *hpuslametl* user, run the ETL servers stop script:

```
$ cd <USLAM ETL InstallDir>/bin/  
$ ./uslam_etl_stop.sh
```

This command will stop the IPS, BODS, and Tomcat ETL servers.

Chapter 4

USLAM Database Configuration

This chapter deals with information about how to configure the USLAM database, and the different requirements that must be met before the configuration.

This chapter also contains information about configuring USLAM Datamart/Reporting DB and details its pre-requisites.

4.1 Introduction

After executing the configuration tool, the system generates one XML file describing the Data source configuration as defined during configuration. This file is saved in:

```
<Install_Dir>/jboss/server/default/deploy/uslam-ds.xml
```

4.2 Database Instance Distribution

At most, there can be two database instances that may be created for USLAM. These are listed below:

- Database instance for USLAM services, for example: **uslam**
- Database instance for Datamart, for example: **uslamDM**

4.2.1 Configuring USLAM Operational Database

There are some important database creation parameters that must be considered before configuring the USLAM operational database. These parameters are:

- Database character set: AL32UTF8
- Type of database: OLTP

You should also consider the Database memory/disk space available on your system before configuring the USLAM database.

To configure the USLAM operational database, you will be required to perform step-by-step procedures. For a complete explanation for the procedures involved and information about the schemas supported, refer to *Section 2.2 Configuring USLAM* in the *HP USLAM Installation and Configuration Guide*.

4.2.2 Configuring USLAM Datamart/Reports Database

There are some important database creation parameters that must be considered before configuring the USLAM Datamart/Reports database. These parameters are:

- Database character set: AL32UTF8

- Option ‘Oracle Partitioning’ is installed (in case you are using Oracle database server)

You should also consider the Database memory/disk space available on your system before configuring the USLAM Datamart database.

To configure the USLAM Datamart database, you will be required to perform step-by-step procedures. For a complete explanation for the procedures involved, refer to *Chapter Installing and Configuring USLAM ETL* and *Chapter Installing and Configuring BO Report* from the *HP USLAM Installation and Configuration Guide*.

The following notes refer to the details about the procedures involved in creating and configuring the schemas supported by the USLAM Datamart database:

- USLAM ETL users *BODS_REPOS* and *IPS*: For more information about creating those schemas, refer to Section *Create USLAM ETL internal database users* in the *HP USLAM Installation and Configuration Guide*
- USLAM Datamart user *SLA_DATAMART*: For more information about creating this schema, refer to *Section Creating Datamart User* in the *HP USLAM Installation and Configuration Guide*
- Load the required Datamart scripts: For more information about executing the required scripts, refer to *Section Executing the Datamart Scripts* in the *HP USLAM Installation and Configuration Guide*
- USLAM CMS database user: For more information about creating this schema, refer to *Section Creating CMS Database User* in the *HP USLAM Installation and Configuration Guide*.

4.2.3 Error “TNS: no appropriate service handler found”

When executing ETL jobs on Oracle database, you may encounter those errors “TNS: no appropriate service handler found” in the *Error* jobs information of the BODS console.

This error means that the number of concurrent connections to the USLAM Service database is too high compared to the current settings of your Oracle server. The oracle server parameter values must be increased.

This can be done by asking your oracle administrator to run this query as *system* user:

```
SQL> SELECT * FROM v$resource_limit WHERE resource_name in ('processes','sessions');
```

If the *MAX UTILIZATION* is close to or equal to *LIMIT VALUE*, the oracle administrator must increase the corresponding oracle parameter value (processes or sessions).

For example, for processes:

```
SQL> ALTER SYSTEM SET PROCESSES=300 SCOPE=SPFILE;
```

and then you must restart the Oracle instance.

This will fix the “TNS: no appropriate service handler found” ETL errors.

4.3 Database Cleanup

USLAM provides an automated cleanup process for USLAM services database. USLAM provides some maintenance tools for the USLAM datamart database.

Note that a different retention policy is applied for data in USLAM services database and for data in USLAM datamart database because the former is the operational database whereas the latter is the historical database.

4.3.1 USLAM Services retention policy

The USLAM Services database cleanup process is part of the Repository component. The cleanup procedure is executed automatically and periodically, and is driven by several configuration parameters as described below.

JMX domain	JMX attribute	Property key	Description	Default value
<i>via JMX</i>		<i>via property file</i>		
uslam.parameters.repo	SupportedCalculationTimeScopeMonths	supported_calculation_time_scope_months	Number of months for data retention in DB. Computed data older than this parameter will be automatically removed from the operational database (note that this does not concern data in USLAM datamart)	4 months
uslam.parameters.repo	DBCleanupScheduledtime	repo.task.cleanup.scheduledTime	The scheduled time (in hours) for running the DB cleanup processing.	24h

4.3.2 USLAM Datamart retention policy

Administrate USLAM Datamart retention with Oracle partitioning

In case you have setup a Datamart with Oracle partitioning, the partitioning mechanism is used for the Datamart most fine grained details tables.

The partitioning behavior is defined in the Datamart schema (and can not be modified once the Datamart schema has been loaded).

This concerns the downtimes, tickets, performance metrics, data records, which are stored on different Oracle partitions (on a quarterly basis).

(This does not concern statuses, delivered values, business impact calculation)

These partitions are **'rolled' over 3 years**.

This means that for those tables:

- there are 14 partitions corresponding to 14 quarters (~ 3 years)
- depending on the date of the data, data is spread over those 14 partitions
- these partitions are 'rolled' over 14 quarters

If no Oracle administrative task is done, the data will be spread over partitions even after 3 years: because of the 'rolling' mechanism, after 3 years the new data will be stored in the same partition than the oldest data (without removing it) and so on ...

So, by default, there is no retention policy: data is kept forever.

Alternatively, you could administrate the partitions in this way:

when all the partitions are filled, an Oracle administrative task can regularly exchange partition in order to transfer oldest data partition in another place (purging or archiving place).

In this way, the downtimes, tickets, performance metrics and data records will have a 3 years retention period in USLAM Datamart.

Administrate USLAM Datamart retention without partitioning

In case you have setup a Datamart without Oracle partitioning, or with Enterprise DB Postgres Plus Advanced Server, USLAM provides some scripts in order to perform purge of data. This concerns internal tables and all facts tables (downtimes, tickets, performance metrics, data records, business impacts, and statuses). So, use this script with caution, because **old data will be removed**.

A function definition script needs to be loaded first (and only once) on Datamart. Then, you can run purge operation.



When launching the purge operation, no ETL must be running.

You can run the purge operation on a given number of months.



Purge operation on Oracle (without partitioning) requires Datamart user to have the following privileges:

```
GRANT CREATE TABLE TO <Datamart User>;
GRANT DROP ANY TABLE TO <Datamart User>;
GRANT ALTER ANY TABLE TO <Datamart User>;
```

For example, if you want to purge your Oracle datamart (without partitioning), by keeping only the last 12 months of data, perform those operations:

```
hpuslametl> cd <ETL_InstallDir>/USLAM_datamart_scripts/.DDLwithoutPartitioning
hpuslametl> sqlplus <Datamart_user>/<Datamart_passwd>@<Datamart_tns>
SQLPLUS> uslam_datamart_oracleWOPartitioning_cleanup_procs.sql
SQLPLUS> exec PROC_CLEANUP_DATAMART_ALL(12);
SQLPLUS> exit;
```

Here is the same example if your Datamart is based on EDB PPAS:

```
hpuslametl> cd <ETL_InstallDir>/USLAM_datamart_scripts/.DDLwithoutPartitioning
hpuslametl> edbplus <Datamart_user>/<Datamart_passwd>
SQLPLUS> uslam_datamart_ppas_cleanup_procs.sql
SQLPLUS> exec PROC_CLEANUP_DATAMART_ALL(12);
SQLPLUS> exit;
```

4.3.3 USLAM Datamart cleanup

By default, there is no USLAM Datamart cleanup processing.

