HP Operations Manager for UNIX 9.10

for the UNIX operating system

Administration UI

Software Version: 9.1.0

Administration and Configuration Guide

Document Release Date: August 2010
Software Release Date: August 2010

Legal Notices

Warranty

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.

The information contained herein is subject to change without notice.

Restricted Rights 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 Notice

Administration UI includes software developped by various open-source projects and organizations as listed below. The corresponding files and components are copyright to the corresponding organization or vendor and all rights reserved. The software files and components distributed under the open-source licenses are distribted on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the license of the corresponding project for specific rights and limitations under the license. Depending on the license, any product derived from the products may not be called with the name of the project nor may the name of the project appear in their name, without prior written permission. For written permission, please contact the corresponding project owner by visiting the corresponding project home page as listed below.

We greatly appreciate the work of those projects and try to contribute as much as possible to some of those projects in order to compensate their contributions.

This product includes software developed by the Acegi System for Spring Project (http://acegisecurity.org/)

This product includes software developed by the ActiveMQ project

(http://activemq.org/)

This product includes software developed by the Ant-Contrib project

(http://sourceforge.net/projects/ant-contrib)

This product includes software developed by the Antlr project

(http://www.antlr2.org)

This product includes software developed by the Apache Software Foundation

(http://www.apache.org/). These are “Ant”, “Batik”,“BCEL”, “Cocoon”, “Commons”, “Derby”, “Excalibur”, “FOP”, “Forrest”, “ FTPServer”, “Jasper”, “Log4j”, “Lucene”, “ORO”, “POI”, “Solr”, “Tuscany”, “Velocity”, “Xalan”, “Xerces” and “XML RPC”, “XML Security”.

This product includes software developed by the Baekmuk project
(http://kldp.net/projects/baekmuk/)
This product includes software developed by the BSF project
(http://jakarta.apache.org/bsf/)
This product includes software developed by the dnsjava project
(http://www.dnsjava.org/)
This product includes software developed by the Docbook project
(http://www.docbook.org/)
This product includes software developed by the dom4j project
(http://dom4j.org/)
This product includes software developed by the Drools project
(http://drools.codehaus.org/)
This product also includes software developed by the dsmltools project
(http://www.dsmltools.org/)
This product also includes software developed by the EditArea project
(http://www.cdolivet.com/editarea/)
This product also includes software developed by the Exist project
(http://www.exist-db.org/)
This product also includes software developed by the Fins project
(http://cocoondev.org/main/117-cd/29-cd.html)
This product also includes software developed by the Fireflysung project
(http://www.study-area.org/apt/firefly-font/)
This product also includes software developed by the Groovy project
(http://groovy.codehaus.org/)
This product also includes software developed by the GWT project
(http://code.google.com/webtoolkit/)
This product also includes software developed by the ICU4C project
(http://www.ibm.com/software/globalization/icu/)
This product also includes software developed by the ICU4J project
(http://www.ibm.com/software/globalization/icu/)
This product also includes software developed by the j2ssh project
(http://sourceforge.net/projects/sshtools/)
This product also includes software developed by the Janino project
(http://www.janino.net/)
This product also includes software developed by the Jasper project
(http://tomcat.apache.org/)
This product also includes software developed by the Jaxen project

(http://jaxen.org/)

This product also includes software developed by the Jaxup project

(http://klomp.org/jaxup/)

This product also includes software developed by the JDOM project

(http://www.jdom.org/)

This product also includes software developed by the Jencks project

(http://jencks.org/)

This product also includes software developed by the Jetty project

(http://jetty.mortbay.org/jetty/)

This product also includes software developed by the JFreeChart project

(http://www.jfree.org/jfreechart/)

This product also includes software developed by the JPam project

(http://jpam.sourceforge.net/)

This product also includes software developed by the mimeutil project

(http://sourceforge.net/projects/mime-util/)

This product also includes software developed by the jRegistryKey project

(http://sourceforge.net/projects/jregistrykey/)

This product also includes software developed by the Jsch project

(http://www.jcraft.com/jsch/)

This product also includes software developed by the Jsdifflib project

(http://snowtide.com/jsdifflib)

This product also includes software developed by the Jython project

(http://www.jython.org)

This product also includes software developed by the MX4J project

(http://mx4j.sourceforge.net)

This product also includes software developed by the Netbeans CVS project

(http://javacvs.netbeans.org/library)

This product also includes software developed by the openadaptor project

(http://www.openadaptor.org)

This product also includes software developed by the Oracle JDBC project

(http://www.oracle.com)

This product also includes software developed by the Prefuse project

(http://prefuse.org)

This product also includes software developed by the Quartz project

(http://www.opensymphony.com/quartz)

This product also includes software developed by the Rhino project

(http://www.mozilla.org/rhino)

Trademark Notices

Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for recent updates or to verify that you are using the most recent edition of a document, go to:

http://h20230.www2.hp.com/selfsolve/manuals

This site requires that you register for an HP Passport and sign in. To register for an HP Passport ID, go to:

http://h20229.www2.hp.com/passport-registration.html

Or click the **New users - please register** link on the HP Passport login page.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HP sales representative for details.

Support

Visit the HP Software Support Online web site at:

www.hp.com/go/hpsoftwaresupport

This web site provides contact information and details about the products, services, and support that HP Software offers.

HP Software online support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the support web site to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HP support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract. To register for an HP Passport user ID, go to:

http://h20229.www2.hp.com/passport-registration.html

To find more information about access levels, go to:

http://h20230.www2.hp.com/new_access_levels.jsp

(intentionally left blank)

Conventions

Font Style	Explanation
Boldface	Words in boldface type represent programs and commands.
Capitalization	Capitalized first letters represent company or product names.
`Computer font`	Words in computer font represent file or path names, command syntax statements, prompts or messages that appear on your screen or text you should type on your workstation or terminal.
Italics	Words in italics represent variables in syntax statements or words that are emphasized in the text.
`{ }`	Represents required elements in a syntax statement. When several elements are separated by the | symbol you must select one of the elements.
`[ ]`	Represents optional elements in a syntax statement.

In all examples the default installation path of Administration UI

on HP-UX, Sun Solaris, Linux is displayed. It is

```
# /opt/OV/OMU/adminUI/
```

(intentionally left blank)

1 Introduction

Overview

Your company's business success relies on high-quality IT services and IT infrastructure agility. To keep your IT services available and well performing, you need a proven operations management solution that gives you control over your ever-changing IT infrastructure. That solution is HP Operations Manager on UNIX (OMU), SOLARIS (OMS) or LINUX (OML).

HP Operations Manager discovers, monitors, controls and reports on the availability and performance of your heterogeneous, large-scale IT environment. It consolidates information for all IT components that control your business: network, systems, storage, databases, and applications. With its service-driven approach, it shows what IT problems affect your business processes, helping you to focus on what's most important for your company's business success.

For a general overview about OMU, OMS and OML feature set, refer to the HP Operations Manager Concepts Guide, which is available in PDF format on the HP product manual website (see below).

About this Document

This document provides information about the architecture, configuration, maintenance and troubleshooting of Administration UI.

This manual applies both to Administration UI for OMU, OMS and OML and any reference throughout this manual to HPOM includes all three versions. System specific specialities are highlighted accordingly.

Please note, that as a result of regular program updates, some information in the printed manual may vary from that found in the online help. For the same reason, there may be slight differences in the presentation of the program's interface. Most screenshots in this manual have been taken during development and may not reflect final content.

Useful Links & Contact Information

Check the following web site periodically for the latest versions of this and other HPOM manuals:

`http://support.openview.hp.com/selfsolve/manuals`

Select “Operations Manager for UNIX“ and version 9.0.

HPOM patches can be downloaded from the following website:

`http://support.openview.hp.com/patches/patch_index.jsp`

It is recommended also to check the current HPOM 8/9 SUMA (support matrix):

`http://support.openview.hp.com/selfsolve/document/KM323488`

2 Architecture & References

Overview

This chapter describes the underlying components of Administration UI.

First an overview of the architecture, the different modules and communication properties will be given.

Furthermore, the directory and file structure and the default passwords will be explained.

The information in this section covers the following areas:

- Architecture Overview on page 4

- Communication and Ports on page 5

- Directory Layout Overview on page 6

- Default Passwords on page 13

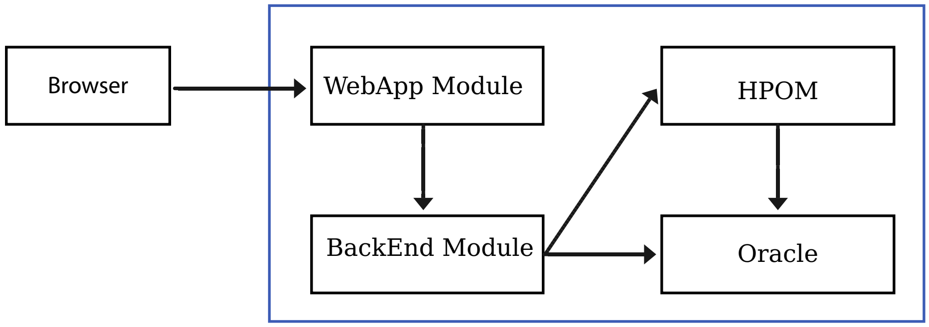
Architecture Overview

The Administration UI is implemented based on a three-tier architecture (see Figure 1 on page 4).

- **Browser**
  The user front-end is a regular web browser like Mozilla Firefox or Internet Explorer. Therefore, no additional software is required on the end-user system. All users can work concurrently.
- **Web Application Module**
  Via the web browser the user connects to the Web Application module (referred to as “WebApp” throughout this manual). It is responsible for generating the dynamic web pages, central data storage and data processing.
- **BackEnd Module**
  This component on one hand interacts with the IT management application (e.g. HPOM server). For simple read-only listings (Example: list all policies) the BackEnd module connects directly to Oracle. For add/modify/delete operations on any HPOM object the BackEnd accesses the HPOM API.
  On the other hand it provides this data as XML through an URL schema to the Administration UI WebApp module.

  The Administration UI WebApp and BackEnd modules always have to be installed on the same machine as the management framework server.

Figure 1 Basic Architecture



Communication and Ports

The communication within Administration UI is TCP/IP based. All port numbers listed here are the default ports. They are generally defined during the installation of the Administration UI software. But it is also possible to modify these settings after the installation. For this purpose advanced scripts exist which help you modfying these ports, see section Advanced Tasks on page 33.

- **9662** (HTTP), **9663** (HTTPS) - The WebApp listens on port 9662 for HTTP and on 9663 for HTTPS requests.
  If a firewall exists between the end-user's network and the HPOM management port(s) 9662 and/or 9663 have to be opened (direction: end-user -> HPOM).

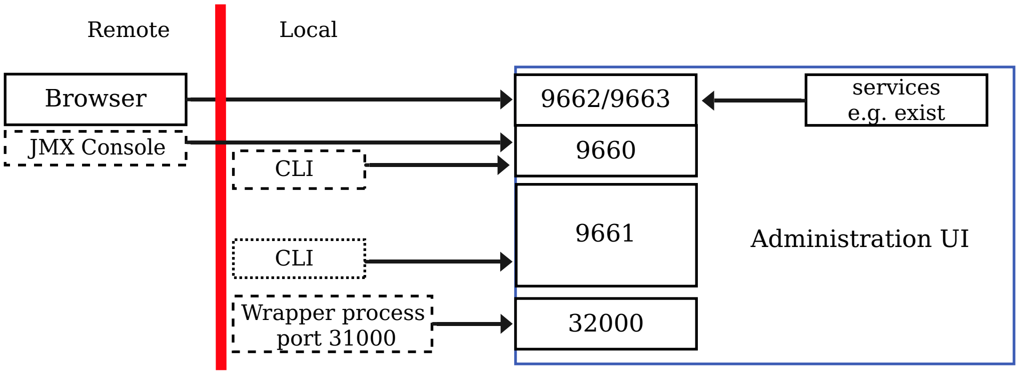
  Some internal services also exist which connect locally to this port, for example, the exist database used to store Administration UI users, user groups, all user roles. The solr service responsible to update the search index also uses port 9662.
- **9661** - All shell commands using `/opt/OV/OMU/adminUI/adminui <cmd>` connect locally to port 9661. Other WebApps or BackEnds or some external CLI or API also use port 9661.
- **9660** - This port is only locally via the CLI for troubleshooting purposes. No firewall opening is usually needed.
- **32000** - Local communication between the wrapper process (port 31000) and the JRE running in a java process, e.g. stop, dump etc.

Therefore, no port openings are generally required for ports 9660, 9664 and 32000 since they are only used locally.

This can be verified by using the `netstat` command, as shown in the following example:

```
[root@deli:/opt/OV/OMU/adminUI/] netstat -an | grep 966
tcp        0      0  *.9660            *.*               LISTEN
tcp        0      0  *.9661            *.*               LISTEN
tcp        0      0  *.9662            *.*               LISTEN
tcp        0      0  *.9663            *.*               LISTEN
[root@deli:/opt/OV/OMU/adminUI/] netstat -an | grep 320
tcp        0   0  127.0.0.1.31000  127.0.0.1.32000      ESTABLISHED
tcp        0   0  127.0.0.1.32000  127.0.0.1.31000      ESTABLISHED
tcp        0   0  127.0.0.1.32000  *.*                  LISTEN
```

Figure 2 Communication and Ports



CLI = Command Line Interface   API = MIDAS API

Directory Layout Overview

This section explains the basic directory layout. The Administration UI installer contains all necessary files & applications required to install and run the program. All Administration UI components are installed into one directory.

The suggested default location is `/opt/OV/OMU/adminUI/`

Inside this directory each component has its own sub-directory.

The most important files and directories are:

`/opt/OV/OMU/adminUI/adminui`	- central script to control Administration UI
`/opt/OV/OMU/adminUI/conf/`	- configuration files
`/opt/OV/OMU/adminUI/data/`	- data location with downloads, XML DB etc.
`/opt/OV/OMU/adminUI/logs/`	- log files

The next sections will give a more detailed overview on the main files and directories.

Main Directory

File or directory	**Purpose**
AdminUI_InstallLog.log	main installation log file
/Uninstall_AdminUI	contains Uninstaller binary
adminui	legacy central control script
/assemblies	available service assemblies
/backup	target folder for local backups
/bin	scripts and binaries
/checksums	checksums location
/components	available Java Business Integration components
/conf	central config file location (see separate section)
/data	contains user data: downloads, archive, XML DB (see separate section)
/datassemblies	default data assemblies used for initial initialization of XML DB, tasks etc.
/deploy	actively deployed assemblies
/docs	third party open source licenses
/install	actively deployed JBI components
installation.log	second installation log
/jre	bundled JAVA SDK
/lib	shared jar files & native libraries
/logs	log, audit and task logs files (see separate section)
`adminui`	central control script
midas_env.sh	contains environment variables HP-UX, SUN, Linux
midas_server.pid	contains server PID at runtime
run.xml	ant file used by `adminui`
/webapps	deployed WebApp
/webassemblies	deployed webassemblies
/work	Servicemix deployment and temp files
wrapper	service wrapper needed to run the application

The Configuration Directory

`/opt/OV/OMU/adminUI/conf` is the default location for all configuration data.

File or directory	Purpose
ant/config.xml	legacy configuration file for ant tasks
auth.properties	SU layer configuration files
auth.xml	SU layer configuration files
backend_local.xml	main local backend configuartion & capabilities
becore.properties	product feature configuration file
cocoon.properties	SU layer configuration files
config.properties	main local configuration file
core.properties	SU layer configuration files
data_local.xml	belongs to ./datassemblies
derby.properties	not used
exec.properties	SU layer configuration files
exist/	configuration files for XML DB, do not touch
file.properties	SU layer configuration files
fonts/	not used
groovy/	do not touch
jetty.properties	SU layer configuration files
jetty.xml	SU layer configuration files jetty configuration file for webserver configuration (ports etc.)
ldap.properties	configuration file for LDAP
local.properties	SU layer configuration files for BackEnd adapter
lock.properties	SU layer configuration files
log4j.xml	central logging configuration file
magic.mime	for mime type detection
midas_analyzer.xml	configuration file for ./`adminui` analyze command
mime-types.properties	for mime type detection
mime.types	for mime type detection

File or directory	**Purpose**
opccfg.properties	SU layer configuration files
ovcert.properties	SU layer configuration files
ovcoda.properties	SU layer configuration files
ovconfig.properties	SU layer configuration files
ovo/	do not touch
ovoappl.properties	SU layer configuration files
ovoconfig.properties	SU layer configuration files, HPOM and Oracle connection settings
ovodistrib.properties	SU layer configuration files
ovoinstall.properties	SU layer configuration files
ovosvc.properties	SU layer configuration files
quartz.properties	global configuration file for all schedulers
repository/	Contains internal configuration files. Do not edit.
schema/	XML schemas for documents
servicemix/	Servicemix components configuration files incl HTTPS
servingxml/	servicingxml config files. Do not edit.
ssh.properties	SU layer configuration files
stylesheets/	Server side xsl files, e.g. ./`adminui` backend output
task.properties	SU layer configuration files
terminal.properties	SU layer configuration files
user.properties	SU layer configuration files
usermgmt.properties	SU layer configuration files
velocity.properties	do not touch
wacore.properties	product feature configuration file
webapp.properties	SU layer configuration files

The Data Directory

The following list shows the contents of the data directory of `/opt/OV/OMU/adminUI/data`, which contains user-specific data. If it already exists, the data directory is not modified during installation or uninstallation; if the data directory does not exist, it is created during the installation process.

File or directory	**Purpose**
archive/	All archived items go in her *.zip, *tar.gz.
clipboard/	All downloads go into this directory
init/	Data to reset the XML DB (loaded when ./`adminui` init is run).
path/	Path alias data files. Do not edit manually.
sandbox/	Currently not used.
scratchpad/	Currently not used.
task/	Task data files. Do not edit manually.
txlog/	Work directory for transaction logs of servicemix
xmldb/	WebApp: database with user model.

The Log Directory

Inside `/opt/OV/OMU/adminUI/log` all main log files are located. Generally each component has its own dedicated log file.

File or directory	Purpose
access.log	Access log (IP and which pages were accessed)
agent/	Agent installation logs
ant.log	Log for the internal ant tasks
audit/	Directory holding daily auditing log files (no rollback, no cleanup). With log level INFO one line per transaction. With log level DEBUG everything is logged.
backend.log	9661 connector, only logged to with DEBUG enabled
dead.log	All illegal requests are logged here
debug.log	not used
events/	not used
exist.log	exist user database
file.log	not used
license.log	
lock.log	Log showing if a lock has occurred on an configuration item (WebApp module)
memory.log	memory consumption
midas.log	default log if no other special log exists
nnm.log	not used
ovo.log	Logging for HPOM and Oracle (opc_op)
ovoadmin.log	not used
package.log	
performance.log	for support purposes
request.log	for support purposes
requests/	Directory with the request logs
results/	not used
search.log	not used
servicemix.log	if an adapter doesn't start etc check this log first
sync.log	not used
task	Dir. containing individual log files written during the execution of tasks (commands,downloads)
task.log	general log whether a task has been run
threadinfo.log	for support purposes
usermgmt.log	User log when AD or LDAP is used for user
vcs.log	not used
velocity.log	used internally

File or directory	Purpose
web.log	WebApp log
wrapper.log	log for wrapper module
xmldb.log	XML DB database log

The most important log files in regard of troubleshooting are:

- wrapper.log
- servicemix.log
- midas.log

wrapper.log and servicemix.log belong to the two core components required to successfully run the application. If an error shows up here there will generally be a problem with the correct operation of the Administration UI. Most of the time this is down to some misconfiguration or file corruption.

See chapter Troubleshooting on page 91 for more in depth information on error analysis and troubleshooting.

Default Passwords

There are a couple of default users already existing in Administration UI which can be split into two groups:

- Several default user(s) in Administration UI with varying user rights, able to access HPOM.
- User(s) for internal administration purpose only (e.g. XML DB access). These are not connected to HPOM in any way.

Below you find a list of all these users and their equivalent default password:

Module	User	Password
Web Application preconfigured users	admin	`secret`
	ompolicy_adm	`secret`
	opc_adm	`OpC_adm`
	readonly	`secret`
XMLDB	admin	`admin`
BackEnd	opc_op	defined during installation

During the installation of the Administration UI software, the installer asks which user should be used to access Oracle in read-only mode. It is suggested to use opc_op, but the user has to provide the password *or* enter another user and password.

(intentionally left blank)

3 Maintenance

Overview

Maintenance here means more or less using shell commands to administer Administration UI. This chapter will first introduce you to all the basic shell commands, helping you to operate the application.

Additionally, all those scripts will be described, that exist to perform advanced tasks. For example to rename the hostname or for port changes. Furthermore, there are separate sections for the Java GUI integration and auditing.

The sections covered are:

- Command Overview on page 16

- Administration UI Commands in Detail on page 18

For very specific advanced tasks which are not used on a daily basis, but otherwise require a lot of manual configuration work, special scripts are available. These advanced scripts are available for example for port, hostname or password changes. They are featured in section:

- Advanced Tasks on page 33

- HPOM Integration on page 41

- Audit information can be found in chapter Auditing on page 60

For troubleshooting information please refer the following chapter:

- Troubleshooting on page 91.

Command Overview

Most commands can be executed by using the `adminui` command. The following table below gives an overview on all available commands. Each sub-commands will be discussed in detail after this overview. To obtain a list of all available commands just run `./adminui` without any option or parameter:

```
#/opt/OV/OMU/adminUI/adminui {sub-command} [-option]
```

sub-command	**Explanation**
ant	- run a ant task with the built-in ant
analyze	- show configuration & analyze logfiles for common errors
backend	- show details of local backend
backup	- backup configuration (correspondent command: restore)
checksum	- generate checksum (used internally)
clean	- remove logfiles and work files (e.g. in case of corruption)
config	- show configuration of components
download	- download and save user management configuration data out of XML DB (corresponding command: upload)
groovy	- run a groovy script (used internally)
help	- show this usage
init [force]	- initialize XML DB (CAUTION: deletes old configuration!)
import	- import file into clipboard
machtypes	- update machine types data
password	- password tool
patch	- apply a fixpack (corresponding command: unpatch)
ping	- ping server
restart	- restart start the server
reload	- reload configuration from save (can be from other BackEnd)
restore	- restore configuration from backup (from same BackEnd)
servicemix	- show servicemix deployments
start	- start the server (options: -nodeamon -clean)
sub-command	**Explanation**
status	- show status of server
save	- save configuration (correspondent command: reload)
stop	- stop the server
support	- collect support information
unpatch	- remove a fixpack (corresponding command: patch)

upload - upload user management configuration data into XML DB (corresponding command: download)

version - show application version information

webassemblies - re-install all webassemblies on a WebApp. This will also re-build the midas.war file and will restart Administration UI.

webassemblies.fast - re-install all webassemblies. Midas.war is not re-build and there is also no Administration UI restart.

xmldb - XML DB administration

Administration UI Commands in Detail

In this section each sub-command is described in greater detail.

Commands which are only for internal use are not discussed here.

The following commands will be discussed:

- Self-help & Healing - “./adminui analyze” on page 19
- Displaying Server Information - “./adminui backend” on page 20
- Creating And Restoring Backups - “./adminui backup|restore” on page 21
- Cleanup, File Corruption Fix - “./adminui clean” on page 23
- Displaying Configuration - “./adminui config” on page 24
- Download User Mgmt. Config. - “./adminui download|upload” on page 25
- Install and Remove Patches - “./adminui patch|unpatch” on page 26
- Starting, Stopping and Restarting - ./adminui start|stop|restart on page 27
- Save Configuration - “./adminui save|reload” on page 28
- Displaying Server Status - “./adminui status” on page 30
- Collect Support Information - “./adminui support” on page 31
- Displaying the product version - “./adminui version” on page 32

Self-help & Healing - “./adminui analyze”

The analyze sub-command is useful in regard of two aspects. First, it lists the configuration of Administration UI. Secondly, the script checks all log files for common errors. If one of these is detected, a troubleshooting tip will be displayed. Before contacting Product Support it is useful to run this command first and check the result.

Example:

```
# ./adminui analyze
...
Configuration values
====================
  Installation:
    Version                          9.1.0 (build: 288)
    OEM Version                      cvpl
    Installation Directory           /opt/OV/OMU/adminUI/
    Installation Type                full
  Server:
    Hostname                         deli.bes-intern.com
    Platform                         unix
    Backend Identifier               deli.bes-intern.com_server
  Communication:
    JMX Port                         9660
    Server Port                      9661
    HTTP Port                        9662
    HTTPS Port                       9663
  XMLDB:
    XML DB User                      midas
  ORACLE:
    Oracle Home                      /opt/oracle/product/11.1.0.6
    Oracle Major Version             11
    Oracle Host                      deli.bes-intern.com
    Oracle Port                      1521
    Oracle SID                       openview
    Oracle User                      opc_op
  Operations Manager:
    Version                          900
    Codeset                          UTF-8
  Licenses:
    License Type                     licensed through OMU

Errors in Logfiles
==================
```

In this example no errors were found.

Displaying Server Information - “./adminui backend”

The backend sub-command can be used to display the configuration settings of the local server.

For displaying extended configuration information use the `analyze` or the `config` sub-command.

Example:

```
# ./adminui backend
...
[http://exist-db.org/ant:exist] Checking collection: xmldb:exist://
deli.bes-intern.com:9662/exist/xmlrpc/db/backends
backend:
     [xslt] Processing /opt/OV/OMU/adminUI/conf/backend_local.xml to
/opt/OV/OMU/adminUI/work/server_20090324191113.txt
     [xslt] Loading stylesheet /opt/OV/OMU/adminUI/conf/stylesheets/
server_view.xsl
     [echo]
     [echo] Server deli.bes-intern.com_server:
     [echo]   Server Identifier: deli.bes-intern.com_server
     [echo]   Hostname: deli.bes-intern.com
     [echo]   Protocol: http
     [echo]   Port: 9661
     [echo]   Secure Communication: false
     [echo]   Platform: unix
     [echo]   Install Directory: /opt/OV/OMU/adminUI/
     [echo]   Services:
     [echo]     task
     [echo]     exec
     [echo]     terminal
     [echo]     file
     [echo]     backend
     [echo]     usermgmt
     [echo]     auth
     [echo]     lock
     [echo]     ovoconfig
     [echo]     ovcert
     [echo]     opccfg
     [echo]     ovconfig
     [echo]     ovoappl
     [echo]     ovoinstall
     [echo]     ovosvc
     [echo]     ovcoda
     [echo]     net
     [echo]     notice
     [echo]
   [delete] Deleting: /opt/OV/OMU/adminUI/work/
server_20090324191113.txt

BUILD SUCCESSFUL
Total time: 1 second
```

Creating And Restoring Backups - “./adminui backup|restore”

The backup and restore sub-commands can be used to backup and restore the complete configuration on a local server. During the backup operation the data is copied into a *.zip file, which is stored in /opt/OV/OMU/adminUI/.

Important: During both activities the Administration UI must be running (otherwise the XML DB containing all the user information cannot be accessed).

Please note, that the backup you want to restore must exactly match the installed Administration UI version you are running! Also the hostname and Administration UI identifier should match (otherwise the advanced rename scripts for hostname and identifier must be run after a restore).

This means that you **cannot** restore a backup created under 4.0.0 to your existing 4.1.0 system. Also a 4.1.1 backup cannot be restored to a 4.0.x or 4.1.0 system. Furthermore, a backup from server A should not be restored to server B if the hostname does not match.

In order to save and transfer configuration data between systems with different hostnames, it is recommended to use the save and reload command.
See Save Configuration - “./adminui save|reload” on page 28

The backup includes:

- The XML DB containing Administration UI users, groups and roles.
- Path aliases.
- Tasks.
- All configuration files in /opt/OV/OMU/adminUI/conf

Backup

The following example illustrates the creation of a backup:

```
[root@deli:/opt/OV/OMU/adminUI/] ./adminui backup
[...]
backup:
[mkdir] Created dir: /opt/OV/OMU/adminUI/work/backup_20090325075955
[echo] backing up OMU Administration UI configuration to /opt/OV/OMU/
adminUI/work/backup_20090325075955
[echo] saving XML DB
[mkdir] Created dir: /opt/OV/OMU/adminUI/work/backup_20090325075955/
xmldb
[xdb:backup] Database driver already registered.
[xdb:backup] Creating backup of collection: xmldb:exist://
deli.bes-intern.com:9662/exist/xmlrpc/db
[xdb:backup] Backup directory: /opt/OV/OMU/adminUI/work/
backup_20090325075955/xmldb
[xdb:backup] writing roles.xml
[xdb:backup] writing usergroups.xml
[xdb:backup] writing users.xml
[xdb:backup] writing users.xml
intern.backup_conf:
     [echo] saving configuration from /opt/OV/OMU/adminUI/conf
```

```
    [mkdir] Created dir: /opt/OV/OMU/adminUI/work/
backup_20090325075955/conf
     [copy] Copying 136 files to /opt/OV/OMU/adminUI/work/
backup_20090325075955/conf
[...]
[zip] Building zip: /opt/OV/OMU/adminUI/backup/backup_20090325075.zip
[delete] Deleting directory /opt/OV/OMU/adminUI/work/
backup_20090325075955
[echo] backup archived in /opt/OV/OMU/adminUI/backup/
backup_20090325075.zip
BUILD SUCCESSFUL
Total time: 18 seconds
```

In the last lines the name and location of the backup zip is displayed.

Restore

To restore a backup use the `restore` sub-command. The path of the backup ZIP file created with an earlier backup has to be stated which should now be restored. As the following example shows when data restore is complete there will be an automatic stop - clean - start performed so the restored data is actually used:

Note, that the server is being restarted during the restore, therefore users should be informed about the downtime.

```
[root@deli:/opt/OV/OMU/adminUI/]./adminui restore \
backup/backup_20090325075.zip
[...]
[mkdir] Created dir: /opt/OV/OMU/adminUI/work/restore_20090325081304
[echo] restoring backup backup_20090325075955.zip
[unzip] Expanding: /opt/OV/OMU/adminUI/backup_20090325075955.zip into
/opt/OV/OMU/adminUI/work/restore_20090325081304
[echo] restoring XML DB
[...]
[xdb:restore] Restoring roles.xml
[xdb:restore] Restoring usergroups.xml
[xdb:restore] Restoring users.xml
[echo] restoring configuration to /opt/OV/OMU/adminUI/conf
[copy] Copying 136 files to /opt/OV/OMU/adminUI/conf
intern.restore_path:
[echo] restoring path aliases to /opt/OV/OMU/adminUI/data/path
[copy] Copying 67 files to /opt/OV/OMU/adminUI/data/path
intern.restore_task:
[echo] restoring tasks to /opt/OV/OMU/adminUI/data/task
[copy] Copying 93 files to /opt/OV/OMU/adminUI/data/task
[...]
     [echo] restarting server
intern.server_stop.unix:
     [echo] Stopping server
     [exec] clean:
intern.server_start.unix:
     [echo] Starting server
intern.server_start.windows:
     [echo] restore successfull
BUILD SUCCESSFUL
Total time: 44 seconds
```

Cleanup, File Corruption Fix - “./adminui clean”

The clean sub-command will remove all (!) log files below /opt/OV/OMU/adminUI/logs and also the /opt/OV/OMU/adminUI/work directory. The work directory’s function is similar to a cache. At application startup all service assemblies will be unpacked into this directory. Any following application startup will benefit from this. At the same time file corruption can sometimes occur inside the work directory at runtime. In case of starting problems it is recommended to first run the analyze followed by the clean sub-command.

Please note, that after a clean the Administration UI application is **not** automatically started.

This has to be done manually either by executing ./adminui start command.

Example:

```
[root@deli:/opt/OV/OMU/adminUI/] ./adminui clean
[...]
Buildfile: /opt/OV/OMU/adminUI/run.xml
[http://exist-db.org/ant:exist] Checking collection: xmldb:exist://
deli.bes-intern.com:9662/exist/xmlrpc/db/backends
clean:
[http://exist-db.org/ant:exist] Database driver already registered.
[http://exist-db.org/ant:exist] Checking collection: xmldb:exist://
deli.bes-intern.com:9662/exist/xmlrpc/db/backends
intern.copy_wahttp:
BUILD SUCCESSFUL
Total time: 3 seconds
```

Displaying Configuration - “./adminui config”

The `config` sub-command shows extensive information about the server settings and configuration of all installed adapters and components. It can be used to find out the settings specified during installation such as ports, hostnames etc. of all deployed components:

Example:

```
[root@deli:/opt/OV/OMU/adminUI/] ./adminui config
[...]
Configuration values
====================
  Installation:
    Version                          9.1.0 (build: 288)
    OEM Version                      cvpl
    Installation Directory           /opt/OV/OMU/adminUI/
    Installation Type                full
  Server:
    Hostname                         deli.bes-intern.com
    Platform                         unix
    Backend Identifier               deli.bes-intern.com_server
  Communication:
    JMX Port                         9660
    Server Port                      9661
    HTTP Port                        9662
    HTTPS Port                       9663
  XMLDB:
    XML DB User                      midas
  ORACLE:
    Oracle Home                      /opt/oracle/product/11.1.0.6
    Oracle Major Version             11
    Oracle Host                      deli.bes-intern.com
    Oracle Port                      1521
    Oracle SID                       openview
    Oracle User                      opc_op
  Operations Manager:
    Version                          900
    Codeset                          UTF-8
  Licenses:
    License Type                     licensed through OMU
```

Download User Mgmt. Config. - “./adminui download|upload”

With the sub-command `download` it is possible to download the Administration UI user management configuration data as defined by an index file. In this index file you define which data is downloaded or uploaded

The syntax is as follows:

- Download the user data dependent on index configuration

  ```
  # cd /opt/OV/OMU/adminUI/
  #./adminui download <indexfile> <targetdirectory>
  ```

- Upload user data dependent on directory

  ```
  # cd /opt/OV/OMU/adminUI/
  #./adminui upload [add|modify]<directory>
  ```

Subentities are not implemented (yet), but this should only affect user roles.

The index file needs to be in this format:

```
<index product=”Administration UI” version=”9.1.0”>
<!-- users to download -->
<um:userref>
<um:name>opc_adm</um:name>
</um:userref>
<!-- groups to download -->
<um:usergroupref>
<um:name>administrators</um:name>
</um:usergroupref>
<!-- roles to download -->
<um:roleref>
<um:name>administrator_role</um:name>
</um:roleref>
</index>
```

Install and Remove Patches - “./adminui patch|unpatch”

Use the `patch` sub-command to install future Administration UI fixpacks (=servicepack/patches). You will receive them in a *.zip format.

To install such a fixpack file copy the ZIP file either into the installation directory `/opt/OV/OMU/adminUI/` or any other location (e.g. `/tmp`). **Do not unzip it!** This will be done by the patch mechanism itself. To install a fixpack execute

```
# /opt/OV/OMU/adminUI/adminui patch /tmp/<fixpack>.zip
```

or if the fixpack zip file is located inside `/opt/OV/OMU/adminUI/` it is sufficient to use:

```
# /opt/OV/OMU/adminUI/adminui patch <fixpack>.zip
```

In order to offer a roll back mechanism the existing configuration files will be backup to a new folder inside `/opt/OV/OMU/adminUI/<old-version>`. E.g. `/opt/OV/OMU/adminUI/9.1.0` *before* the actual patch is applied.

The `unpatch` sub-command removes a previously installed fixpack. You must specify the ID of the fixpack to be de-installed as shown in the following example:

```
# /opt/OV/OMU/adminUI/adminui unpatch 4.1.0
```

The ID is defined by the directory name in which the backup files are located in.