But, it could happen, depending on the model you loaded and the volume of data you regularly inject in USLAM, that some internal datamart tables grow regularly with some temporary data. This concerns the tables

STG_SLSC_INTERMEDIATE, STG_SL_STATUS_CHANGE, STG_INTERMEDIATE_SLSC ASSO,
STG_ER_DATE_DATA, STG_ER_NUMBER_DATA, STG_ER_STRING_DATA, STG_TKT_RAW_DATA,
STG_MKPI_PER_ACP, STG_KPI_VALUE_FOR_KQI, STG_TICKET_TTR_VALUE,
STG_STATIC_KQI_VALUE, FCT_SLA_SLI_EVENT.

For doing some cleanup, by removing all the outdated internal data, *hpuslametl* user could execute a SQL cleanup script on Datamart database. The goal is to decrease the Datamart database size and possibly increase performance.

The scripts are located in `<ETL_InstallDir>/USLAM_datamart_scripts/`

If your Datamart relies on Oracle, you can run this SQL script on Datamart DB:

*Regular**Oracle**DatamartMaintenance_CleanupTmpData.sql*

If your Datamart relies on PPAS, you can run this SQL script on Datamart DB:

*Regular**PPAS**DatamartMaintenance_CleanupTmpData.sql*

The script can be executed when no ETL is running, on a monthly basis for example.

Chapter 5

USLAM Platform Distribution

5.1 Overview

The USLAM platform consists of three major components i.e. USLAM Services (which includes Web UI and Repository, Calculation Engines and Data Collector), USLAM ETL, and USLAM Reports (Web/I Server).

Normally the USLAM Services, USLAM ETL and USLAM Reports are deployed separately under different paths on a single or multiple machines.

Since the USLAM Services component comprises of three independent modules i.e. Web UI and Repository, Calculation Engines and Data Collector, therefore it can be deployed using two different deployment modes.

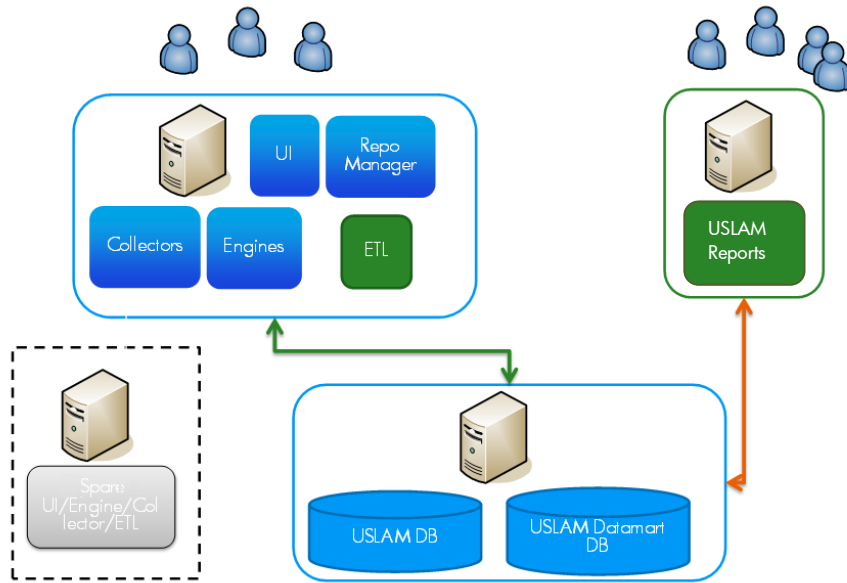
The first deployment mode involves all three components of the USLAM Services being deployed on a single JBoss server; this mode is called **Monolithic Platform**. The second deployment mode allows the three different modules of the USLAM Services to be deployed independently on multiple JBoss servers; this mode is called **Distributed Platform**.

In the following sections we will discuss the characteristics of the two deployment modes, and the procedures involved in performing the two types of platform deployments.

5.2 Monolithic Platform

A monolithic platform involves deploying all three modules of the USLAM Services component on a single JBoss server. The following sections discuss the characteristics and procedures involved in performing this type of platform deployment.

The following figure depicts a typical deployment for the monolithic platform:



5.2.1 Characteristics

The monolithic platform deployment may be selected for the following characteristics:

- It is easy to deploy
- Can be used even if there is only one server for deployment.

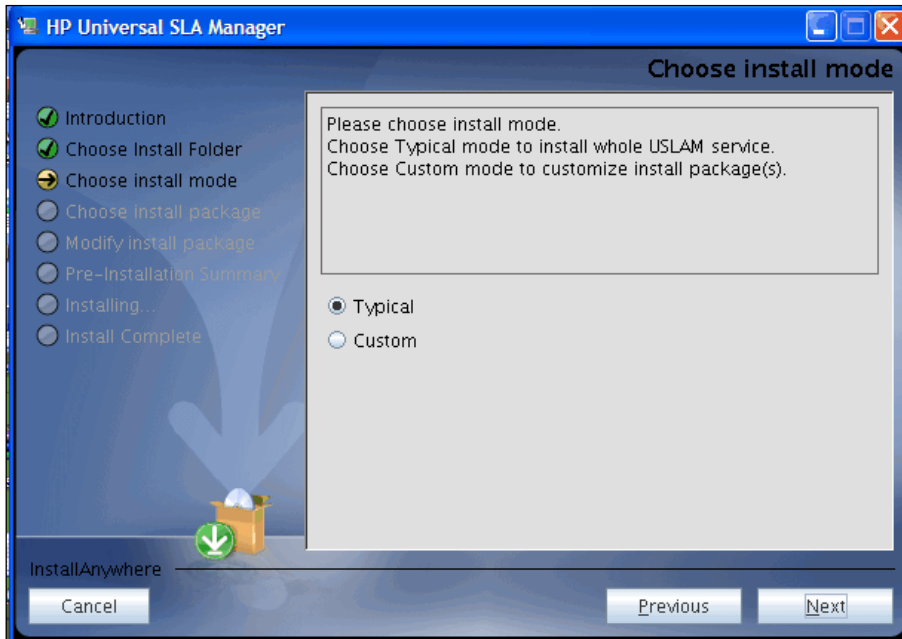
5.2.2 Deployment Instructions

5.2.2.1 USLAM Services

In order to perform a monolithic platform deployment for the USLAM Services component you should follow the USLAM Services installation steps described in *Section 2.1.2 USLAM Services Installation Wizard* in the *HP USLAM Installation and Configuration Guide*.

Once you proceed to the step where the Choose Install Mode screen displays, select the **Typical** installation mode and then continue to perform the remaining steps to complete the installation of the USLAM Services component.

An example of the Choose Install Mode screen is shown in the following figure:



This will complete the monolithic platform deployment for the USLAM Services modules on a single JBoss server.

5.2.2.2 USLAM ETL

For information about how to install USLAM ETL component, refer to *Chapter 3 Installing and Configuring USLAM ETL* in the *HP USLAM Installation and Configuration Guide*.

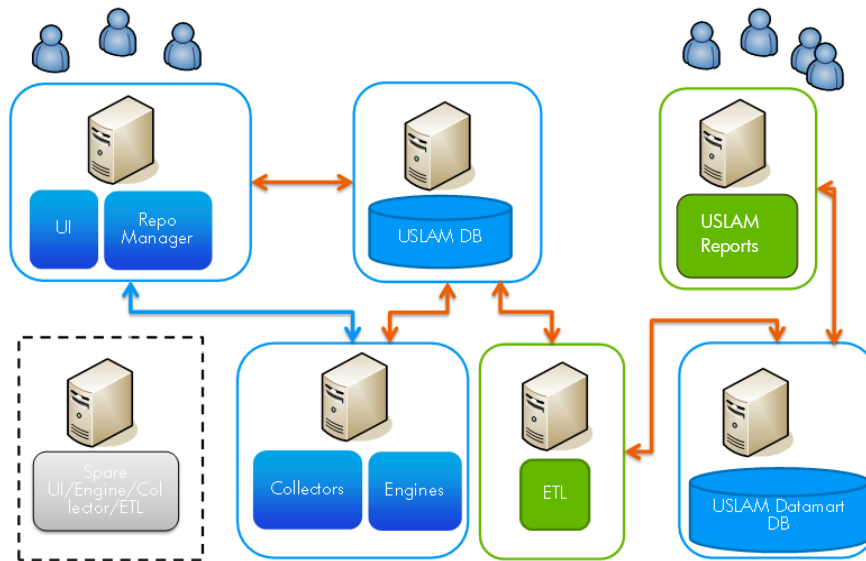
5.2.2.3 USLAM Reports

For information about how to install USLAM Reports component, refer to *Chapter 4 Installing and Configuring BO Report* in the *HP USLAM Installation and Configuration Guide*.

5.3 Distributed Platform

A distributed platform involves deploying the three independent modules of the USLAM Services component on multiple (different) JBoss servers. The following sections discuss the characteristics and procedures involved in performing this type of platform deployment.

The following figure depicts a typical deployment for the distributed platform:



5.3.1 Characteristics

The distributed platform deployment may be selected for the following characteristics:

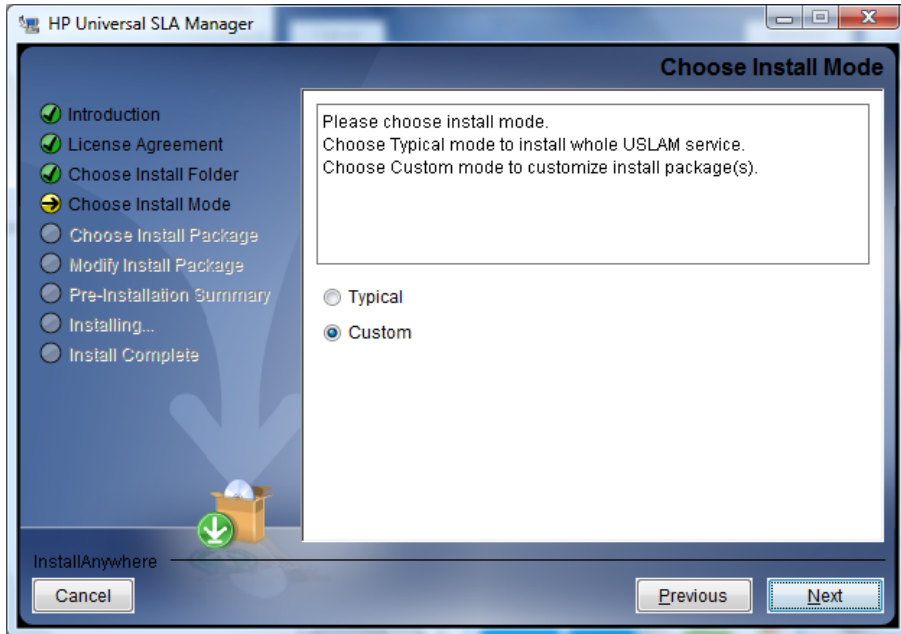
- Independent components can run separately
- Eliminates unnecessary effects caused by operational inter-dependability of different modules. For example, Web UI and Repository can work smoothly while SLA Engines module is down
- Performance is increased as compared to that of a monolithic platform
- Can be used if there are multiple (2 or more) servers for deployment.

5.3.2 Deployment Instructions

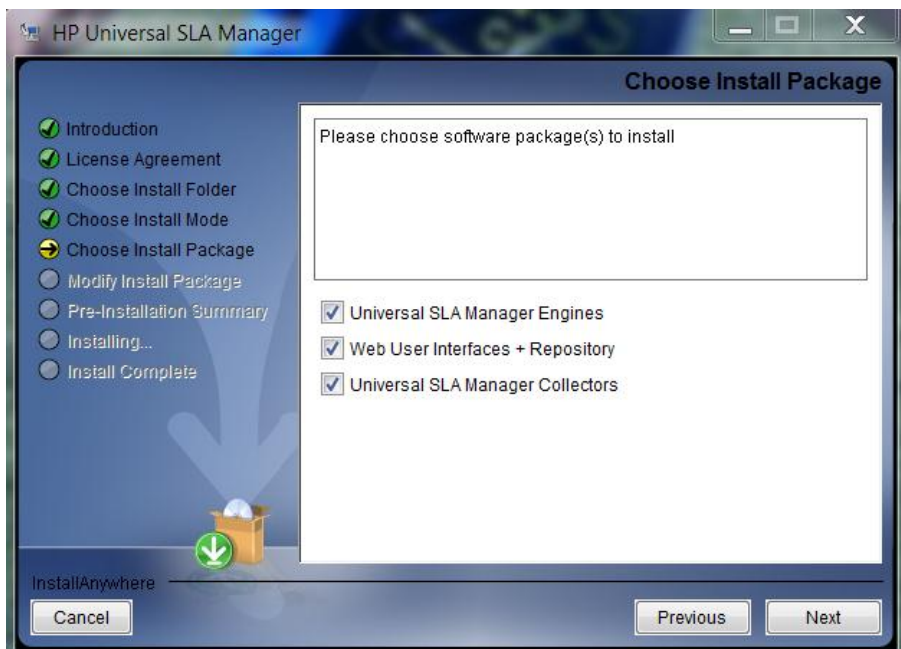
5.3.2.1 USLAM Services

In order to perform a distributed platform deployment for the USLAM Services component, you will be required to perform the following steps:

1. Refer to the USLAM Services installation steps described in Section 2.1.2 USLAM Services Installation Wizard in the HP USLAM Installation and Configuration Guide
2. Once you proceed to the step where the Choose Install Mode screen displays, select the **Custom** installation mode



3. Click [Next] to continue
4. The Choose Install Package screen displays



5. You can select one or two USLAM Services modules depending on the number of JBoss servers available (i.e. if you have two JBoss servers, you should select two modules to be installed on one server and the remaining one module on the second server. On the other hand if you have 3 JBoss servers, you can install each module on a different server, in this case select only one module)
6. Perform the remaining steps described in *Section 2.1.2 USLAM Services Installation Wizard* in the *HP USLAM Installation and Configuration Guide* to complete the installation of the USLAM Services component.