TIP: please run the ./`adminui` backup command in order to create a backup for the newly patched latest version. This sis because backups for older versions cannot be used for a restore.
See: Creating And Restoring Backups - “./adminui backup|restore” on page 21

Please restart manually the Administration UI software by:

```
# /opt/OV/OMU/adminUI/adminui clean
# /opt/OV/OMU/adminUI/adminui start
```

Note that the server needs a restart after a patch or unpatch process, therefore users should be informed about the downtime.

Starting, Stopping and Restarting - ./adminui start|stop|restart

Start, stop and restart is controlled by the following sub-command:

```
# /opt/OV/OMU/adminUI/adminui start|stop|restart
```

The `stop` sub-command stops the complete application on the local system:

```
# /opt/OV/OMU/adminUI/adminui stop
```

The `start` sub-command starts the application on the local system:

```
# /opt/OV/OMU/adminUI/adminui start
```

The restart sub-command will stop and restart the application:

```
# /opt/OV/OMU/adminUI/adminui restart
Restarting in 30 seconds ...
```

IMPORTANT: The start command returns to the shell prompt immediately, however Administration UI is still starting up. Depending on the speed of the local system, the start-up may take up to a few minutes.

To monitor the startup you can use the commands below:

TIP: To check the start-up progress, use

```
# tail -f /opt/OV/OMU/adminUI/logs/wrapper.log
# tail -f /opt/OV/OMU/adminUI/logs/servicemix.log
```

Logging starts in wrapper.log, then servicemix.log.

By rule of thumb if no more logging takes place inside the `wrapper.log` and `servicemix.log` the application startup is complete.

Save Configuration - “./adminui save|reload”

The sub-command save and reload is a stripped down version of the backup|reload sub-command. In contrast to the backup|restore sub-command configuration data from /opt/OV/OMU/adminUI/conf will **not** be saved.

The backup from /opt/OV/OMU/adminUI/adminui save will include:

- XML DB with users, user groups, user roles.
- Path alias.
- Tasks.

Therefore, it is possible to use this command to save the above-mentioned configuration on server A and reload the data on server B.

The backup will be placed inside /opt/OV/OMU/adminUI/save_<datestamp>.zip

Important: During both activities the Administration UI must be running (otherwise the XML DB cannot be accessed).

Example:

```
[root@deli:/opt/OV/OMU/adminUI/] ./adminui save
[...]
intern.backup_xmldb.usermgmt:
[...]
[xdb:backup] writing roles.xml
[xdb:backup] writing usergroups.xml
[xdb:backup] writing users.xml
[...]
intern.backup_path:
     [echo] saving path aliases from /opt/OV/OMU/adminUI/data/path
    [mkdir] Created dir: /opt/OV/OMU/adminUI/work/
backup_20090325143306/data/path
     [copy] Copying 67 files to /opt/OV/OMU/adminUI/work/
backup_20090325143306/data/path
intern.backup_task:
     [echo] saving tasks from /opt/OV/OMU/adminUI/data/task
    [mkdir] Created dir: /opt/OV/OMU/adminUI/work/
backup_20090325143306/data/task
     [copy] Copying 93 files to /opt/OV/OMU/adminUI/work/
backup_20090325143306/data/task
[...]
intern.backup_zip:
  [zip] Building zip: /opt/OV/OMU/adminUI/save_20090325143306.zip
[...]
  [echo] save archived in /opt/OV/OMU/adminUI/save_20090325143306.zip
BUILD SUCCESSFUL
Total time: 5 seconds
```

In the last lines you find the the filename and its location.

Reload

To reload this data back into the Administration UI use the sub-command `reload`:

```
# /opt/OV/OMU/adminUI/adminui reload save_<timestamp>.zip
```

As you can see in the output example below, the sub command `reload` will restart Administration UI (performing a stop-> clean -> start) at the end of the operation.

> ‼ Note that the server is being restarted during the reload, therefore users should be informed about the downtime.

```
[root@deli]./adminui reload save_20090325143306.zip
[...]
[xdb:restore] Restoring roles.xml
[xdb:restore] Restoring usergroups.xml
[xdb:restore] Restoring users.xml
intern.restore_path:
     [echo] restoring path aliases to /opt/OV/OMU/adminUI/data/path
     [copy] Copying 67 files to /opt/OV/OMU/adminUI/data/path
intern.restore_task:
     [echo] restoring tasks to /opt/OV/OMU/adminUI/data/task
     [copy] Copying 93 files to /opt/OV/OMU/adminUI/data/task
[...]
intern.server_stop.unix:
     [echo] Stopping server
[...]
     [exec] clean:
     [exec] [http://exist-db.org/ant:exist] Database driver already
registered.
     [exec] [http://exist-db.org/ant:exist] Checking collection:
xmldb:exist://deli.bes-intern.com:9662/exist/xmlrpc/db/backends
     [exec] intern.copy_wahttp:
     [exec] BUILD SUCCESSFUL
     [exec] Total time: 3 seconds
[...]
intern.server_start.unix:
     [echo] Starting server
BUILD SUCCESSFUL
Total time: 1 minute 53 seconds
```

Displaying Server Status - “./adminui status”

The sub-command `status` displays the status of the processes including connection status and whether the server assembly is correct or not.

Example (extract):

```
# ./adminui status
[...]
intern.status_service:
     [echo] sending status request to service ovconfig...
     [copy] Copying 1 file to /opt/OV/OMU/adminUI/work/20090325104339
  [reqpost] sending request to http://deli.bes-intern.com:9661/
[...]
     [echo] status of backend deli.bes-intern.com_server
     [echo]
     [echo]     Server:  deli.bes-intern.com_server
     [echo]     Service: auth
     [echo]       Name:        User Authentication Filter
     [echo]       Status:      connected
     [echo]       Error count: 0
     [echo]       Status:      unlicensed
     [echo]
     [echo]     Server:  deli.bes-intern.com_server
     [echo]     Service: backend
     [echo]       Name:        Local Backend Server
     [echo]       Status:      connected
     [echo]       Error count:
     [echo]       Status:      unlicensed
[...]
BUILD SUCCESSFUL
Total time: 25 seconds
```

If the application isn’t correctly running a BUILD FAILED message will be received.

Collect Support Information - “./adminui support”

In case of technical problems HP Support will require detailed information about the individual configuration of Administration UI.

With the `support` sub-command it is possible to quickly collect all required log & configuration files.

```
# /opt/OV/OMU/adminUI/adminui support
```

The shell output should look like this:

```
[root@deli:/opt/OV/OMU/adminUI/] ./adminui support
[...]
support.zip:
     [echo] collecting support information ...
     [echo] collecting version info ...
     [echo] collecting installed files ...
     [echo] collecting Java properties ...
[propertyfile] Creating new property file:
[...]
intern.checksum_check:
     [echo] checking checksums ...
     [echo] creating support zip ...
      [zip] Building zip: /opt/OV/OMU/adminUI/
support_20090325162209.zip
[echo] cleaning up ...
[echo] send the file /opt/OV/OMU/adminUI/support_20090325162209.zip
to support
BUILD SUCCESSFUL
Total time: 2 minutes 30 seconds
```

At the end of the output you see the support filename’s name and location.

Displaying the product version - “./adminui version”

Via the sub-command `version` you can display the version & build.

```
# /opt/OV/OMU/adminUI/adminui version
```

Example:

```
[root@deli:/opt/OV/OMU/adminUI/] ./adminui version
[...]
[http://exist-db.org/ant:exist] Checking collection: xmldb:exist://
deli.bes-intern.com:9662/exist/xmlrpc/db/backends
version:
     [echo] installed product = HP Operations Manager for Unix
Administration UI (OMU Administration UI)
     [echo] OMU Administration UI version = 9.1.0
     [echo] OMU Administration UI build number = 288
     [echo] OMU Administration UI build date = 20090317
     [echo] OMU Administration UI install date = 3/17/09 12:07 PM
     [echo] OMU Administration UI installation directory = /opt/OV/
OMU/adminUI/
     [echo] installed OMU Administration UI products:
     [echo]   OMU Administration UI Documentor Backend
     [echo]   OMU Administration UI Configurator Backend
     [echo]   OMU Administration UI Light Web Application
BUILD SUCCESSFUL
Total time: 1 second
```

Advanced Tasks

For very specific advanced tasks which are not used on a daily basis, but otherwise require a lot of manual configuration work, special scripts are available.

These advanced scripts are available for example for port, hostname or password changes.

This section will cover the following advanced tasks:

- Machtypes Update on page 34
- Update of opc_op Password on page 34
- Importing HPOM Download Data on page 34
- Renaming the BackEnd Identifier on page 35
- Changing the Hostname on page 35
- Changing the BackEnd Port (9661) on page 35
- Changing the WebApp’s HTTP or/and HTTPS Port(s) on page 36
- Disabling the WebApp’s HTTP Port (9662) on page 37
- JMX Port Change on page 38
- Resetting the Default Password for User “admin” on page 39
- Changing between HTTP and HTTPS Communication on page 39
- Re-initialization of XML DB on page 40

Machtypes Update

When new machtypes are introduced in HPOM with patches or by a new agent version, the MIDAS HPOM BackEnd module will dynamically read those. Therefore, no manual command execution is needed there.

The WebApp module though needs to be manually updated. Here it is necessary to run the command:

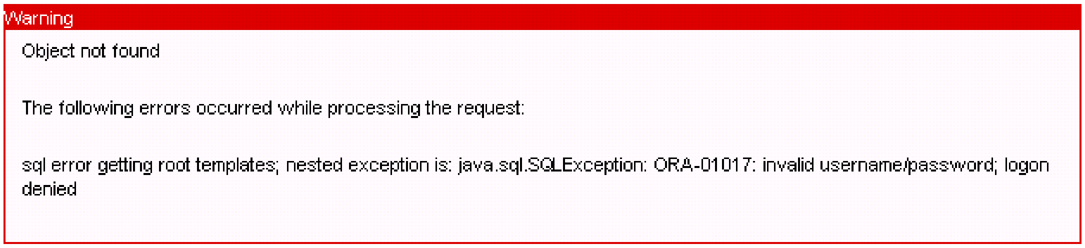
```
# /opt/OV/OMU/adminUI/adminui machtypes
```

No restart is necessary of the WebApp module, so there is no downtime.

Update of opc_op Password

Whenever the opc_op password is changed and updated it is also necessary that this password change is performed in Administration UI! Otherwise all list operations (e.g. list Policy Bank, list All Nodes) will fail and the user will be presented with an error message as below (Figure 3 on page 34):

Figure 3 Wrong opc_op Password



Warning

Object not found

The following errors occurred while processing the request:

```
sql error getting root templates; nested exception is: java.sql.SQLException: ORA-01017: invalid username/password; logon denied
```

This section explains in short how to update the opc_op password in Administration UI. The command is:

```
# /opt/OV/OMU/adminUI/adminui password -u ovodb -a -p <password>
```

After that please restart the application in the following way:

```
# /opt/OV/OMU/adminUI/adminui clean
# /opt/OV/OMU/adminUI/adminui start
```

For more in-depth details please refer to the following configuration chapters:

- Access to HPOM and Oracle on page 53

- Problems with Passwords on page 48

Importing HPOM Download Data

The `import` sub-command can be used to import a HPOM configuration download directory into the Administration UI’s Clipboard directory. From there it can be processed further using regular GUI functionality.

Example:

```
# /opt/OV/OMU/adminUI/adminui import /tmp/my_download
```

To access the Clipboard directory please use from the Administrative Menu: “Browse” -> “Downloads”.

Renaming the BackEnd Identifier

The BackEnd identifier is a central attribute of each system. The syntax is usually “hostname_server”. It is stored and maintained in multiple configuration files. If the identifier has to be changed (usually in conjunction with a rename of the hostname), please run the following command:

```
# /opt/OV/OMU/adminUI/adminui -f conf/ant/admin.xml rename_backend \
-Dbackend=<newname>
```

Please note, that the server is being restarted during the rename, therefore users should be informed about the downtime.

Changing the Hostname

The hostname is stored in multiple configuration files.

Therefore, action needs to be taken either before (preferred) or after the name change. The rename command’s syntax is:

```
# /opt/OV/OMU/adminUI/adminui ant -f conf/ant/admin.xml
rename_hostname \ -Dhost=<newhost>
```

Please note, that the server is being restarted during the rename, therefore users should be informed about the downtime.

TIP: When the rename is performed before renaming the hostname, the server is ready to be restarted when the hostname has been changed.

If the hostname was already changed it is possible that the hostname change within the Administration UI will not 100% succeed.

It is recommended to manually restart Administration UI and re-run the rename hostname script.

Changing the BackEnd Port (9661)

In case that the BackEnd port (default 9661) needs to be changed simply run the following command providing the new port.

```
# /opt/OV/OMU/adminUI/adminui ant -f conf/ant/admin.xml \
change_server_port -Dport=<new-port>
```

Restart Administration UI after this configuration change, using the following command:

```
# /opt/OV/OMU/adminUI/adminui clean
# /opt/OV/OMU/adminUI/adminui start
```

Please do not forget to inform your users about the downtime.

Changing the WebApp’s HTTP or/and HTTPS Port(s)

If it is necessary to change the HTTP (default 9662) or/and HTTPS (default 9663) port the following command is available to do this:

HTTP port change only:

```
# /opt/OV/OMU/adminUI/adminui ant -f conf/ant/admin.xml
change_web_port \ -Dport.http=<new-port>
```

HTTPS port change only:

```
# /opt/OV/OMU/adminUI/adminui ant -f conf/ant/admin.xml
change_web_port \ -Dport.https=<new-port>
```

Combined HTTP and HTTPS port change:

```
# /opt/OV/OMU/adminUI/adminui ant -f conf/ant/admin.xml
change_web_port \ -Dport.http=<new-port> -Dport.https=<new-port2>
```

Restart Administration UI with the following command:

```
# /opt/OV/OMU/adminUI/adminui clean
# /opt/OV/OMU/adminUI/adminui start
```

‼ Please do not forget to inform your users about the downtime.

Disabling the WebApp’s HTTP Port (9662)

In order to access Administration UI through a web-browser, two access options exist: For HTTP the default port is 9662 and for HTTPS it is 9663.

- for unencrypted access use `HTTP://HP-OM-Server:9662`
- for encrypted access use `HTTPS://HP-OM-Server:9663`

In order to enforce usage of HTTPS, it is possible to disable HTTP access via port 9662. This is achieved by binding port 9662 to “localhost”.

In order to implement this change the following steps have to be applied (This setup assumes, that Administration UI is up and running):

- (1) Edit the following file:

```
# /opt/OV/OMU/adminUI/conf/jetty.xml
```

At the beginning of the file search for this block:

```
<!-- default http connector -->
<bean class="org.mortbay.jetty.bio.SocketConnector">
```

After this block add this line:

```
<property name="host" value="localhost"/>
```

The block should look like this after the modification:

```
<!-- default http connector -->
<bean class="org.mortbay.jetty.bio.SocketConnector">
  <property name="host" value="localhost"/>
  <property name="port" value="9662"/>
  <property name="headerBufferSize" value="12000"/>
```

- (2) Edit the following file:

```
/opt/OV/OMU/adminUI/conf/config.properties
```

Change the hostname to localhost, so the configuration block will look like this:

```
vendor = blue elephant systems GmbH
backend = rhel-support_server
hostname = localhost
server.port = 9661
```

- 3) Edit the following file:

```
/opt/OV/OMU/adminUI/conf/usermgmt.properties
```

Change the URL so it looks like this:

```
xmldb.dbUrl=xmldb:exist://localhost:9662/exist/xmlrpc/db/
```

- (4) Restart Administration UI via the command:

```
# ovc -stop adminui
# /opt/OV/OMU/adminUI/adminui clean
# ovc -start adminui
```

Please note, that if a Administration UI patch is applied, this modification has to be re-applied.

Please do not forget to inform your users about the downtime.

JMX Port Change

The JMX port is only used locally. It is used for troubleshooting purposes and by some ANT scripts.

Two configuration files exist in which the port can be changed if necessary:

- servicemix.properties

  ```
  # vi /opt/OV/OMU/adminUI/conf/servicemix/servicemix.properties
  [...]
  rmi.port                   = 9660
  rmi.host                   = apollo
  ```

- config.properties

  ```
  # vi /opt/OV/OMU/adminUI/conf/config.properties
  [...]
  # JMX
  jmx.port = 9660
  jmx.user = ${backend.user}
  ```

Restart Administration UI after this configuration change with the following command:

```
# /opt/OV/OMU/adminUI/adminui clean
# /opt/OV/OMU/adminUI/adminui start
```

Please do not forget to inform your users about the downtime.

Resetting the Default Password for User “admin”

After the first login into the Administration UI’s GUI the user is asked to change the default password of the user “admin”. In case this password is forgotten, you can reset it to its initial value which is:

```
secret
```

The reset operation will not impact any other existing user. The command is:

```
# /opt/OV/OMU/adminUI/adminui ant -f conf/ant/admin.xml \
reset_admin_password
```

The screen output should look like this:

```
Buildfile: conf/ant/admin.xml
reset_admin_password:
     [echo] Resetting password of admin user
[xdb:extract] Extracting resource: users.xml to /opt/OV/OMU/adminUI/
work/users_20090327141005.xml
[xdb:store] Database driver already registered.
     [echo] Re-login as user admin, password secret
BUILD SUCCESSFUL
Total time: 3 seconds
```

No application restart is necessary here!

Changing between HTTP and HTTPS Communication

The communcation on port 9661 (default) between the WebApp and BackEnd is HTTP by default. If this needs to be changed to HTTPS afterwards, the endpoints and certificates need to be re-generated/created.

In order to do so please run:

```
# /opt/OV/OMU/adminUI/adminui ant -f conf/ant/admin.xml
backend_convert
```

After that please exchange and import the HTTPS certificates accordingly. See HTTPS/SSL on page 109 for details.

The same command can be used to change also from HTTPS to HTTP.