You will be required to repeat the above steps on each JBoss server depending on your desired deployment configuration (as explained in step 5 above).



You will be required to copy the installation kit **HP_USLAM_Services.bin** to the target JBoss server before repeating the installation steps on each (different) server.

This will complete the distributed platform deployment for the USLAM Services modules on multiple JBoss servers.

5.3.2.2 USLAM ETL

For information about how to install USLAM ETL component, refer to *Chapter 3 Installing and Configuring USLAM ETL* in the *HP USLAM Installation and Configuration Guide*.

5.3.2.3 USLAM Reports

For information about how to install USLAM Reports component, refer to *Chapter 4 Installing and Configuring BO Report* in the *HP USLAM Installation and Configuration Guide*.

Chapter 6

BI Report Publisher Administration

6.1 Introduction

The **USLAM BI Automatic Report Publisher** is an optional tool for the USLAM reporting solution which allows scheduling periodical report generation and publication.

This chapter describes how to run and configure the BI Automatic Report Publisher.

6.2 Starting BI Automatic Report Publisher

The BI Automatic Report Publisher supports two running mode: it runs either in an automatic mode for scheduled report generation or in manual mode for on demand report generation.

6.2.1 Automatic Report Publisher

To launch the automatic Report Publisher tool, the executable file **SendContractReportByEmail-Automatically.bat** should be run from following location:

```
<Report publisher install_Dir>/bin/
```

At start, the BI Automatic Report Publisher displays a window that must be kept opened in order to keep the tool running.



To run the tool automatically at system startup, you can add the *.bat* file to the Windows Startup programs list or create a scheduled task. Refer to Windows Help for more information about how to perform these tasks.



Do not start more than one *SendContractReportByEmail-Automatically.bat* at the same time to avoid duplicate processing.

6.2.2 On-demand Report Publisher

If you need to generate a one-time report, you can run the BI Report Publisher tool manually. In this case the Report Publisher requests following input parameters to identify the requested report:

- **SLA_ID**: the ID of the SLA for which the report is requested

- **DATE:** a date allowing to identify the reference period to report

The Report Publisher can be launched using one of the two approaches below.

1. Run the **SendContractReportByEmail-Manually.bat** executable file from following location:
`<Report Publisher install_Dir>/bin`

When starting, the BI Automatic Report Publisher displays a window prompting for the parameters listed above. Enter the requested information at prompt and press [Enter] to run the tool.

2. The tool can also be run from the command line

Open the command-line window (Start → Run → cmd.exe) and type the following command:

```
< Report Publisher Install_Dir>/bin/SendContractReportByEmail-
Manually.bat <SLA_ID> <DATE>
```

Press [Enter] to run the tool.



The default date format is **YYYY-MM-DD**.



To stop the BI Automatic Report Publisher tool, just close the window running the *.bat* program.

6.3 Configuring Report Generation Criteria

You can also configure the report generation conditions. This can be done by editing the report publisher configuration parameters in the database.

6.3.1 Setting a configuration parameter

All the Report Publisher related tables created during installation procedure are stored in the USLAM datamart schema. This is the case for the configuration parameters (stored in the `CTL_RPT_CONFIGURATION` table).

You can use your favorite sql tool in order to configure the parameters, for example:

```
# sqlplus SLA_DATAMART/SLA_DATAMART@uslamDM
```

```
SQL> update CTL_RPT_CONFIGURATION set PARAMETER_VALUE=<new value> where
PARAMETER_NAME='<name of the parameter to modify>';
```

Then commit your change as follows

```
SQL> commit;
```

For example:

```
SQL> update CTL_RPT_CONFIGURATION set PARAMETER_VALUE=5 where
PARAMETER_NAME='bi.reporting.sla_delay_period.MONTHLY';
```

```
1 row updated.
```

```
SQL> commit;
```

```
Commit complete.
```


6.3.2 Main Report Publisher configuration parameters

Parameter Name	Default Value	Description
bi.reporting.sla_delay_period.MONTHLY	10	The number of days the reports will be generated after the end of a monthly reference period
bi.reporting.sla_delay_period.WEEKLY	3	The number of days the reports will be generated after the end of a weekly reference period
bi.reporting.report_gen.doc_name	1 SLA, user requested period_Detailed report	The name of the report that will be generated for each SLA at the end of the reference period + delay.
bi.reporting.report_gen.folder_id	USLAM V4 standard reports/Achievements	The BOE folder where is located the definition of the report to generate.
bi.reporting.mail_carrier.mail_from	<i>The authentication mail address set at install</i>	The sender of the emails that will contain the published report.
bi.reporting.mail_carrier.mail_subject	SLA_DETAILS REPORT	The subject of the emails that will contain the published report.
bi.reporting.mail_carrier.mail_content	This is the SLA_DETAILS report.	The body text of the emails that will contain the published report.
bi.reporting.cooktime	1-6	The '24 hours' time range within which the tool works. The default value is 1-6, i.e. 1:00 to 6:00, in order to avoid any overload of the BOE server during the business day. If value is set to -1, it means 'disabled', i.e. the tool will work all day long.

Note that, if '*xxx_mail_xxx*' parameters are not set or set incorrectly, the Report Publisher will work anyway: only generating *.pdf* files without sending mails.



The Report Publisher tool must be restarted after any change of the configuration

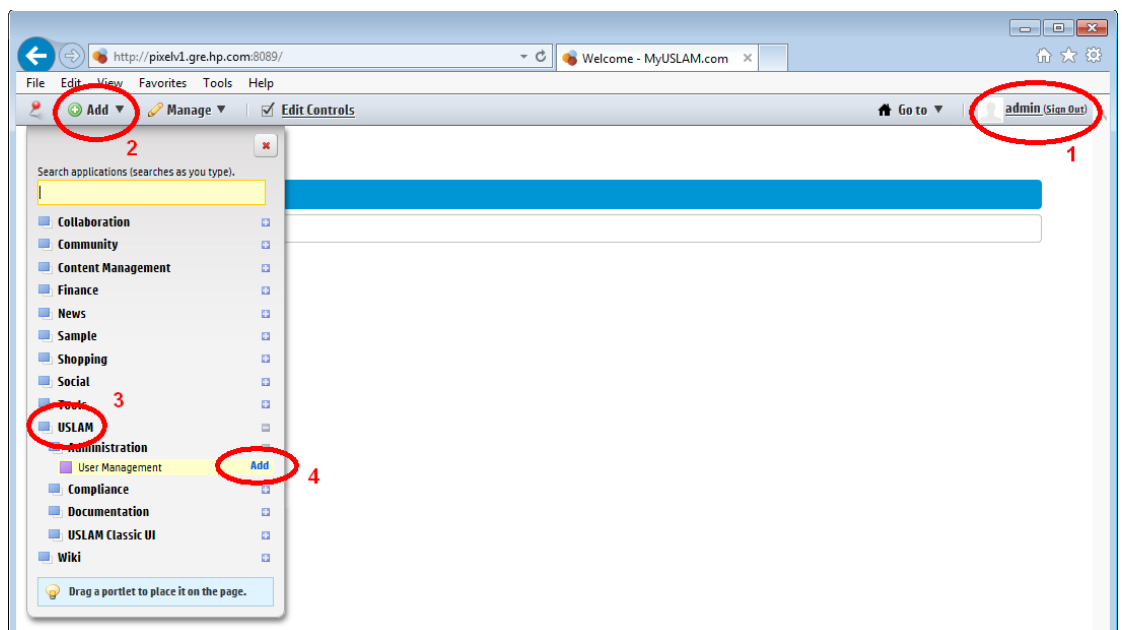
Chapter 7

MyUSLAM Portal Administration

7.1 Importing USLAM users

The only way to add users to MyUSLAM is by using the user management portlet. This portlet is part of the MyUSLAM portlets kit. To add this portlet:

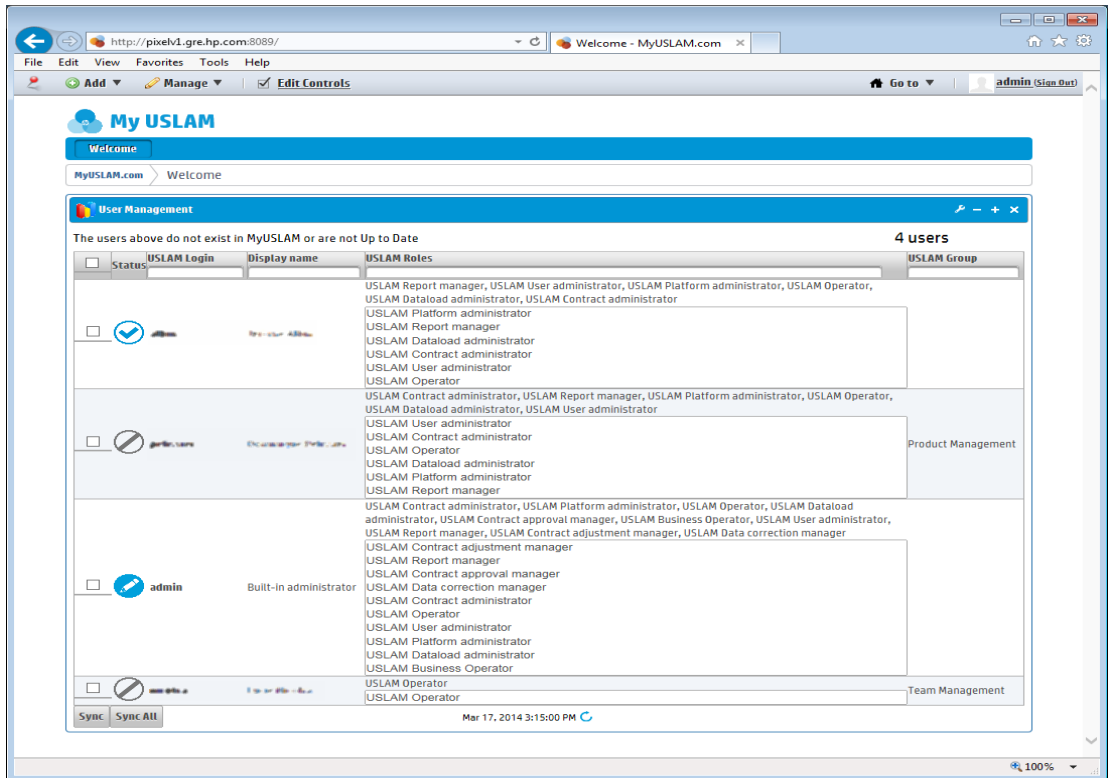
1. Log in as administrator
2. Open the Add menu
3. Open the USLAM folder
4. Add the User Management portlet



The aim of this Portlet is to give the MyUSLAM administrator the possibility to import the users from USLAM.

You can filter the results by entering the desired words on the search field of each column.

The users displayed after the filter operation can be imported by clicking on the sync button after selecting the users to import.



Chapter 8

USLAM Licensing

8.1 Overview

HP USLAM product requires a valid license to run properly. Obtaining and installing a license is mandatory and is under user responsibility.

Several types of licenses exist that are described below:

- **Trial License (InstantOnLicense):** this license is automatically generated after installation. This license allows a trial for 90 days and cannot be extended.
- **Permanent Licenses:** This is a commercial license that will never expire after being installed.
- **Evaluation Licenses:** This license is only valid for a specified time period, for e.g. 6 months, 1 year etc. It is dedicated to an evaluation usage.



BusinessObjects Master key embedded in USLAM products.

8.2 Installing or Updating a USLAM License

During the trial usage or if you get an expired license, you will need to get a valid license from the webware site and apply the following steps to install the license.

1. Get a new license file for USLAM. For more information, visit the [HP licensing system](#) website
2. Stop USLAM services.
3. Edit and copy/paste the license key to the license file available under `<Install_Dir>/license.txt`, and then save the modified file. The license is installed
4. Test your license file with the license tool **uslam_license** (see 8.4 *Checking the USLAM License*)
5. Restart USLAM services.

For more information about obtaining a license, stopping or starting USLAM services, please refer to the *HP USLAM Installation and Configuration Guide*.

8.3 Removing a USLAM License

You can easily remove an expired license, following these steps:

1. Stop USLAM services.

2. Edit and delete the license key from the license file available under `<Install_Dir>/license.txt`, and then save the modified file. The license will be removed
3. Test your license file with the license tool **uslam_license** (see 8.4 *Checking the USLAM License*) to verify if you still have a valid license installed
4. Restart USLAM services.

For more information about stopping or starting USLAM services, please refer to the HP USLAM Installation and Configuration Guide.

8.4 Checking the USLAM License status

A dedicated license tool is installed in `<installDir>/bin` to check your license validity. You can check and list all your licenses (on-demand), by running the following script:

<InstallDir>/bin/uslam_license

This script provides a status about USLAM license indicating:

- the installed valid and expired USLAM licenses (explicit license received by webware)
- the valid USLAM features (implicit license)
- the license check status: *License successfully check*, or *License check failed*



Please use this tool in case of license issue to troubleshoot a problem.

8.5 USLAM License management in details

USLAM License key are stored in the USLAM license file:
`<InstallDir>/license.txt`

After installation, a trial license will be generated in the license file and let you use the product for 90 days. The USLAM License file has to be updated with a commercial license before the end of this trial delay.

USLAM Service always checks the license at startup. If the license is not valid or expired, USLAM service will not start. You can view the license status message from the log file located at:

`<INSTALL_DIR>/jboss/server/default/log/slam_licensecheck.log`.

USLAM Logs

9.1 Configuring USLAM Services Logs

There are two ways to modify logs configuration, one way is to modify the **log4j** configuration file, and the other way is to directly update configuration items using the JMX web console. Both of these methods are explained in the following sections:

9.1.1 Log4j Configuration File

To modify existing logs configuration using the **log4j** configuration file, you will be required to modify the file located at the following path:

<Install_Dir>/jboss/server/default/conf/jboss-log4j.xml.

Each entry in this file is described by:

- Category: normally it is the package name for each USLAM (sub) component
- Priority: denotes the type of the captured information. Can be any one of TRACE, DEBUG, INFO, WARN, ERROR, FATAL
- Log file: output file name where the log information is recorded.

The administrator can update the existing **Category** or add a new category by modifying the content of the **jboss-log4j.xml** file. The following steps show how to modify the logs configuration:

1. Locate the Category for which you want to modify the logs configurations, in the jboss-log4j.xml file. For example, [**com.hp.sqm.slam.actionexecutor**]

```
<category name="com.hp.sqm.slam.actionexecutor">  
  <priority value="INFO" />  
  <appender-ref ref="debug-actionexecutor-appender"/>  
</category>
```

2. For instance you can update the [priority] property from **INFO** to **WARN**, as shown in the following example:

```
<category name="com.hp.sqm.slam.actionexecutor">  
  <priority value="WARN" />  
  <appender-ref ref="debug-actionexecutor-appender"/>  
</category>
```

3. Another example can be about changing the log name. Locate the [**debug-actionexecutor-appender**] property in the jboss-log4j.xml file

```

<appender name="debug-actionexecutor-appender" class="org.jboss.logging.appender.RollingFileAppender">
  <errorHandler class="org.jboss.logging.util.OnlyOnceErrorHandler"/>
  <param name="File" value="${jboss.server.log.dir}/slam_actionexecutor.log"/>
  <param name="Append" value="true"/>
  <param name="MaxFileSize" value="50MB"/>
  <param name="MaxBackupIndex" value="5"/>
  <layout class="org.apache.log4j.PatternLayout">
    <!-- logging Class and Method is slow, only use for debugging -->
    <param name="ConversionPattern" value="%d{yy-MM-dd HH:mm s.SSS } %-5p %l %M - %m%n"/>
  </layout>
</appender>

```

4. Update the value of the parameter File from slam_actionexecutor.log to new_slam_actionexecutor.log.

```

<appender name="debug-actionexecutor-appender" class="org.jboss.logging.appender.RollingFileAppender">
  <errorHandler class="org.jboss.logging.util.OnlyOnceErrorHandler"/>
  <param name="File" value="${jboss.server.log.dir}/new_slam_actionexecutor.log"/>
  <param name="Append" value="true"/>
  <param name="MaxFileSize" value="50MB"/>
  <param name="MaxBackupIndex" value="5"/>
  <layout class="org.apache.log4j.PatternLayout">
    <!-- logging Class and Method is slow, only use for debugging -->
    <param name="ConversionPattern" value="%d{yy-MM-dd HH:mm s.SSS } %-5p %l %M - %m%n"/>
  </layout>
</appender>

```



When **jboss-log4j.xml** is modified, you do not need to restart JBoss for the changes to take effect.

9.1.2 JMX Web Console

To access and modify the Logs configuration information using the JMX console, you will be required to perform the following steps:

1. Login to the JMX console (refer to *Section 3.4 JMX Administration*). The JMX Agent View window displays.
2. Click [jboss.system] in the Objects list on the left pane. The relevant properties display in the Agent View window
3. Click [service=Logging,type=Log4jService] to access the logs configuration page, as shown in the following example

Operation	Return Type	Description	Parameters
getLoggerLevel	java.lang.String	Displays the log level of a given logger.	logger java.lang.String The name of the logger to display. <input type="text"/> <input type="button" value="Invoke"/>
setLoggerLevel	void	Sets the log level for a given logger.	logger java.lang.String The name of the logger to set. level java.lang.String The log level to set. <input type="text"/> <input type="button" value="Invoke"/>
setLoggerLevels	void	Sets the levels of each logger specified by the given comma separated list of logger names.	loggers java.lang.String The list of loggers level java.lang.String The log level to set <input type="text"/> <input type="button" value="Invoke"/>

4. To acquire the current log level information about a specific USLAM component, enter the name of the component (e.g. com.hp.sqm.slam.poller) in the **Parameters** field assigned to the **getLoggerLevel** operation, and then click [Invoke]. This will return the current log level set for the specified component (e.g. WARN)
5. To modify the current log level for a specific USLAM component, enter the name of the component (e.g. com.hp.sqm.slam.poller) in the **Parameters: logger** field assigned to the **setLoggerLevel** operation, and then enter the new log level to be set for the component (e.g. INFO) in the **Parameters: level** field, and then click [Invoke]

- To modify the current log level for multiple USLAM components at once, you can either repeat the previous step, or you can enter multiple components (e.g. `com.hp.sqm.slam.poller`, `com.hp.sqm.slam.kiengine.monitor`) in the **Parameters: logger** field assigned to the **setLoggerLevels** operation, and then click [Invoke].



You can also perform the above operations using the command line. For example to perform the **setLoggerLevel** operation you can use the following command:
`${JBoss_HOME}/twiddle.sh -s <hostname> invoke jboss.system:service=Logging,type=Log4jService setLoggerLevel DEBUG.`

9.1.3 USLAM Services Log Files

All the log files relevant to the USLAM Services are located under the following path: `<Install_Dir>/jboss/server/default/log`.

Every USLAM components log information in a dedicated log file.

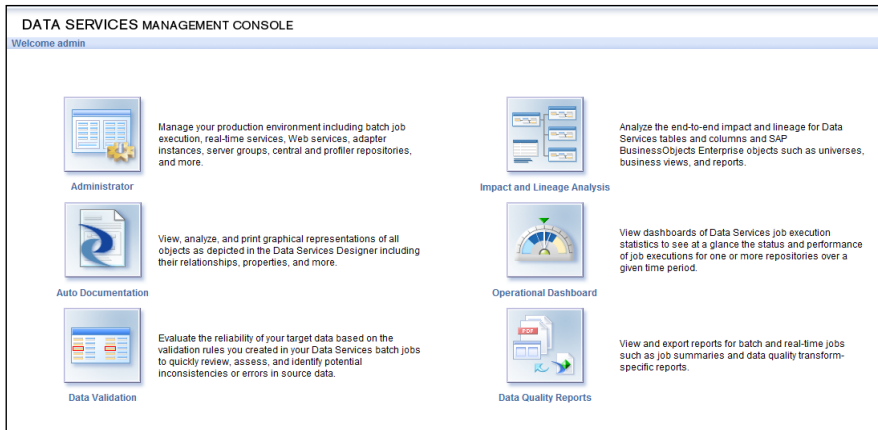
In addition to the per-component logging, several utilities log files are also provided as described below:

Log Name	Description
boot.log	Logs information about the start of the product
server.log	Encompasses the log information for all the product's components
slam_licensecheck.log	Logs information about the license checking
hibernate.log	Logs information about the hibernate requests to the database
slam_ui_sessions.log	Logs information about user login/logout to the Web UI.