Known issue in MIDAS v4.2.0 (other versions are not impacted):

When switching between HTTP/HTTPS the search index module does not get updated. Therefore, it is necessary to additionally run the following command on the MIDAS Web Application system:

```
# /opt/OV/OMU/adminUI/adminui ant -f run.xml intern.solr_http
```

TIP: In order to check the status whether secure or unsecure HTTP communication is used, execute:

```
# /opt/OV/OMU/adminUI/adminui backend
```

Re-initialization of XML DB

The sub-command `init` can be used to re-initialize the XML DB.

Unless you are advised by Product Support, please do NOT use this command!

It is **highly recommended** to run the sub-command `./adminui backup` or `./adminui save` before executing the `init` command, in order to backup any existing user database.

Generally, it is not necessary to run this command.

Only under very special circumstances in which the XML DB wasn’t correctly initialized (right after an installation) the `init` sub-command is needed.

This command can be executed with two options:

- `/opt/OV/OMU/adminUI/adminui init`

  Here an upload and reload of only missing parts inside the XML DB will take place.

- `/opt/OV/OMU/adminUI/adminui init force`

  This will reset the XML DB database completely, **therefore all existing user, user groups and user roles will be lost!** This option has generally to be used if the XML DB setup failed after an installation. Therefore, the initial login will be again:

  — username: admin

  — password: secret.

No restart of Administration UI is necessary here.

HPOM Integration

A tight integration into HPOM OMU exists which should help all users in their daily operative work. In order to enable the HPOM integration it is necessary to configure self-monitoring.

This section covers the following two topics:

- Self-Monitoring on page 41

- HPOM Java GUI on page 44

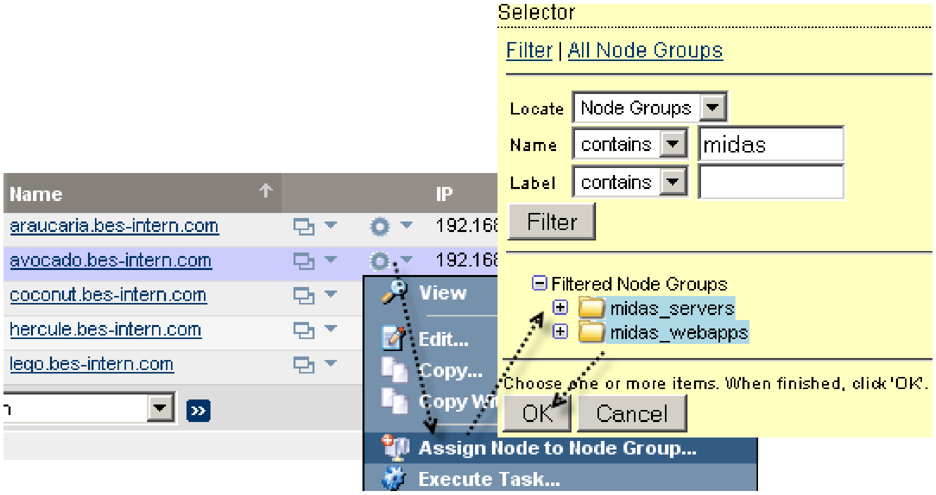
Self-Monitoring

The Administration UI comes with a set of policies, tools as well as node groups, profiles, etc., which are all intended for self-monitoring.

To use them in a HPOM environment, it is required to run through the following steps:

- Assign the Administration UI server (Example below: “deli”) into the corresponding Node Groups (midas_servers, midas_webapps), see Figure 4 on page 41:

Figure 4 Assign Administrator UI Server to Node Groups



Selector

Locate: Node Groups

Name: contains midas

Label: contains

Choose one or more items. When finished, click 'OK'.

With this node groups assignment the applicable policies will be assigned.

- After the node group assignment it is necessary to deploy those policies and scripts to the target node. This will be the HPOM management server itself.

  Run through the following procedure (see Figure 5 on page 42):

  — Go to “Deployment” and select “Deploy Configuration” (Step 1)

  — Select “Policies” and “Monitors” as components (include monitors since there are some scripts included!) (Step 2)

  — Use the “Select” button to open the selector (Step 3)

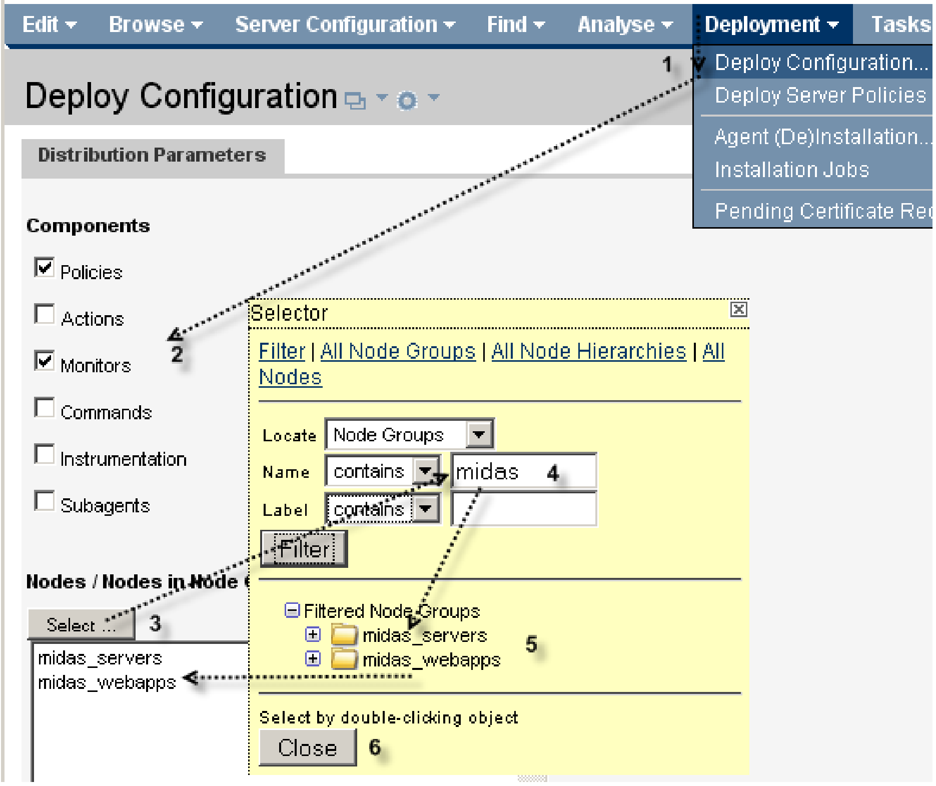
  — From the drop down menu “Locate” select “Node Groups” and enter as part of the name “midas” (Step 4)

  — Double click both entries “midas_servers” and “midas_webapps” so these two are moved to the selection window (Step 5)

  — Close the selector (Step 6)

  — Press the “Distribute” button on the right hand side.

Figure 5 Deploy Configuration



Deploy Configuration

Distribution Parameters

Components

☑ Policies

☐ Actions

☑ Monitors

☐ Commands

☐ Instrumentation

☐ Subagents

Nodes / Nodes in Node (

Select ...

midas_servers
midas_webapps

Selector

Filter | All Node Groups | All Node Hierarchies | All Nodes

Locate: Node Groups

Name: contains midas

Label: contains

Filter

Filtered Node Groups
midas_servers
midas_webapps

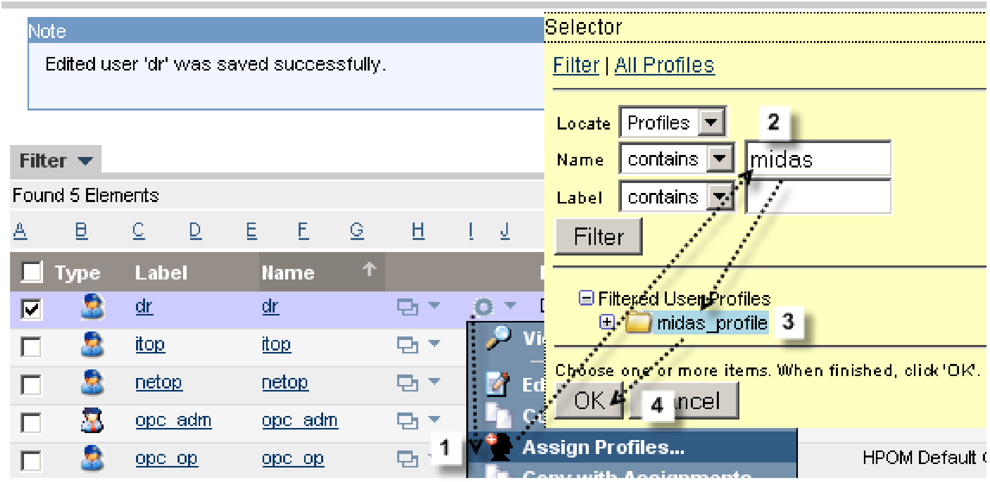
Select by double-clicking object

Close

- In a last step make the alarm messages available to all HPOM operators. To do so assign the profile with the name “midas_profile” to your users (see Figure 6 on page 43). The steps are:
  - — For an user select “Assign Profiles” (Step 1)
  - — Enter in the field “name”: “midas” (Step 2). This will list inside the filtered results section:“midas_profile”
  - — Select “midas_profile” so it gets highlighted (Step 3)
  - — Finish operation with “OK” (Step 4)

No other steps are necessary. Re-load the HPOM Java GUI sessions.

Figure 6 Assign Profile to Users



Note

Edited user 'dr' was saved successfully.

Filter ▼

Found 5 Elements

A B C D E F G H I J

□ Type	Label	Name ↑
☑	dr	dr
□	itop	itop
□	netop	netop
□	opc_adm	opc_adm
□	opc_op	opc_op

Selector

Filter | All Profiles

Locate: Profiles

Name: contains midas

Label: contains

Filter

⊟ Filtered User Profiles
⊞ midas_profile

Choose one or more items. When finished, click 'OK'.

OK Cancel

In case any of the initial self-monitoring policies, scripts or tools are deleted there is an easy solution to restore them. All configuration data for the self-monitoring setup and the HPOM Java GUI integration are stored inside:

```
/opt/OV/OMU/adminUI/data/init/ovo/selfmon
```

The opccfgupld command can be used to upload this data, for example via:

```
# cd /opt/OV/OMU/adminUI/
# opccfgupld -add $PWD/data/init/ovo/selfmon
```

HPOM Java GUI

In addition to the classic HPOM SMART-Plug-in capabilities like starting, stopping the server status or launching a GUI, Administration UI provides a set of HPOM tools, which can be used most efficiently in the HPOM Java GUI. They all launch the GUI as a HTML page inside the Java GUI and are able to directly jump to the desired context, e.g. to the policy condition, which has caused a message or the node where the message originated.

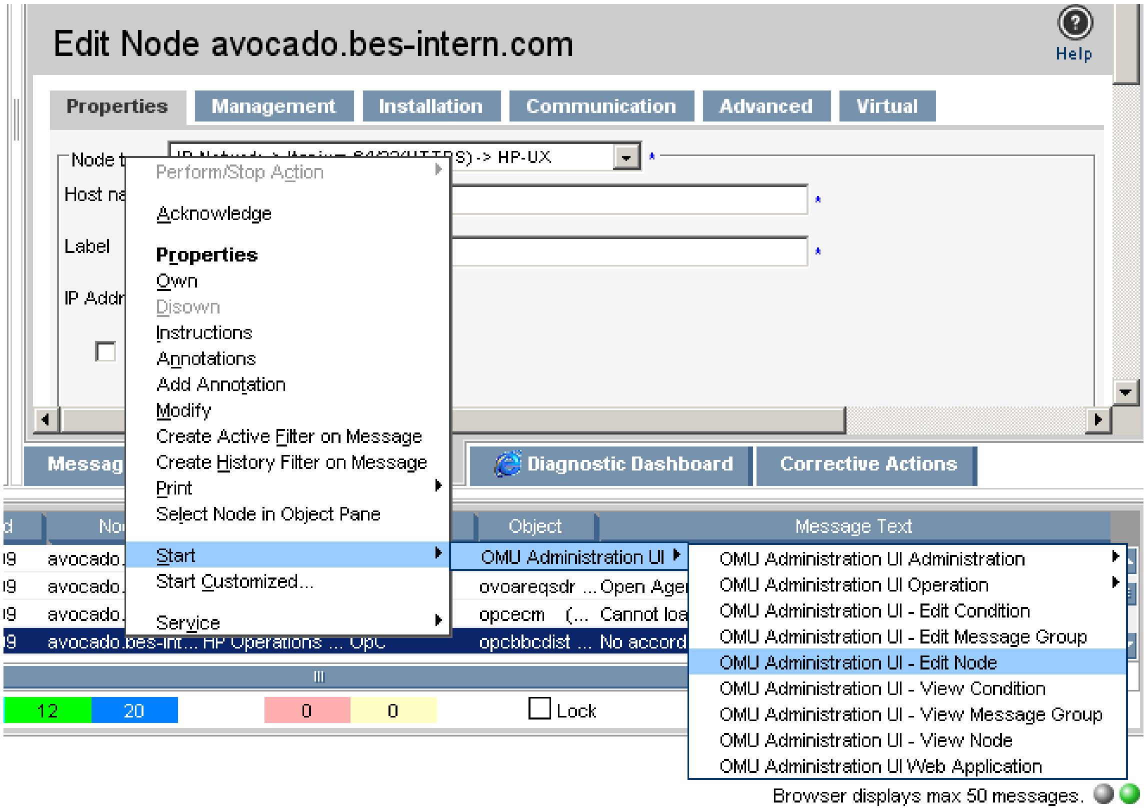
Previous versions of the Java GUI the embedded browser caused some problems. Therefore, make sure you are using version 9.0 of the Java GUI!

In order to use this feature, please make sure that the HPOM users have the profile “midas_profile” assigned (see previous section).

The Administration UI functions are available via the context menu in the Java GUI: “Start” -> “OMU Administration UI” (Figure 7 on page 44)..

!! First time you use any of these functions during your session, you will be asked to provide your Administration UI username and password.

Figure 7 Java GUI Integration



Edit Node avocado.bes-intern.com
Help
Properties
Management
Installation
Communication
Advanced
Virtual
Node t
Host na
Label
IP Addr
HP-UX
Perform/Stop Action
Acknowledge
Properties
Own
Disown
Instructions
Annotations
Add Annotation
Modify
Create Active Filter on Message
Create History Filter on Message
Print
Select Node in Object Pane
Start
Start Customized...
Service
OMU Administration UI
OMU Administration UI Administration
OMU Administration UI Operation
OMU Administration UI - Edit Condition
OMU Administration UI - Edit Message Group
OMU Administration UI - Edit Node
OMU Administration UI - View Condition
OMU Administration UI - View Message Group
OMU Administration UI - View Node
OMU Administration UI Web Application
Messag
Diagnostic Dashboard
Corrective Actions
Object
Message Text
avocado
ovoareqsdr ...Open Ager
opcecm (... Cannot loa
avocado.bes-int... HP Operations ... OpC
opcbbcdist ... No accord
12
20
0
0
□ Lock
Browser displays max 50 messages.

4 Configuring

Overview

This chapter contains information for modifying the Administration UI environment.

During the installation, all relevant setup parameters will be defined and stored in the various Administration UI configuration files.

However, if the HPOM environment is changing after the initial installation of Administration UI, you will have to make sure that the changes do not have an adverse impact on Administration UI. For example, if you change the password for the HPOM administrator, opc_op, you will have to update Administration UI, as well.

The information in this chapter covers the following topics:

- Changing Passwords on page 46
- Access to HPOM and Oracle on page 53
- Changing Passwords on page 46
- Problems with Passwords on page 48
- Oracle 11 Password Aging on page 51
- Log Files and Trace Levels on page 56
- Auditing on page 60
- Request Logging Mechanism on page 63
- High Availability Environments on page 64
- Advanced Communication Options on page 64
- Tuning Java Parameters on page 72
- Web Interface Timeout on page 74

Changing Passwords

Administration UI comes with a couple of default users (and their equivalent default passwords), but also some modules have either a default password set or it is defined during the installation.

This section explains how these passwords can be changed using Administration UI command line.

Please note, that users inside the XMLDB which are used to login into the Administration UI web application GUI are not covered here. In order to reset the default user “admin” which is used to access the WebApp GUI please refer to the following section: Resetting the Default Password for User “admin” on page 39.

Default Passwords

See section Default Passwords on page 13 for a complete listing.

Password Tool

For security reasons, all passwords stored in Administration UI are encrypted. To change a password (e.g. that of the Oracle user opc_op), please use the sub-command `password`: `/opt/OV/OMU/adminUI/adminui password <options>`

The full syntax is (examples can be found below):

```
# /opt/OV/OMU/adminUI/adminui password -u <useralias> [-a][-c][-p
<password>]
```

The parameters are as follows:

- `-u <useralias>` It is mandatory to specify a user alias name to select which password (Oracle DB user opc_op, etc.) is to be encrypted and updated. All useralias definitions can be found below.
- `-a` Using -a will **automatically update** the password in the corresponding configuration files, so no manual update is needed.
- `-c` This will perform a **password check**. This will verify the entered password against the password used in the different Administration UI configuration files.
  If they match you will receive a message “Password is same for user xxxx”. This feature is useful if you are unsure if the password has been update in Administration UI or not.
- `-p <password>` Provide the **new password.**

Please note, that the -a and -c command **CANNOT** used be together since it does not make sense to check and update a password at the same time.

In order to receive a complete useralias listing you can also simply enter without using any additional parameters:

```
# /opt/OV/OMU/adminUI/adminui password
```

This command will simply display a list of all existing useraliases.

Currently for HPOM 9 the following users exist:

Alias	Description
ovodb	HPOM database user password (opc_op or opc_rep)
xmldb	XML database administration user

NOTE: If you encounter a startup problem in Administration UI, after a password change inside HPOM but without changing it in Administration UI, the following restart and update procedure is recommended:

1) `# /opt/OV/OMU/adminUI/adminui clean`

2) Change the password

3) `# /opt/OV/OMU/adminUI/adminui start`

Example:

- Update the Administration UI configuration with a new opc_op password without updating the Administration UI configuration files manually but automatically by executing using the command line:

  ```
  # /opt/OV/OMU/adminUI/adminui password -u ovodb -a -p <password>
  ```

It is common to forget to update the Administration UI configuration after changing a HPOM password. If the Administration UI server does not start but Oracle and HPOM are running fine, check the password settings first. For more information about typical error codes for problems relating to incorrect passwords, see Problems with Passwords on page 48 (below).

Problems with Passwords

If Administration UI does not start or is not working correctly, a common problem is, that the HPOM access parameters (particularly passwords) are incorrect. During the installation of Administration UI it is required to provide a user and its password, which can be used for read-only access into Oracle. If this password is incorrect or alternatively, the password of this user is changed in Oracle at some point after the installation without updating it also inside Administration UI all OMU object class listings will fail.

If you need to find out how to solve problems related to missing or incorrect passwords, have a look at the information in the following sections:

- Testing an Oracle Password on page 48

- Resetting an Oracle Password on page 50

- HPOM Administrator Access on page 50

- Identifying Password Errors on page 51

- Oracle 11 Password Aging on page 51

Testing an Oracle Password

If the required password is unknown, either ask someone who knows it or if you have some ideas what it could be, first test it using a standalone command.

Determine the related Oracle parameters, for example by running:

```
# cat /etc/opt/OV/share/conf/ovdbconf
DB_VENDOR Oracle
DB_NAME openview
DB_RELEASE 10.1.0
DB_TIME_STAMP "Tue Aug 1 14:50:20 METDST 2006"
DB_USER ovdb
ORACLE_SID openview
ORACLE_HOME /opt/oracle/product/10.1.0
ORACLE_BASE /opt/oracle
DBA_USER oracle
DATA_DIR /opt/u01/oradata/openview
CREATE_DIR /opt/oracle/admin/openview/create
INDEX_DIR /opt/u01/oradata/openview
ADMIN_DIR /opt/oracle
OS_AUTHENT_PREFIX
CHARACTER_SET WE8ISO8859P15
BASE_DATA_TS_SIZE 25
BASE_INDEX_TS_SIZE 5
DATA_TS_SIZE 25
INDEX_TS_SIZE
TEMP_TS_SIZE 2
DATA_TS_EXTENT_SIZE 2
DATA_TS_MAX_SIZE 500
INDEX_TS_EXTENT_SIZE
ECHO_CMD echo
PROMPT TRUE
DBA_PROGRAM sqlplus
```

```
OV_USER ovdb
DBA_LOGFILE /var/opt/OV/share/log/sqlplus_log
ORACLE_BASE_REV 10
ORACLE_SECOND_REV 1
NLS_LANG american_america.WE8ISO8859P15
ITO_DATADIR /opt/u01/oradata/openview
ITO_INDEXDIR /opt/u01/oradata/openview
SQLNET_ALIAS ov_net
```

Switch to the user `oracle` and try to connect to the database using the user password combination you want to test. If the HPOM database instance runs under a different user account, you will have to check the `ovdbconf` file for the appropriate values for the user name and password. The example above shows an excerpt from an ovdbconf file.

```
# su – oracle -c sqlplus
SQL*Plus: Release 11.1.0.7.0 - Production on Wed Apr 1 09:55:17 2009
Copyright (c) 1982, 2008, Oracle.  All rights reserved.
Enter user-name: opc_op
Enter password:
Connected to:
Oracle Database 11g Enterprise Edition Release 11.1.0.7.0 - 64bit
Production
SQL> exit
```

If the sqlplus command displays the response Connected, the user name and password you tested are correct; if not, you will have to try again.

This should work with all Oracle versions.

If you found out the password this way, update the Administration UI configuration as described above (Password Encryption).

CAUTION: DO NOT USE the following command to test the login (or similar):

```
Enter user-name: opc_op as sysdba
```

Here you can enter any password even a wrong one and you can still login, so this is no real check.

Resetting an Oracle Password

If it is not possible to find out the Oracle password, you will have to change it.

CAUTION: You should change the Oracle password only if absolutely necessary. For more information about changing the password, see the opcdbpwd man page.

To change the Oracle password for the OpenView database, use the HPOM command opcdbpwd as illustrated in the following example:

Please backup this file first:

```
/etc/opt/OV/share/conf/OpC/mgmt_sv/.opcdbpwd.sec
```

Then change the opc_op password in HPOM via:

```
# /opt/OV/bin/OpC/opcdbpwd -set
```

You must use the opcdbpwd command to change the Oracle password for the OpenView database (rather than the Oracle SQL command alter). The opcdbpwd command also updates HPOM's internal security file opcdbpwd.sec with the new authentication, which is essential for HPOM to continue to work properly after the password change. If you use the opcdbpwd command to change the Oracle password, make sure you also update the Administration UI configuration as described in the section Password Encryption.

The Oracle opc_report password cannot be changed using opcdbpwd, instead use the appropriate sqlplus commands as shown in the following example:

```
SQL> alter user opc_report identified by <new password>;
SQL> commit;
```

HPOM Administrator Access

From Administration UI version 9.1.0 onwards there is no login/password configuration necessary for the opc_adm user as it was the case with MIDAS / Configuration Value Pack (CVP) 3.1.

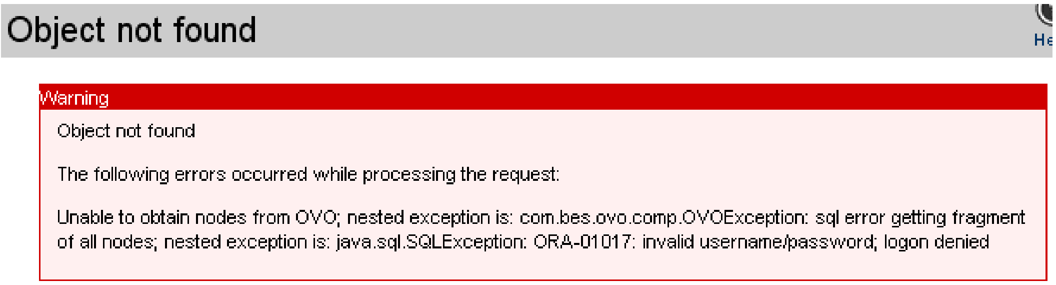
Identifying Password Errors

In case the opc_op password is incorrect, Administration UI will startup, but viewing any HPOM object will result in an error message. HPOM and Oracle related password problems can be found inside

```
/opt/OV/OMU/adminUI/logs/ovo.log
```

If the password of the Oracle DB user opc_op is wrong, you will receive the following error when trying to access HPOM inside the Administration UI webinterface, e.g. when requesting to list all nodes or policies

Figure 8 Oracle Connection Password Error



Object not found

Warning

Object not found

The following errors occurred while processing the request:

Unable to obtain nodes from OVO; nested exception is: com.bes.ovo.comp.OVOException: sql error getting fragment of all nodes; nested exception is: java.sql.SQLException: ORA-01017: invalid username/password; logon denied

This error will show inside the `/opt/OV/OMU/adminUI/logs/ovo.log` like this:

```
ERROR - 2008-01-14 21:21:06,768 | OVODBServer.getConnection(240) |
Failed to get connection from pool ovoconfig java.sql.SQLException:
ORA-01017: invalid username/password; logon denied.
```

Please note, that while starting up, Administration UI will not try and connect to the Oracle database. Only when a user actually requests via the WebApp interface a listing of nodes, policies, etc., the connection will be established. Therefore, this error can only be seen after a user has actually tried to view some HPOM items via Administration UI.

Oracle 11 Password Aging

Oracle 11g uses by default password aging. Passwords (for example for the opc_op user) will expire after 6 months!

If the password of the Oracle user opc_op expires, the HPOM processes & MIDAS Configurator are no longer able to connect to the database.

HPOM 8 with Oracle 10.2.0 doesn't have an expiry date.

Problem

The HPOM 9 Management Server runs for six months, then fails to start and fails to connect to the Oracle database.

In the `/var/opt/OV/log/System.txt` file you will see the following error:

```
0: ERR: Tue Oct 27 10:53:25 2009: opcmsgm (13301/1):
[chk_sqlcode.scp:95]: Database: ORA-
28001: the password has expired
(OpC50-15)
Could not connect to database ov_net.
```

Solution

Two solutions exist:

- Define a new password every 6 months.
- Disable the password aging mechanism.

Define new Password

1. To change the password in the Oracle database and in HPOM, call opcdbpwd:

```
# /opt/OV/bin/OpC/opcdbpwd -set
```

2. Update the AdminGUI with the new password:

```
# /opt/OV/OMU/adminUI/adminui clean
# /opt/OV/OMU/adminUI/adminui password -u ovodb -a -p <password>
```

3. Next, you should be able to start the processes again. To make sure processes that were started before changing the password use the new password, re-start the HPOM server processes:

```
# ovc -kill
# ovc –start
```

Disable Password agin Mechanism in Oracle

If you choose not to have to change the password every six months, as is the new default behaviour with Oracle 11g, then you can disable password aging. To do so:

```
# su - oracle
$ sqlplus /nolog
SQL> connect / as sysdba
SQL> ALTER PROFILE default LIMIT PASSWORD_LIFE_TIME UNLIMITED;
```

You can verify the status as follows:

```
SQL> select USERNAME, ACCOUNT_STATUS, LOCK_DATE, EXPIRY_DATE, CREATED
SQL> from dba_users;

USERNAME ACCOUNT_STATUS LOCK_DATE EXPIRY_DATE CREATED
------------------------------ ----------------------
OUTLN OPEN 19-APR-10 21-OCT-09
OPC_REPORT OPEN 19-APR-10 21-OCT-09
OPC_OP OPEN 19-APR-10 21-OCT-09
SYS OPEN 19-APR-10 21-OCT-09
SYSTEM OPEN 19-APR-10 21-OCT-09
DBSNMP EXPIRED & LOCKED 21-OCT-09 21-OCT-09
TSMSYS EXPIRED & LOCKED 21-OCT-09 21-OCT-09
DIP EXPIRED & LOCKED 21-OCT-09 21-OCT-09
ORACLE_OCM EXPIRED & LOCKED 21-OCT-09 21-OCT-09
```

Access to HPOM and Oracle

The information in this section helps you to set up access to both HPOM and to the Oracle database. The information covers the following areas in greater detail:

- Oracle Connectivity on page 53

For all password related questions please refer to:

- Changing Passwords on page 46

Oracle Connectivity

If any of the existing HPOM Oracle database settings is changed it is also important to update the corresponding configuration entries inside Administration UI. Otherwise, Administration UI will not be able to connect to Oracle and any OM object class listing request will fail (for example list policy bank).

The following Oracle changes will also impact Administration UI are:

- Hostname change of a remote Oracle server.
- Change of HPOM Oracle database port.
- Change of HPOM Oracle database SID.
- Communication of Oracle is changed: secure (oci) ./. unsecure (thin).

If any of these configuration parameters is changed, it is necessary also to update Administration UI.

All Oracle related information is stored in three configuration files which are:

```
# /opt/OV/OMU/adminUI/conf/ovoappl.properties
# /opt/OV/OMU/adminUI/conf/ovoconfig.properties
# /opt/OV/OMU/adminUI/conf/ovoinstall.properties
```

Each of these three properties files contain an URL, which defines the relevant Oracle connectivity information. It could look like this for example:

```
ovodb.url=jdbc:oracle:thin:@avocado.hp.com:1521:openview
```

The syntax and fields which could require modification are:

```
ovodb.url=jdbc:oracle:<thin|oci>@<Oracle_host>:<port>:<SID>
```

Most items should be self-explanatory:

`<Oracle_host>` - Oracle server hostname which hosts the OMU database.

`<port>` - Oracle port.

`<SID>` - HPOM Oracle database instance name (SID).

`<thin|oci>` - Oracle can operate using unencrypted (thin) or encrypted communication (oci).

For more information regarding Oracle JDBC as used by Administration UI please refer to `http://www.oracle.com/technology/tech/java/sqlj_jdbc/htdocs/jdbc_faq.htm`.

Configuration Check

The type of communication used and the Oracle configuration settings can be checked manually by using one of the following commands. $ORACLE_HOME needs to be correct:

```
$ORACLE_HOME/bin/tnsping <oracle_server>
```

The output could look like this:

```
[...](DESCRIPTION=(CONNECT_DATA=(SERVICE_NAME=avocado.bes-intern.com)
)(ADDRESS=(PROTOCOL=TCP)(HOST=192.168.123.123)(PORT=1521)))
```

or

```
$ORACLE_HOME/bin/lsnrctl status
```

The output of the latter command could look like this:

```
[...] Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=openview)))
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=avocado.bes-intern.com)
(PORT=1521)))
Services Summary...
 [...]
```

(PROTOCOL=TCP) stands for unencrypted communication.

(PROTOCOL=ICP) indicates that you need to enable “thin” inside the mentioned configuration files of Administration UI.

If ICP and TCP are both available we recommend to use an unsecure Oracle connection, as it is the case in the last example.

If only secure Oracle communication is allowed but normal communication via TCP/IP should be added, this can be achieved by modifying the `listener.ora` file on the target system like this:

BEFORE

```
===
LISTENER =
  (ADDRESS_LIST =
        (ADDRESS=
          (PROTOCOL=IPC)
          (KEY= openview)
        )
  )
STARTUP_WAIT_TIME_LISTENER = 0
CONNECT_TIMEOUT_LISTENER = 10
LOG_DIRECTORY_LISTENER = /appl/ora/product/10.1.0/network/log
LOG_FILE_LISTENER = listener
SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME=openview)
      (ORACLE_HOME=/appl/ora/product/10.1.0)
    )

)
```

AFTER

```
LISTENER =
  (ADDRESS_LIST =
        (ADDRESS=
          (PROTOCOL=IPC)
          (KEY= openview)
        )
                (ADDRESS =
                  (PROTOCOL = TCP)
                  (HOST = abcdefg1)
                  (PORT = 1521)
                )
  )

STARTUP_WAIT_TIME_LISTENER = 0
CONNECT_TIMEOUT_LISTENER = 10
LOG_DIRECTORY_LISTENER = /appl/ora/product/10.1.0/network/log
LOG_FILE_LISTENER = listener
SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME=openview)
      (ORACLE_HOME=/appl/ora/product/10.1.0)
    )
  )
```

Log Files and Trace Levels

Logging stands for all activities where Administration UI writes some status information into a dedicated log file. Normally these are errors only, but for troubleshooting scenarios, also tracing data can be obtained by this way. The information in this section covers the following logging areas:

- Log4j Logs on page 57

See also

- Auditing on page 60

- Request Logging Mechanism on page 63

Log4j Logs

Both logging and tracing are covered by the same logging mechanism log4j (for details see http://logging.apache.org).

When changing log levels NO application restart is necessary. The log4j.xml file is read every 60 sec.

General logging for the Administration UI Server is configured in the following file (also containing more explanations about the syntax, meaning of values etc.):

```
# cat /opt/OV/OMU/adminUI/conf/log4j.xml
[...]
Log4J Configuration Quick Reference:
====================================
Priority order is DEBUG < INFO < WARN < ERROR < FATAL
PatternLayout conversion characters:
%c   Category of the logging event
%C   Fully qualified class name of the caller
[...]
  <!-- MIDAS adaptor default log file -->
  <appender name="midas"
class="org.apache.log4j.RollingFileAppender">
    <param name="File" value="logs/midas.log"/>
    <param name="MaxFileSize" value="1MB"/>
    <param name="MaxBackupIndex" value="10"/>
    <layout class="org.apache.log4j.PatternLayout">
      <param name="ConversionPattern" value="%p - %d | %C{1}.%M(%L) |
%m%n"/>
    </layout>
  </appender>
[...]
<!-- MIDAS adaptor specific log files -->
<appender name="ovo" class="org.apache.log4j.RollingFileAppender">
[...]
<!-- apache stuff -->
<logger name="org.apache.servicemix" additivity="false">
    <level value="INFO"/>
    <appender-ref ref="servicemix"/>
  </logger>
[...]
<!-- MIDAS adpators -->
  <logger name="com.bes.itm.comp.servicemix" additivity="false">
    <level value="DEBUG"/>
    <appender-ref ref="midas"/>
  </logger>
[...]
<!-- auditing -->
  <logger
name="com.bes.itm.comp.servicemix.DORequestAuditTransformer"
additivity="false">
    <level value="INFO"/>
    <appender-ref ref="audit"/>
  </logger>
[...]
```

```
  <!-- default -->
  <root>
    <level value="INFO"/>
    <appender-ref ref="midas"/>
  </root>
</log4j:configuration>
```

An **<appender>** tag describes the actual log target (e.g. name of log file, entry format, rolling behavior, ...).

A **<logger>** tag defines which appender to use with which log level for a Administration UI component. Therefore, the amount logged is defined here.

If you want to configure the log level, change the appropriate **<logger>** tag; to change the log file behavior itself, change the **<appender>** tag.

The next table displays a list of the trace-level settings you can choose in log files:

Log level	**Description**
DEBUG	Most detailed level of logged information.
INFO	Detailed logging level to report minor and major error conditions, warnings, as well as correct system behavior.
WARN	Detailed logging of all unusual warning and error conditions.
ERROR	Level to report only error conditions. Warnings and correct system behavior is not logged.
FATAL	Only critical system failures are logged.

CAUTION: The log level to NONE disables error logging completely. It is strongly recommended to set the log level at least to WARN; ideally, you should set the logging level to INFO.

Web Application Logs

To configure additional logging of the Administration UI Web Application, use the `logkit.xconf` file

```
# cat /opt/OV/OMU/adminUI/webapps/midas/work/webapp/WEB-INF/
logkit.xconf
[...]
  <targets>
  [...]
    <!--
      This log file gets only messages with log level ERROR and below.
    -->
    <priority-filter id="error" log-level="ERROR">
      <cocoon>
        <filename>${context-root}/WEB-INF/logs/error.log</filename>
        <format type="cocoon">          %7.7{priority} %{time}
[%{category}] (%{uri}) %{thread}/%{class:short}:
%{message}\n%{throwable}
        </format>
        <append>false</append>
      </cocoon>
    </priority-filter>
    <cocoon id="debug">
      <filename>${context-root}/WEB-INF/logs/debug.log</filename>
      <format type="cocoon">          %7.7{priority} %{time}
[%{category}] (%{uri}) %{thread}/%{class:short}:
%{message}\n%{throwable}
      </format>
      <append>false</append>
    </cocoon>
[...]
```

The WebApp actually uses Apache logkit (http://excalibur.apache.org/logger.html), but the settings are very similar as shown in the example above.