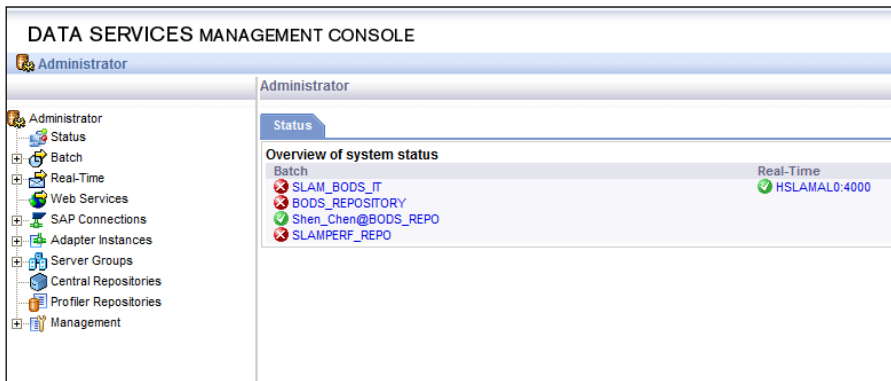
9.2 Viewing USLAM ETL Logs

To view the USLAM ETL logs using the BO Data Services Management Console, you will be required to perform the following steps:

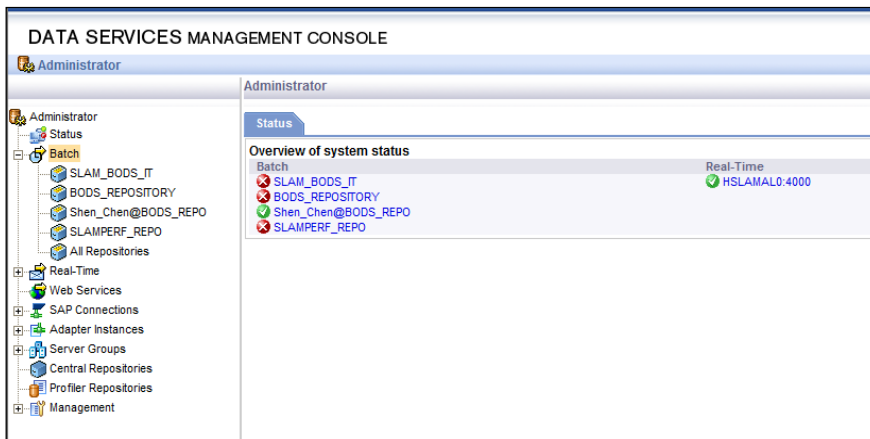
- Login to the BODS management console using the URL:
`http://<web_server>:8180/DataServices/launch/logon.do`. `<web_server>` is the IP address or the hostname of the Web server, and `<port_number>` is the Port number of the Web server (default port: 8180)
- The main dashboard displays



3. Click [Administrator]. The Administrator window displays



4. Expand [Batch] from the **Administrator** tree in the left pane. The relevant job groups display in the tree



5. Click the name of a system, for example, **SLAM_BODS_IT**. All jobs in this group will be displayed in the main window

Select	Status	Job name	System configuration	Job Server	Job Information	Start time	End time
<input type="checkbox"/>	✓	TEST		HSLAMAL0:3500	Trace,Monitor,Error,Performance Monitor	Sep 11, 2010 5:51:47 PM	Sep 11, 2010 5:52:12 PM
<input type="checkbox"/>	✓	New_Job		HSLAMAL0:3500	Trace,Monitor,Error,Performance Monitor	Sep 11, 2010 5:40:51 PM	Sep 11, 2010 5:40:51 PM
<input type="checkbox"/>	✓	Copy_1_TEST		HSLAMAL0:3500	Trace,Monitor,Error,Performance Monitor	Sep 11, 2010 3:40:25 PM	Sep 11, 2010 3:40:59 PM
<input type="checkbox"/>	✓	JB_Dim_Sla_Clause		HSLAMAL0:3500	Trace,Monitor,Error,Performance Monitor	Sep 10, 2010 9:57:23 PM	Sep 10, 2010 9:57:49 PM
<input type="checkbox"/>	✓	JB_Fct_Sla_Sl_Inc		HSLAMAL0:3500	Trace,Monitor,Error,Performance Monitor	Sep 10, 2010 6:29:50 PM	Sep 10, 2010 6:33:32 PM
<input type="checkbox"/>	✓	JB_Dim_Sla_Item		HSLAMAL0:3500	Trace,Monitor,Error,Performance Monitor	Sep 10, 2010 6:28:55 PM	Sep 10, 2010 6:29:32 PM
<input type="checkbox"/>	✓	JB_Fct_Sla_Item_Status		HSLAMAL0:3500	Trace,Monitor,Error,Performance Monitor	Sep 10, 2010 6:09:17 PM	Sep 10, 2010 6:10:26 PM
<input type="checkbox"/>	✓	JB_Fct_Sla_Clause_Status		HSLAMAL0:3500	Trace,Monitor,Error,Performance Monitor	Sep 10, 2010 5:11:53 PM	Sep 10, 2010 5:12:40 PM
<input type="checkbox"/>	✓	JB_Dim_Sl_Instance		HSLAMAL0:3500	Trace,Monitor,Error,Performance Monitor	Sep 10, 2010 2:32:27 PM	Sep 10, 2010 2:33:05 PM
<input type="checkbox"/>	✓	JB_Dim_Calendar		HSLAMAL0:3500	Trace,Monitor,Error,Performance Monitor	Sep 10, 2010 2:28:37 PM	Sep 10, 2010 2:29:09 PM
<input type="checkbox"/>	✓	JB_Dim_Service_Offering		HSLAMAL0:3500	Trace,Monitor,Error,Performance Monitor	Sep 10, 2010 2:21:11 PM	Sep 10, 2010 2:21:37 PM
<input type="checkbox"/>	✓	JB_Dim_Service		HSLAMAL0:3500	Trace,Monitor,Error,Performance Monitor	Sep 10, 2010 2:18:38 PM	Sep 10, 2010 2:19:15 PM
<input type="checkbox"/>	✓	JB_Dim_Service_Definition		HSLAMAL0:3500	Trace,Monitor,Error,Performance Monitor	Sep 10, 2010 2:15:15 PM	Sep 10, 2010 2:15:43 PM
<input type="checkbox"/>	✓	JB_Dim_Customer		HSLAMAL0:3500	Trace,Monitor,Error,Performance Monitor	Sep 10, 2010 2:12:56 PM	Sep 10, 2010 2:13:14 PM
<input type="checkbox"/>	✓	JB_Dim_Time		HSLAMAL0:3500	Trace,Monitor,Error,Performance Monitor	Sep 10, 2010 1:35:22 PM	Sep 10, 2010 1:35:40 PM
<input type="checkbox"/>	✓	JB_Lkp_Reps_Dictionary		HSLAMAL0:3500	Trace,Monitor,Error,Performance Monitor	Sep 10, 2010 1:31:15 PM	Sep 10, 2010 1:31:27 PM

6. To view the trace information for a job, click [Trace] in the **Job Information** column. The log trace displays for the selected job

```

Repository: SLAM_BODS_IT
Job Trace Log Job Monitor Log Job Error Log
Log: trace_09_11_2010_17_51_46_362__2b86e4f5_98da_4f4a_80a4_a022e7002a14.txt Job Server: HSLAMAL0:3500
Page: 1 View Full

Job name: TEST
(12.2) 09-11-10 17:51:46 (22636:2343913328) JOB: The initial environment locale <eng_us_utf-8> has been coerced to <Unicode (UTF-16)> ().
(12.2) 09-11-10 17:51:47 (22636:2343913328) JOB: Reading job <2b86e4f5_98da_4f4a_80a4_a022e7002a14> from the repository; Server version is <12.2.1.3>; Repository version is <12.2.0.0000>.
(12.2) 09-11-10 17:51:47 (22636:2343913328) JOB: Current directory of job <2b86e4f5_98da_4f4a_80a4_a022e7002a14> is <home\hpuslamet\HP_Universal_SLAM_BO_ETL\businessobjects\dataservices\bin>.
(12.2) 09-11-10 17:51:47 (22636:2343913328) JOB: Starting job on job server host <HSLAMAL0>, port <3500>.
(12.2) 09-11-10 17:51:47 (22636:2343913328) JOB: Job <TEST> of name <20100911175147226362343913328> is initiated by user <hpuslamet\>.
(12.2) 09-11-10 17:51:47 (22636:2343913328) JOB: Processing job <TEST>.
(12.2) 09-11-10 17:51:47 (22636:2343913328) JOB: Initializing transcoder for datastore <DS_Stem_Target> to transcode between engine codepage<Unicode (UTF-16)> and datastore codepage <<DEFAULT>>.
(12.2) 09-11-10 17:51:47 (22636:2343913328) JOB: Initializing transcoder for datastore <Ds_Bods_Repo> to transcode between engine codepage<Unicode (UTF-16)> and datastore codepage <<DEFAULT>>.
(12.2) 09-11-10 17:51:47 (22636:2343913328) JOB: Initializing transcoder for datastore <Ds_Stem_Repo> to transcode between engine codepage<Unicode (UTF-16)> and datastore codepage <<DEFAULT>>.
(12.2) 09-11-10 17:51:47 (22636:2343913328) JOB: Optimizing job <TEST>.
(12.2) 09-11-10 17:51:47 (22636:2343913328) JOB: Job <TEST> is started.
(12.2) 09-11-10 17:51:49 (22636:2343913328) PRINTFN: Job root path is <home\hpuslamet\HP_Universal_SLAM_BO_ETL>.
(12.2) 09-11-10 17:51:49 (22636:2343913328) PRINTFN: Job TEST last job instance status checking...
(12.2) 09-11-10 17:51:49 (22636:2343913328) PRINTFN: status synchronizing start
(12.2) 09-11-10 17:51:50 (22636:2343913328) PRINTFN: status synchronizing finish
(12.2) 09-11-10 17:51:51 (22636:2343913328) PRINTFN: ETL job execution start time : 20100911 17:50:53
(12.2) 09-11-10 17:51:51 (22636:2343913328) PRINTFN: JOB PARAMETERS AND THEIR PASSED VALUES
  
```

7. Select the [Job Monitor Log] tab to view the monitor log information

```

Repository: SLAM_BODS_IT
Job Trace Log Job Monitor Log Job Error Log
Log: monitor_09_11_2010_17_51_46_362__2b86e4f5_98da_4f4a_80a4_a022e7002a14.txt Job Server: HSLAMAL0:3500

Job name: TEST
  
```

Path name	State	Row Count	Elapsed time (Sec)	Absolute time (Sec)
-DF_Stg_Sl_Status_Change\ry_join	STOP	0	0.000	22.363
-DF_Stg_Sl_Status_Change\TCR6_11	STOP	0	0.000	22.363
-DF_Stg_Sl_Status_Change\Ftable_Comparison: 0	STOP	0	0.000	22.363
-DF_Stg_Sl_Status_Change\Ftable_Comparison: 1	STOP	0	0.000	22.363
-DF_Stg_Sl_Status_Change\STO_SL_STATUS_CHANGE_STO_SL_STATUS_CHANGE	STOP	0	0.000	22.676

1. Select the [Job Error Log] tab to view error log information

```

Repository: SLAM_BODS_IT
Job Trace Log Job Monitor Log Job Error Log
Log: error_09_11_2010_17_51_46_362__2b86e4f5_98da_4f4a_80a4_a022e7002a14.txt Job Server: HSLAMAL0:3500

Job name: TEST
Empty file received.
  
```



For more information about working with logs using the BODS Management Console, refer to the *BusinessObjects Data Services Administration Guide*.

9.3 Viewing USLAM Reports Log

9.3.1 BusinessObjects Enterprise

Several administrative tools can be used in order to monitor the BO platform:

- BusinessObjects Enterprise Central Management Console
- Central Configuration Manager

For more information please refer to the *BusinessObjects Enterprise Administration Guide*.

9.3.2 USLAM BI Auto Report Publisher

All the log files relevant to the BI Automatic Report Publisher tool are located under the following path: <Report Publisher Install_Dir>/logs.

9.4 Configuring MyUSLAM Logs

There are two different ways for configuring MyUSLAM Portal Logs, the first one is through the **standalone.xml** configuration file and the second one through **Liferay's Control Panel**, this latter is the recommended approach. Both these ways are explained in the following sections.

9.4.1 standalone.xml Configuration File

This approach is similar to USLAM's Log4j approach. To modify existing logs configuration using the **standalone.xml** configuration file, you will have to modify the file located at the following path:

```
<Install_Dir>\jboss\standalone\configuration\standalone.xml.
```

In this file you'll find entries that look like the following:

```
<logger category="com.arjuna">
  <level name="WARN"/>
</logger>
```

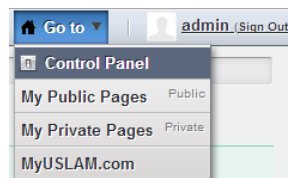
- logger category: normally it is the package name for each component
- level name: denotes the type of the captured information. Can be any one of TRACE, DEBUG, INFO, WARN, ERROR, FATAL

You can modify the existing one or add new loggers for new components.

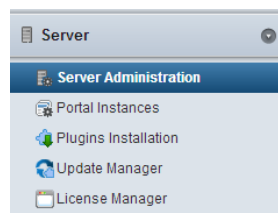
9.4.2 Liferay Control Panel (Advisable approach)

This approach is done directly in the MyUSLAM Portal, once you sign in as an administrator you can follow the following steps to configure the logging of the Portal.

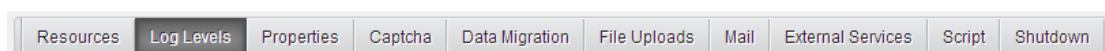
1. Go to the Control Panel.



2. Click on **Server Administration** from **Server** section in control panel.



3. Click on **Log Levels**.



You will notice that, there are various **Categories** and **Log level** defined. You can modify the existing **Loggers** by changing the **Levels** or add new ones;

4. Click on **Add Category**.
5. Give the **package name**. for example **com.hp.uslam.portlets.slastatus**
6. Set the **log level** from the drop down menu.
7. Save you configuration.

The logger levels follow the following order.

1. DEBUG
2. INFO
3. WARN
4. ERROR
5. FATAL

which means,

- if we set Log level to WARN then only the WARN, ERROR and FATAL level logs will be displayed.
- If we set log level to ERROR then only the ERROR and FATAL will be displed.

Once the logs are configured, the log files are available at this directory:

<Install_Dir>\jboss\standalone\log

- The file **server.log** contains all server logs including MyUSLAM ones.
- The file **myuslam.log** contains only MyUSLAM components logs.

Appendix A

1. List of ports used by USLAM

The USLAM components use several ports that may require a specific firewall configuration for the product to be able to execute properly.

Here is the list of port that should be considered while configuring the USLAM platform.

Component	Host	Default Port number	Usage
Web UI	Repository component's server	8080	JBoss Web HTTP connector socket.
BO Xi	Report server	8080	Listening port of the Business Objects server
SMTP Action Executor	Engines component's server	25	Mail server port
SNMP Action Executor	Engines component's host	161	SNMP agent port
Web UI	Repository component's server	389	The listening port of your LDAP Server
ETL	ETL server	8180	Port number of the BODS management console's web server
MyUSLAM	Portal Server	8089	Default port of MyUSLAM, can be change when configuring the portal for the first time