Auditing

Auditing covers the tasks of keeping a log of who has done what and when. This is supported and enabled by default. Technically, log4j is used for this. Therefore, all configuration regarding auditing has to be done by modifying log4j configuration files. For details see chapter Log Files and Trace Levels on page 56.

For auditing, there are two log levels:

- **INFO** - one line per operation representing a summary, who did what. However not all details are are tracked.
- **DEBUG** - the full internal requests and responses exchanged of an operation are logged.

An example of the according audit records (INFO level) looks like this:

```
INFO,2009-04-07 12:44:48,168,modifyresponse,1239101088110,
5fnw9g49a0,tge,Web UI,avocado_server,ovoconfig,modify,ovo:policy,
,ok,,,,,,1,,,Bad Logs (11.x HP-UX),logfile,2.0,,,,false,false,true
```

Here, the user *tge* has modified the logfile policy named *Bad Logs (11.x HP-UX)* on the BackEnd *avocado_server*.

With DEBUG level set inside `log4j.xml`

```
[...]
  <!-- auditing -->
  <logger
name="com.bes.itm.comp.servicemix.DORequestAuditTransformer"
additivit
y="false">
    <level value="DEBUG"/>
    <appender-ref ref="audit"/>
  </logger>
[...]
```

the full data flow will be captured and logged. This contains all details about the objects being modified. The same operation as above now looks like this:

```
DEBUG,2009-04-07 12:59:57,171,<modifyresponse ...
  [...]
  <backend do:type="String" xml:space="preserve">avocado_server</backend>
  <operation do:type="String" xml:space="preserve">modify</operation>
  <objectclass do:type="String" xml:space="preserve">ovo:policy</objectclass>
  <user do:type="String" xml:space="preserve">tge</user>
  <version do:type="String" xml:space="preserve">2.1</version>
[...]
  <objectname do:type="String" xml:space="preserve">Bad Logs (11.x HP-UX)</
objectname>
  [...]
```

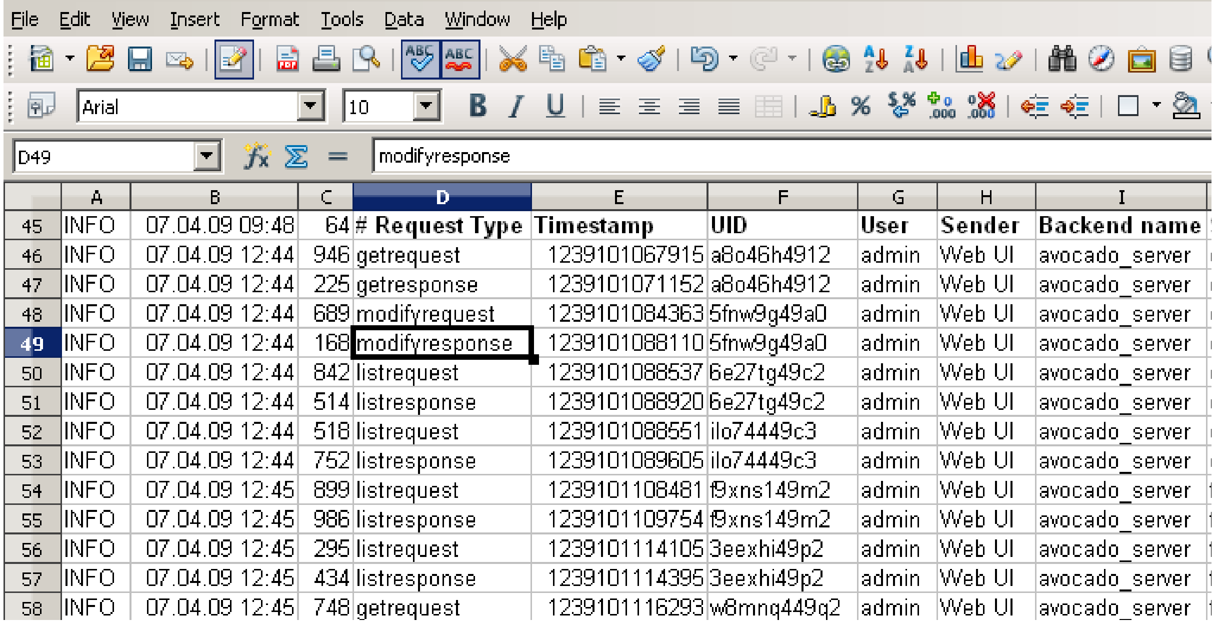
The audit files are written in `/opt/OV/OMU/adminUI/logs/audit`. The active log file currently being written is `audit.log`. By default, this file will be rolled daily yielding files like `audit.log.2009-03-20`. This behavior can be controlled by editing `/opt/OV/OMU/adminUI/conf/log4j.xml`:

```
<appender name="audit"
class="org.apache.log4j.DailyRollingFileAppender">
  <param name="File" value="logs/audit/audit.log"/>
  <param name="DatePattern" value="'.'yyyy-MM-dd"/>
```

```
    <layout class="org.apache.log4j.PatternLayout">
      <param name="ConversionPattern" value="%p,%d,%m%n"/>
    </layout>
  </appender>
  [...]
  <logger name="com.bes.itm.comp.servicemix.DORequestAuditTransformer"
          additivity="false">
    <level value="DEBUG"/>
    <appender-ref ref="audit"/>
  </logger>
```

Administration UI itself currently does not provide any tools to review the audit records. However, the audit log files can be loaded as CSV files into spreadsheet applications like MS Excel or OpenOffice for evaluation. (Figure 9 on page 61):

Figure 9 Spreadsheet Import

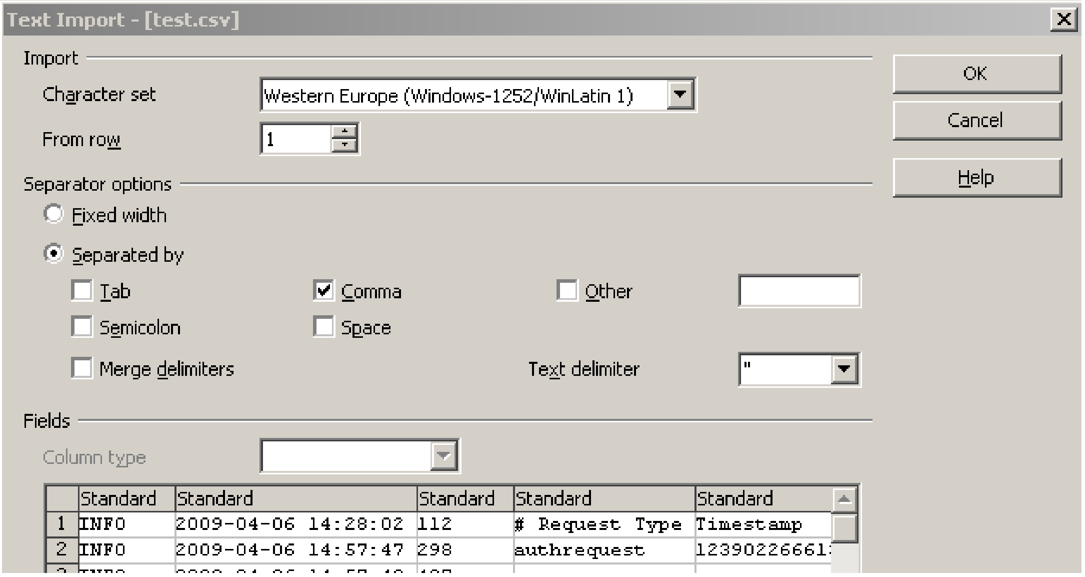


D49 modifyresponse

	A	B	C	D	E	F	G	H	I
45	INFO	07.04.09 09:48	64	# Request Type	Timestamp	UID	User	Sender	Backend name
46	INFO	07.04.09 12:44	946	getrequest	1239101067915	a8o46h4912	admin	Web UI	avocado_server
47	INFO	07.04.09 12:44	225	getresponse	1239101071152	a8o46h4912	admin	Web UI	avocado_server
48	INFO	07.04.09 12:44	689	modifyrequest	1239101084363	5fnw9g49a0	admin	Web UI	avocado_server
49	INFO	07.04.09 12:44	168	modifyresponse	1239101088110	5fnw9g49a0	admin	Web UI	avocado_server
50	INFO	07.04.09 12:44	842	listrequest	1239101088537	6e27tg49c2	admin	Web UI	avocado_server
51	INFO	07.04.09 12:44	514	listresponse	1239101088920	6e27tg49c2	admin	Web UI	avocado_server
52	INFO	07.04.09 12:44	518	listrequest	1239101088551	ilo74449c3	admin	Web UI	avocado_server
53	INFO	07.04.09 12:44	752	listresponse	1239101089605	ilo74449c3	admin	Web UI	avocado_server
54	INFO	07.04.09 12:45	899	listrequest	1239101108481	f9xns149m2	admin	Web UI	avocado_server
55	INFO	07.04.09 12:45	986	listresponse	1239101109754	f9xns149m2	admin	Web UI	avocado_server
56	INFO	07.04.09 12:45	295	listrequest	1239101114105	3eexhi49p2	admin	Web UI	avocado_server
57	INFO	07.04.09 12:45	434	listresponse	1239101114395	3eexhi49p2	admin	Web UI	avocado_server
58	INFO	07.04.09 12:45	748	getrequest	1239101116293	w8mnq449q2	admin	Web UI	avocado_server

If needed, select the comma character as a separator (Figure 10 on page 61):

Figure 10 Import Using Comma as Separator



Text Import - [test.csv]

Import

Character set: Western Europe (Windows-1252/WinLatin 1)

From row: 1

Separator options

○ Fixed width

● Separated by

☐ Tab ☑ Comma ☐ Other

☐ Semicolon ☐ Space

☐ Merge delimiters

Text delimiter: "

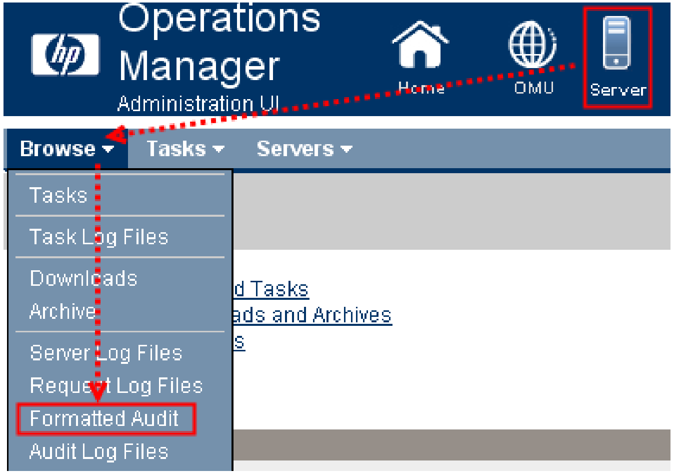
Fields

Column type

	Standard	Standard	Standard	Standard	Standard
1	INFO	2009-04-06 14:28:02	112	# Request Type	Timestamp
2	INFO	2009-04-06 14:57:47	298	authrequest	12390226661

Furthermore, there is also a formatted audit output available which can be viewed inside the Administration UI web interface. This function can be accessed via the context of the Server icon -> “Browse” -> “Formatted Audit” (Figure 11 on page 62).

Figure 11 Formatted Audit Log Access



Audit Fields – The contents in the different columns are (from left to right):

log level	log level, usually INFO
log date	time stamp when entry was logged
request type	type of request or response (getrequest, getresponse, listrequest, listresponse, ...)
timestamp	time stamp in Unix time when the request was created
uid	A unique ID of the request that can also be found in the response or follow-on requests
user	the user name that sent the request
sender	from which module did the request come, usually Web GUI
backend name	backend identifier the request is being routed to
service name	target adaptor name the request is being routed to
operation	operation performed (get,list,create,modify,delete,assign, etc)
object class	class of object handled with the request. Mass requests have a object class name "all" with namespace prefix.
context	class of object handled with the request. Mass requests have a object class name "all" with namespace prefix.
status	for responses,the status of the response operation (ok or error)
details mode	details mode that allow to modify the output format of get and list operations etc.
force flag	flag often used to enforce a operation (e.g. caching, backend resolution, lock overide etc.)
comment	optional text entered in comment field
version	version in version control system

Depending on the type of request further attributes may be logged such as objectname, objecttype, contextobjectname, or flags like recursive, all or inherited.

Request Logging Mechanism

Overview

There is also an additional request logging mechanism available, logging every operation in detail. Additionally those up to now “unloggable” internal requests can be logged.

Please note that by default the request logging mechanism is **NOT** enabled, since a considerable amount of disk space will be needed! Therefore, when enabling this please check regularly your disk space for `/opt/OV/OMU/adminUI/logs/requests`.

NOTE: This much more detailed logging mechanism is only recommended for debugging or if this really needed.

The request logging mechanism can be enabled for:

- audit request logging: typical user actions, e.g. edit, add, remove requests
- internal request logging: internal server requests, e.g. triggered by a VCS checkin. When e.g. a policy group is checked into VCS all policies and policy groups need to be resolved. These resolve operations will also now be logged.

Setup

As explained previously apart from the standard `audit.log`, it is possible to enable the much more detailed request logging mechanism in order to log the users actions in detail. But this is usually only recommended for troubleshooting purposes, since this will push the used disk space and also CPU consumption. The request logging mechanism can be enabled inside

```
/opt/OV/OMU/adminUI/conf/becore.properties
```

“*false*” disables the request logging, whereas “*true*” enables it.

The contents of `becore.properties` is:

```
# layer config file for backend documentor features
# request auditing (incoming requests/responses)
auditing = false
# request logging (authenticated or internal requests/responses)
logging = false
# eof
```

Two parameters can be modified:

- **auditing**: typical user actions, e.g. edit, add, remove requests
- **logging**: this refers to the internal request logging for those items otherwise not logged during internal Administration UI operations. Again this is designed especially for debugging purposes.

Audit.log lists all external requests, while request.log (when enabled as described above) will list adapter specific internal requests.

High Availability Environments

The basic concepts of using and installing Administration UI in a HA cluster are described in the separate Administration UI Installation Guide.

Advanced Communication Options

This section describes how to set up advanced communication options to improve the performance of the installed software. The information covers the following topics in greater detail:

- Changing Default Ports on page 64
- Using HTTPS on page 65
- Using Proxies on page 71
- Using Administration UI in Firewall Environments on page 71

Changing Default Ports

The communication relationships and default ports used by Administration UI are explained in chapter Communication and Ports on page 5.

Most ports are defined during the installation.

> If you modify port settings in any way, remember to perform a clean restart Administration UI!
>
> ```
> # /opt/OV/OMU/adminUI/adminui clean
> ```
>
> ```
> # /opt/OV/OMU/adminUI/adminui start
> ```
>
> Also make sure all users are informed of the downtime.

If it is necessary to change any port, hostname or identifier, please refer to one of these sections:

- Renaming the BackEnd Identifier on page 35
- Changing the Hostname on page 35
- Changing the BackEnd Port (9661) on page 35
- Changing the WebApp’s HTTP or/and HTTPS Port(s) on page 36
- Disabling the WebApp’s HTTP Port (9662) on page 37
- JMX Port Change on page 38

Using HTTPS

This section explains how to configure HTTPS to improve the security of communication between the installed components. The information covers the following areas:

- Understanding HTTPS on page 65

- Configuring HTTPS in Administration UI on page 66

- HTTPS between Browser and Web Application on page 66

- Using Custom Certificates on page 69

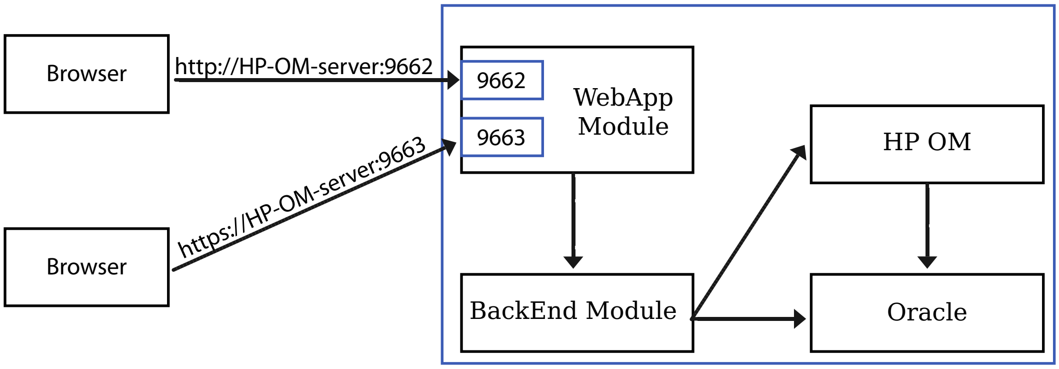
Understanding HTTPS

HTTPS communication provides the following two security-related features in addition to basic HTTP:

- authentication – the communication peers can be sure to really talk to the partner they think they do
- encryption – the HTTPS data flow is SSL encrypted

Figure 12 on page 65 shows the relationship between HTTP and HTTPS within Administration UI:

Figure 12 Communication Protocols



- GUI/WebApp – the WebApp acts as HTTP/HTTPS server and the GUI as client

There are the following aspects of HTTPS communication within by Administration UI:

- Authentication
  - Server authentication – the HTTPS server has to present some credential the client can verify. This way the client can be sure to talk to a real server. This is mandatory when using HTTPS.
  - Client authentication – in addition to server authentication, also the client has to authenticate itself to the server. This way also the server knows that it talks to the client it thinks it does.
- Authorization – this is actually not part of HTTPS but can be used as well, particularly to restrict the set of clients who are allowed to perform operations on the server.

It is also conceivable and useful for the GUI – WebApp communication to allow only a certain set of users access (those who have a client certificate which is registered at the WebApp).

Configuring HTTPS in Administration UI

HTTPS communication can be used within Administration UI in this combination:

- between GUI and WebApp
  - server authentication only
  - client authentication (optional, by default disabled)

By default, Administration UI generates self-signed certificates on all Administration UI servers which can be used for out-of-the-box HTTPS communication. If this is an acceptable security level, nothing has to be done in addition. Otherwise certificates may need to be generated and imported key- and truststores.

Certificates within Administration UI are standard PKCS12 certificate file in DER format, which can be generated using any desired mechanism, e.g. `keytool`, openssl, the SSH key generator command or the HPOM SecCore functionality. Again, `keytool` is recommended.

The key- and truststores used by an Administration UI server are in

```
# ls -l /opt/OV/OMU/adminUI/conf/servicemix/*store*
-r--r--r--  1 root root 1093 2006-11-22 16:24 keystore_endpoint.jks
-r--r--r--  1 root root 3243 2006-11-15 09:32 keystore.jks
-r--r--r--  1 root root  686 2006-11-22 16:24 truststore_endpoint.jks
-r--r--r--  1 root root  662 2006-11-15 09:32 truststore.jks
```

The files named *_endpoint.jks are related to the Administration UI HTTPS communication. **All following references to keystore and truststore are related to these two files.**

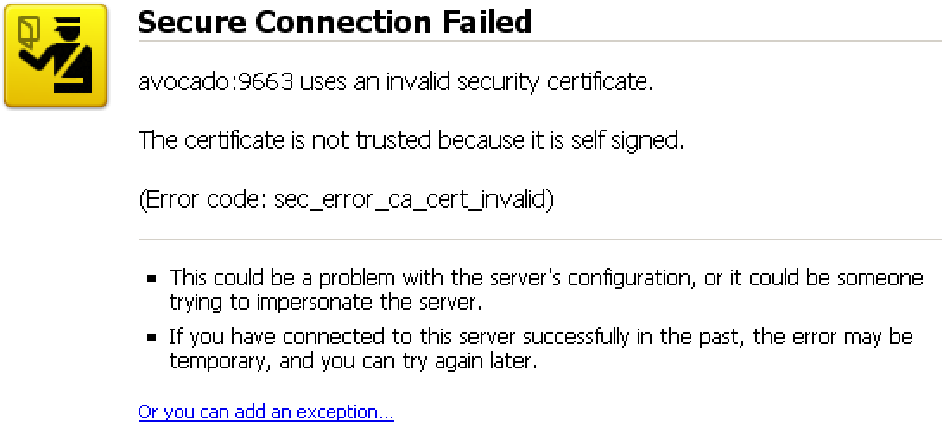
The other two files are required by infrastructure components (Jetty Web container and ServiceMix). These must NOT be modified.

Authorization within Administration UI is controlled entirely by the Administration UI user model. Therefore, there is nothing to be configured elsewhere. If an HTTPS client is able to establish an HTTPS connection (and the certificate exchange has been successful), it is also considered as authorized to perform all tasks.

HTTPS between Browser and Web Application

For out-of-the-box server-side authentication, nothing has to be configured, the user has just to accept the WebApp certificate in the browser. Figure 13 on page 67: shows a situation in which the web browser complains that it cannot validate the default self-signed certificate, because it does not have the signing CA certificate.

Figure 13 Self Sign Warning



Secure Connection Failed

avocado:9663 uses an invalid security certificate.

The certificate is not trusted because it is self signed.

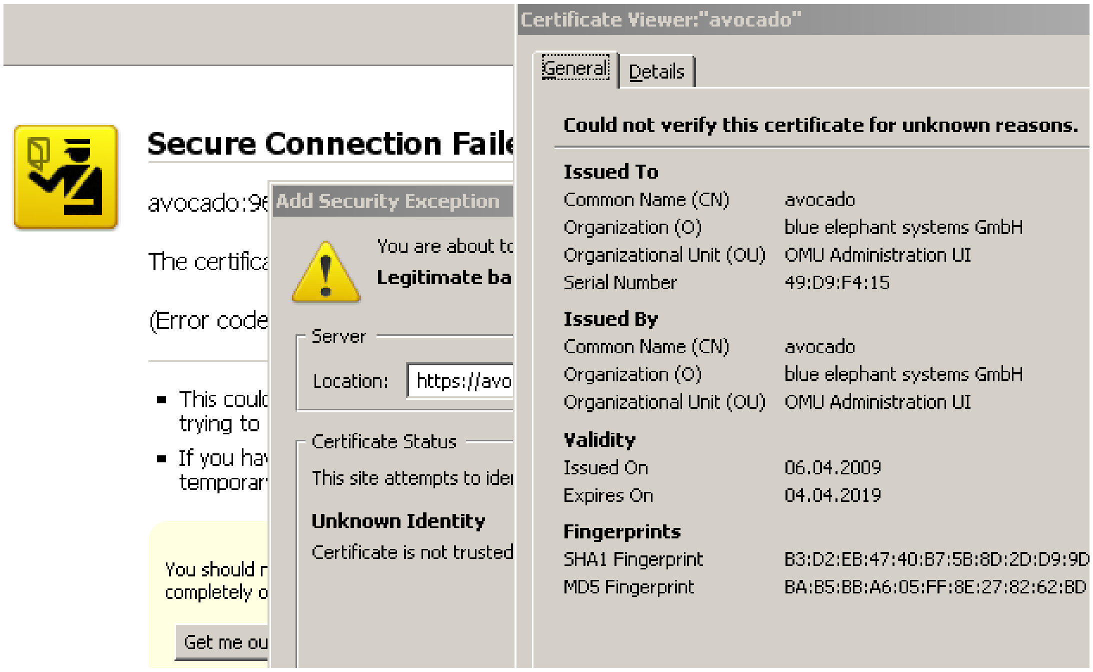
(Error code: sec_error_ca_cert_invalid)

- This could be a problem with the server's configuration, or it could be someone trying to impersonate the server.
- If you have connected to this server successfully in the past, the error may be temporary, and you can try again later.

Or you can add an exception…

To inspect the certificate add an exception and click the link “Or you can add an exception...” and hit the “Add Exception” button, which displays the dialog illustrated in Figure 14 on page 67. Here you can fetch the certificate, view it and finally confirm the security exception that it will be added in the browser.

Figure 14 Add Exception and View Certificate



Certificate Viewer:"avocado"

Could not verify this certificate for unknown reasons.

Issued To

Common Name (CN)	avocado
Organization (O)	blue elephant systems GmbH
Organizational Unit (OU)	OMU Administration UI
Serial Number	49:D9:F4:15

Issued By

Common Name (CN)	avocado
Organization (O)	blue elephant systems GmbH
Organizational Unit (OU)	OMU Administration UI

Validity

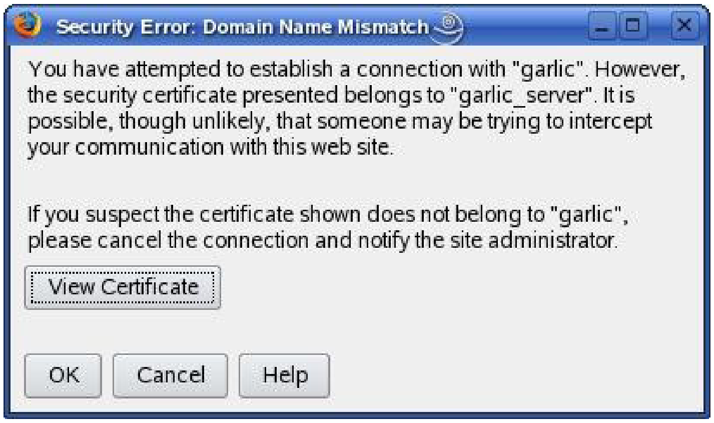
Issued On	06.04.2009
Expires On	04.04.2019

Fingerprints

SHA1 Fingerprint	B3:D2:EB:47:40:B7:5B:8D:2D:D9:9D
MD5 Fingerprint	BA:B5:BB:A6:05:FF:8E:27:82:62:BD

Figure 15 on page 68 displays the warning that might appear if there is mismatch between the server host name and the common name (CN), which is the name to which the HTTPS server certificates are usually issued. Since this has not been implemented in Administration UI click OK to proceed.

Figure 15 Domain Name Mismatch



Security Error: Domain Name Mismatch

You have attempted to establish a connection with "garlic". However, the security certificate presented belongs to "garlic_server". It is possible, though unlikely, that someone may be trying to intercept your communication with this web site.

If you suspect the certificate shown does not belong to "garlic", please cancel the connection and notify the site administrator.

You can choose whether to accept this certificate permanently or repeat this process again next time. If accepting the certificate permanently, it will be placed in the web browsers truststore.

> If the names displayed in the Domain-Name-Mismatch dialog are not the HPOM host name and the Administration UI server ID defined during installation of the Administration UI, it is possible that a real security threat exists.

This problem can be avoided entirely, if the WebApp certificate is signed by a certification authority (CA) registered in the web browser. To install a signed certificate on the WebApp and register it in the log-in browser:

1) Put a CA-signed certificate in the Administration UI keystore

2) Load the CA certificate into the browser you want to use to login

3) After exchanging the certificates, restart the Administration UI application.

In addition, you may configure client-side certificates for the GUI – WebApp relationship:

- Create CA-signed certificate and load this into the web browser as client certificate
- Put the CA certificate of the CA, who has signed the client certificate, into Administration UI truststore (if not yet in)
- Configure the Administration UI to enforce client-side certificates (see below)

When connecting to the Administration UI with the web browser, you may need to select the correct client certificate (if there is more than one)

To configure client certificates in Administration UI, set the property needClientAuth as:

```
# vi /opt/OV/OMU/adminUI/conf/jetty.xml
jetty.connectors[0].port=9662
jetty.connectors[1].port=9663
jetty.connectors[1].needClientAuth=true
```

Using Custom Certificates

If custom keys and certificates are to be used (e.g. the customer has an own CA or keys/certificates used elsewhere), just the default keys and certificates need to be replaced.

To install a custom CA certificate, it must be imported to the web browser or the JRE on the Administration UI server. If the Administration UI server certificate has not been signed by a well-known CA, it is necessary to register that CA's certificate as trusted on the HTTPS client. This is not needed if the server certificate has been issued by any CA which is listed when calling the following command on the Administration UI server:

```
# /opt/OV/OMU/adminUI/jre/bin/keytool -list -keystore \     /opt/OV/
OMU/adminUI/jre/jre/lib/security/cacerts -storepass changeit
[...]
thawtepersonalfreemailca, Feb 12, 1999, trustedCertEntry,
Certificate fingerprint (MD5):
  1E:74:C3:86:3C:0C:35:C5:3E:C2:7F:EF:3C:AA:3C:D9
thawtepersonalbasicca, Feb 12, 1999, trustedCertEntry,
Certificate fingerprint (MD5):
  E6:0B:D2:C9:CA:2D:88:DB:1A:71:0E:4B:78:EB:02:41
verisignclass3ca, Jun 29, 1998, trustedCertEntry,
Certificate fingerprint (MD5):
  78:2A:02:DF:DB:2E:14:D5:A7:5F:0A:DF:B6:8E:9C:5D
thawtepersonalpremiumca, Feb 12, 1999, trustedCertEntry,
Certificate fingerprint (MD5):
  3A:B2:DE:22:9A:20:93:49:F9:ED:C8:D2:8A:E7:68:0D
[...]
```

If your server certificate has been issued by any of these CAs it is trusted automatically - then nothing needs to be done here. Otherwise import your CA certificate into the Administration UI truststore by executing:

```
# /opt/OV/OMU/adminUI/jre/bin/keytool -import -keystore \
/opt/OV/OMU/adminUI/conf/servicemix/truststore_endpoint.jks
-storepass \ <password> -file <your CA cert file in DER format>
Then the truststore looks like this:
Alias name: mykey
Creation date: Dec 4, 2003
Entry type: trustedCertEntry

Owner: EMAILADDRESS=tge@blue-elephant-systems.com, CN=tge, OU=R&D,
O=blue elephant systems GmbH, L=Stuttgart, ST=Baden-Wuerttemberg,
C=DE
Issuer: EMAILADDRESS=tge@blue-elephant-systems.com, CN=tge, OU=R&D,
O=blue elephant systems GmbH, L=Stuttgart, ST=Baden-Wuerttemberg,
C=DE
Serial number: Administration UI
Valid from: Wed Dec 03 14:26:54 CET 2003 until: Fri Dec 02 14:26:54
CET 2005
Certificate fingerprints:
         MD5:  14:D1:9B:08:7B:4D:61:B3:D4:29:B8:26:E9:A2:B5:CE
         SHA1:
9D:01:8D:15:D6:C7:8C:93:21:FF:39:6A:3E:74:BC:08:36:9F:8E:61
```

Here the -v option has been used to print more details. Note again the entry type trustedCertEntry, the certificate identity and the CA identity - for CA certificate they are normally identically (here both are 'tge').

To use custom client or server certificates (in this example for the Administration UI server ios_server), perform the following steps:

1) Create a keystore containing the custom certificate:

```
# keytool -genkey -keystore /tmp/my_keystore \
 -storepass <password> -alias ios_server
```

The alias name must be the Administration UI server ID.

2) Answer the questions about your identity. Then extract a certificate request which has to be sent to a CA:

```
# keytool -certreq -alias ios_server -keystore /tmp/my_keystore
 -storepass <password> -file ios-certreq.pem
```

3) The CA signs the certificate request and returns the certificate (needed in DER format). Import this into the keystore:

```
# keytool -import -alias ios_server -keystore
          <MIDAS_HOME>/conf/servicemix/keystore_endpoint.jks
          -storepass <password> -file ios-cert.der
```

The actual value of the alias is not important for client certificates, but it must be the same as specified when generating the initial key pair. Then the contents of the keystore looks like this:

```
Alias name:  ios_server
Creation date: Dec 8, 2003
Entry type: keyEntry
Certificate chain length: 2
Certificate[1]:
Owner: CN=ios_server, OU=TM4WM, O=blue elephant systems GmbH,
ST=Baden-Wuerttemberg, C=DE
Issuer: EMAILADDRESS=tge@blue-elephant-systems.com, CN=tge, OU=R&D,
O=blue elephant systems GmbH, L=Stuttgart, ST=Baden-Wuerttemberg,
C=DE
Serial number: 5
Valid from: Mon Dec 08 18:42:13 CET 2003 until: Tue Dec 07 18:42:13
CET 2004
Certificate fingerprints:
         MD5:  04:FD:A6:95:F4:B1:19:BC:70:E8:5E:48:D6:3D:CA:17
         SHA1:
59:61:C7:DB:E9:34:17:9F:5B:9D:E2:14:B0:B0:6E:40:B0:C7:8B:A6
Certificate[2]:
Owner: EMAILADDRESS=tge@blue-elephant-systems.com, CN=tge, OU=R&D,
O=blue elephant systems GmbH, L=Stuttgart, ST=Baden-Wuerttemberg,
C=DE
Issuer: EMAILADDRESS=tge@blue-elephant-systems.com, CN=tge, OU=R&D,
O=blue elephant systems GmbH, L=Stuttgart, ST=Baden-Wuerttemberg,
C=DE
Serial number: 0
```

```
Valid from: Wed Dec 03 14:26:54 CET 2003 until: Fri Dec 02 14:26:54
CET 2005
Certificate fingerprints:
         MD5:  14:D1:9B:08:7B:4D:61:B3:D4:29:B8:26:E9:A2:B5:CE
         SHA1:
9D:01:8D:15:D6:C7:8C:93:21:FF:39:6A:3E:74:BC:08:36:9F:8E:61
```

Here we see the certificate issued to the identity of ios_server by the CA tge. The entry type keyEntry states that this is not a CA certificate but a regular private key/ certificate pair.

4) Finally, the CA's certificate has to be registered on the HTTPS server as trusted certificate.

Using Proxies

Using a HTTP proxy between Web browser and Administration UI WebApp is currently not supported.

Using Administration UI in Firewall Environments

Please see Communication and Ports on page 5

Tuning Java Parameters

The information in this section describes how to set up important Java parameters, for example: the amount of virtual memory that is available, the size of the stack for Java threads, and how to control server start-up times. The information covers the following areas:

- Virtual Memory on page 72

- JRE Start-up on page 73

Virtual Memory

For large numbers of users or objects, and if the system the Administration UI server is running on has plenty of RAM installed, performance can be improved significantly by allowing the JRE to obtain more virtual memory.

The recommended max. amount of RAM is 1024mb or 2048mb where there is enough physical memory available.

In order to change the memory setting please edit:

```
# /opt/OV/OMU/adminUI/conf/servicemix/wrapper.conf
```

Search for this block:

```
[...]
# Maximum Java Heap Size (in MB)
wrapper.java.maxmemory=512
```

and change it to

```
# Maximum Java Heap Size (in MB)
wrapper.java.maxmemory=1024
```

An application restart is required. Use the

```
# /opt/OV/OMU/adminUI/adminui restart
```

command for this.

Do no decrease the value below the initial setting – this will also decrease performance and Administration UI may not function properly anymore.

Don’t forget to inform your users about the downtime.

JRE Start-up

On slow systems, the start-up of the Administration UI server may take quite a while. To prevent the controlling wrapper process from erroneously re-starting the JRE, the following parameter may be increased (in bold):

```
# vi /opt/OV/OMU/adminUI/conf/servicemix/wrapper.conf
[...]
# Number of seconds to allow for the JVM to be launched and contact
the
# wrapper before the wrapper should assume that the JVM is hung and
# terminate the JVM process. Administration UI means never time out.
# Defaults to 30 seconds.
wrapper.startup.timeout=300
# Number of seconds to allow between the wrapper pinging the JVM and
# the response. Administration UI means never time out. Defaults to 30
seconds.
wrapper.ping.timeout=100
# Number of seconds to allow for the JVM to shutdown before the
wrapper # should assume that the JVM is hung and terminate the JVM
process.
# Administration UI means never time out. Defaults to 30 seconds.
wrapper.shutdown.timeout=300
[...]
```

After any of these changes please restart the application using:

```
# /opt/OV/OMU/adminUI/adminui restart
```

Don’t forget to inform your users about the downtime.

Web Interface Timeout

This section lists the configuration steps which are necessary to increase the timeout for the Administration UI web interface. Otherwise, for example if a user leaves the policy editor window open and returns after 3 hours his session will have timed out.

It is possible to increase the timeout. In order to do so three timeout settings have to be modified:

- (1) Continuation Timeout
- (2) Backend Session Timeout
- (3) Webapp Session Timeout

> It is important to know that these three timouts are dependent on each other. A strict hiearchy must be obeyed! From (1) to (3) the timeout must increase. In other words, the timeout of (2) must be greater than (1). And the timeout defined for (3) must be greater than (2). Any difference, for example of a minute or an hour between each entry is enough.
>
> Please also notice, these manual modifications will be lost after the installation of an Administration UI fixpack.
>
> It is also important to remember that the higher the timeout is set to, the higher the memory consumption will be.

Configuration

Remember to backup the following files before you modify them in any way.

Continuation Timeout

The configuration file can be found here:

```
/opt/OV/OMU/adminUI/conf/cocoon.properties
```

Search for the following line:

```
continuations-manager.time-to-live=3600000
```

If this line does not exist, simply add it. The position inside the `.properties` file is not of importance.

The value is in milliseconds (ms). Therefore, a calculation does look like this

(hours) x 60 x 60 x 1000 = (timeout in milliseconds)

Example:

6hours x 60 x 60 x 1000= 21600000

BackEnd Session Timeout

The configuration file can be found here:

```
/opt/OV/OMU/adminUI/conf/auth.properties
```

Search for the following line:

```
caches.timeout=7200000
```

Its value is also in milliseconds (ms).

WebApp Session Timeout

The configuration file is located here:

```
/opt/OV/OMU/adminUI/webapps/midas/work/webapp/WEB-INF/web.xml
```

Search for the following section:

```
[...]
  <session-config>
    <session-timeout>240</session-timeout>
  </session-config>
[...]
```

Important: here the entry is in minutes! After applying an application fixpack (patch) to Administration UI, this entry will be reset to the original value.

Again please remember that the different timeouts are configured in such a way, that the Continuation Timeout < (is smaller than the) Backend Session Timeout < (which is in return smaller than the) Webapp Session Timeout.

.

After such a modification a restart of Administration UI is necessary. The command is:

```
# /opt/OV/OMU/adminUI/adminui restart
```

Please inform your users about the downtime.

(intentionally left blank)

5 SSH Based Agent Installation

Overview

This chapter will cover the SSH based installation of HPOM agents via the Administration UI GUI:

Introduction

The goal is to install the HPOM agent binaries onto managed nodes:

- Initiated from HPOM server
  - Actual work done by HPOM `inst.sh` script
  - „Off-line installation“ is not considered
- Fresh install or update (patch install)
- Asynchronous request/response behavior
  - Purely non-interactive

HPOM always offered a method to install the agent software on a managed node. This agent installation could be triggered from the Motif GUI (HPOM 8) or from the command line – in both cases however, the „`inst.sh`“ script was used..

Administration UI also uses the „`inst.sh`“ script. There are no special or independent scripts in Administration UI for that purpose. Therefore, existing limitations or well known specifics still apply.

Installing an agent can mean either a complete fresh install or an update installation (for example to install an agent patch). Both ways are supported by `inst.sh` and therefore by Administration UI as well.

Unlike the Motif GUI, Administration UI is a web-based GUI. As a result it is not easily possible to start `inst.sh` as a child process and display input/output in a terminal.

Thus, the Administration UI agent installation is purely non-interactive and works in a request/response fashion. All parameters will be collected, checked and then passed on to the `inst.sh` script. When the `inst.sh` has finished the results will be sent back to the Administration UI and saved in an agent installation log file.

SSH Based Agent Installation

It is possible to use SSH during the the agent installation. If you plan to do this, make sure that “Use SSH during installation” is enabled for these nodes.

To do so, select for a node: “Edit” (Action Menu). Inside the tab “Installation” enable the checkbox [X] Use SSH during installation.

Please remember that the following pre-requisits also have to be met:

- if necessary, generate SSH keys on the HPOM server
- for each node, connect manually once to the node and accept the host key
- copy the public key of the HPOM server to the target node
- for each node, check the connection if ok (without being asked for a password!)
- edit `/opt/OV/OMU/adminUI/conf/ovoinstall.properties` uncommenting the line:

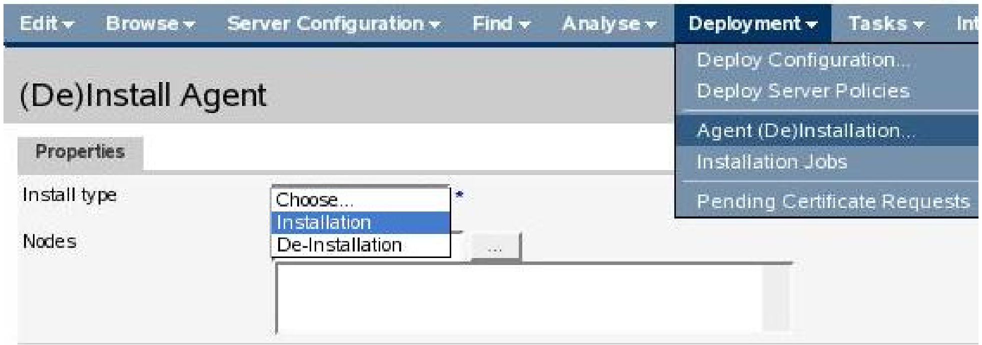
  ```
  ovoinstall.sshCmd = ssh -o StrictHostKeyChecking=no
  ```
- Restart Administration UI: `# ovc -restart adminui`
- Do a test installation by using standard mode (remsh)

Agent Installation Start

The agent installation can be triggered from two locations:

- For multiple in- or de-installations. From the Administrative menu select "Deployment" -> "Agent (De)Installation" (Figure 16 on page 79)

Figure 16 Agent Installation Start

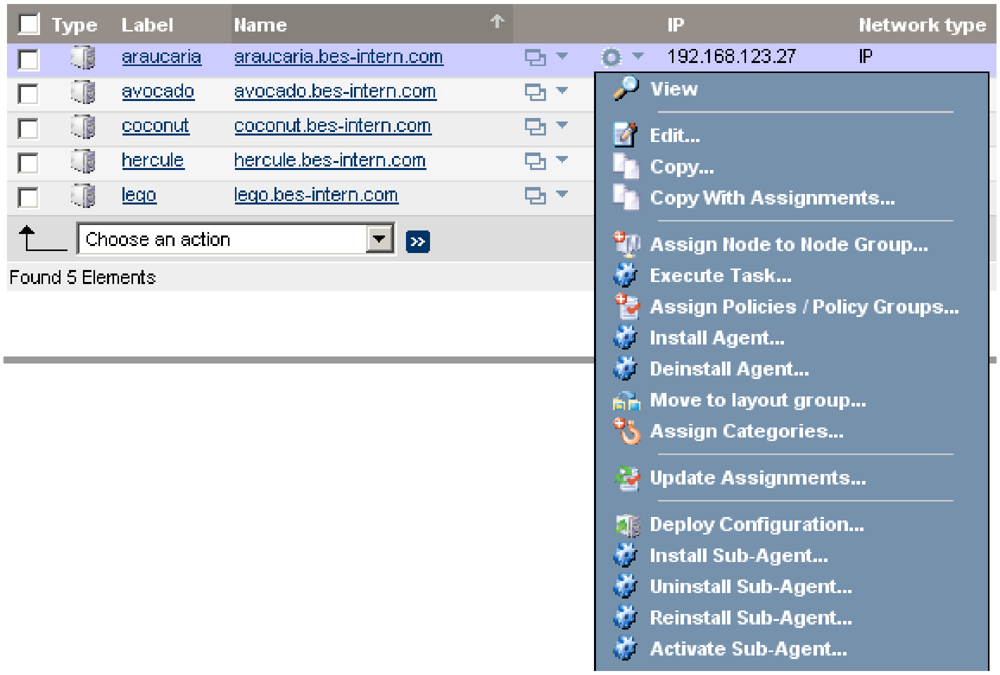


Install type: Choose... *

Nodes

- Single context based installation. Here use the context Action Menu of a node and select: "Install Agent" (Figure 17 on page 79)

Figure 17 Context Menu Installation Start



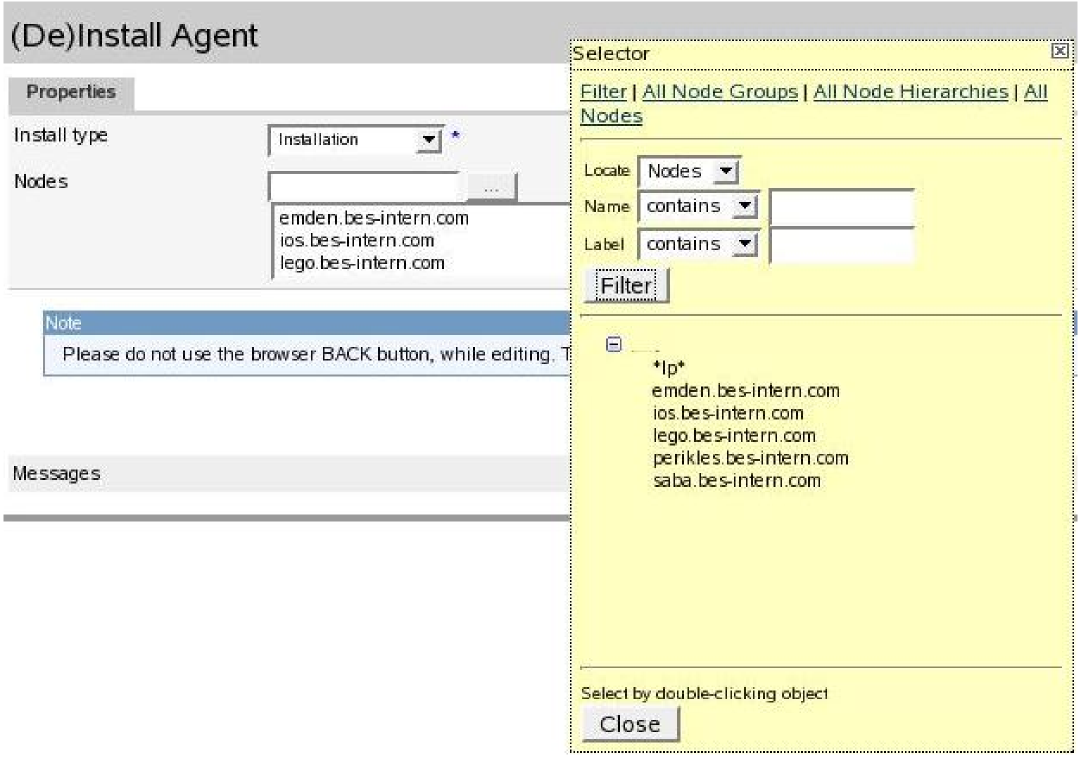
□ Type	Label	Name	IP	Network type
□	araucaria	araucaria.bes-intern.com	192.168.123.27	IP
□	avocado	avocado.bes-intern.com		
□	coconut	coconut.bes-intern.com		
□	hercule	hercule.bes-intern.com		
□	lego	lego.bes-intern.com		

Choose an action

Found 5 Elements

- When selecting the global menu "Deployment" -> "Agent (De)Installation" the first step is to choose the installation mode – installation or de-installation.
- Secondly, press the browse button and select the set of target nodes from the selector as shown below (Figure 18 on page 80). Node names may also be typed into the text field left of the "..." browse button, however the name must be literally identical to the primary host name as configured in HPOM (there is no name resolution!).

Figure 18 Selection of Target Node(s)



(De)Install Agent

Properties

Install type: Installation *

Nodes:
emden.bes-intern.com
ios.bes-intern.com
lego.bes-intern.com

Note
Please do not use the browser BACK button, while editing. T

Messages

Selector

Filter | All Node Groups | All Node Hierarchies | All Nodes

Locate: Nodes
Name: contains
Label: contains

lp
emden.bes-intern.com
ios.bes-intern.com
lego.bes-intern.com
perikles.bes-intern.com
saba.bes-intern.com

Select by double-clicking object

Select as many nodes as needed and press “Close” to close the selector window. After that please click the "Preinstall check" button to start the prerequisites analysis. This analysis may run a while and will validate all selected nodes.

Pre-Install Check Result

When the prerequisites analysis is complete, the list of target nodes is displayed as shown in the example below (Figure 19 on page 81). Please check for each node the analysis result. The important columns are:

- **Status**. This can indicate:
  - — *ready* (all prerequisites are fulfilled, installation may proceed)
  - — *invalid* (some error occurred; an installation is not possible)
  - — *passwdrequired* (prerequisites OK, but installation method requires a password, which has to be entered manually in the text field at the very right)
- **Method**. Lists the installation method, which either
  - — SSH (expressesly configured), see SSH Based Agent Installation on page 78.
  - — Has been discovered as applicable (*local*, *HTTPS*, *standard*) or if an error has occurred (*invalid*).
- **Comment**. Lists the reason of each status result. A click on the “?” will display more information.

More details can be found below.

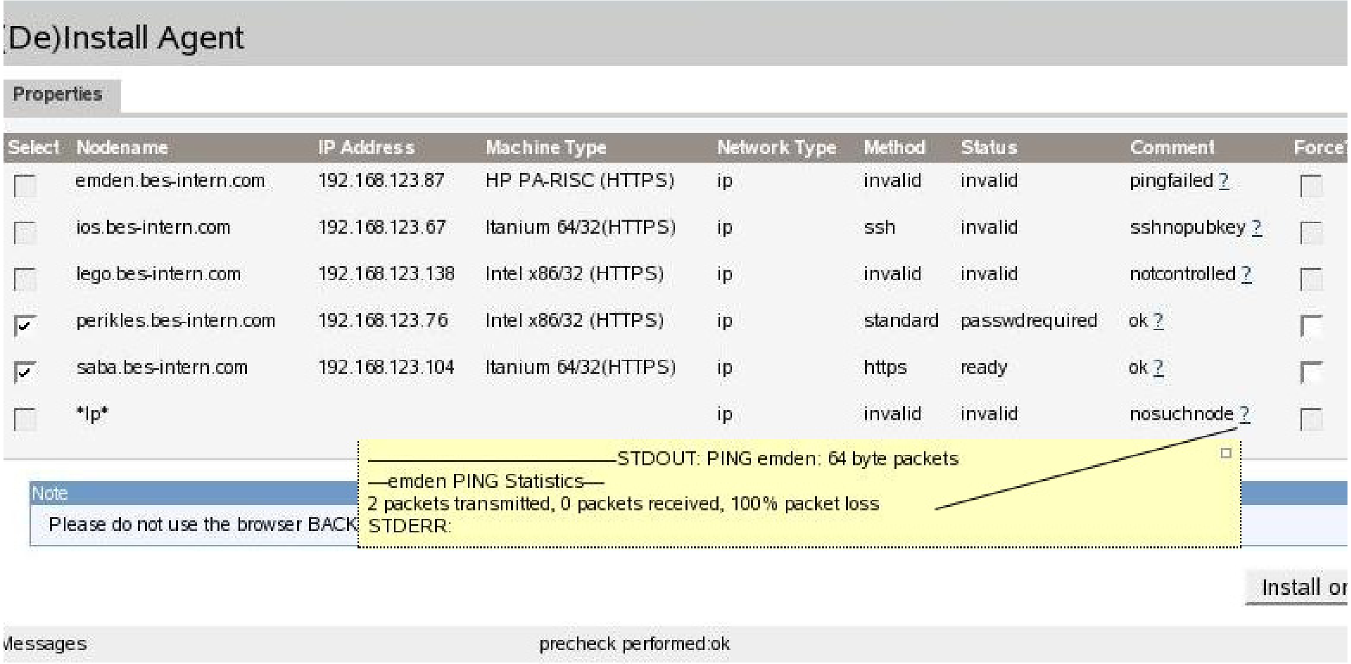
Nodes, which have been considered as installable, are automatically selected on the very left. If desired, individual nodes may be manually deselected to exclude them from the actual installation. Nodes, where an error occurred during the prerequisites check, are automatically deselected and cannot be selected.

If required, enter a password. In order to re-install an existing agent, enable the checkbox “Force”. This value will be passed on to the `inst.sh` script..

Only nodes passing the analysis phase can be installed!

If no nodes at all have been found as installable, the "Install ..." button will be greyed out and disabled.

Figure 19 Pre-Install Check Results



(De)Install Agent

Select	Nodename	IP Address	Machine Type	Network Type	Method	Status	Comment	Force?
□	emden.bes-intern.com	192.168.123.87	HP PA-RISC (HTTPS)	ip	invalid	invalid	pingfailed ?	□
□	ios.bes-intern.com	192.168.123.67	Itanium 64/32(HTTPS)	ip	ssh	invalid	sshnopubkey ?	□
□	lego.bes-intern.com	192.168.123.138	Intel x86/32 (HTTPS)	ip	invalid	invalid	notcontrolled ?	□
☑	perikles.bes-intern.com	192.168.123.76	Intel x86/32 (HTTPS)	ip	standard	passwdrequired	ok ?	□
☑	saba.bes-intern.com	192.168.123.104	Itanium 64/32(HTTPS)	ip	https	ready	ok ?	□
□	*Ip*			ip	invalid	invalid	nosuchnode ?	□

```
————————————————STDOUT: PING emden: 64 byte packets
—emden PING Statistics—
2 packets transmitted, 0 packets received, 100% packet loss
STDERR:
```

Note

Please do not use the browser BACK

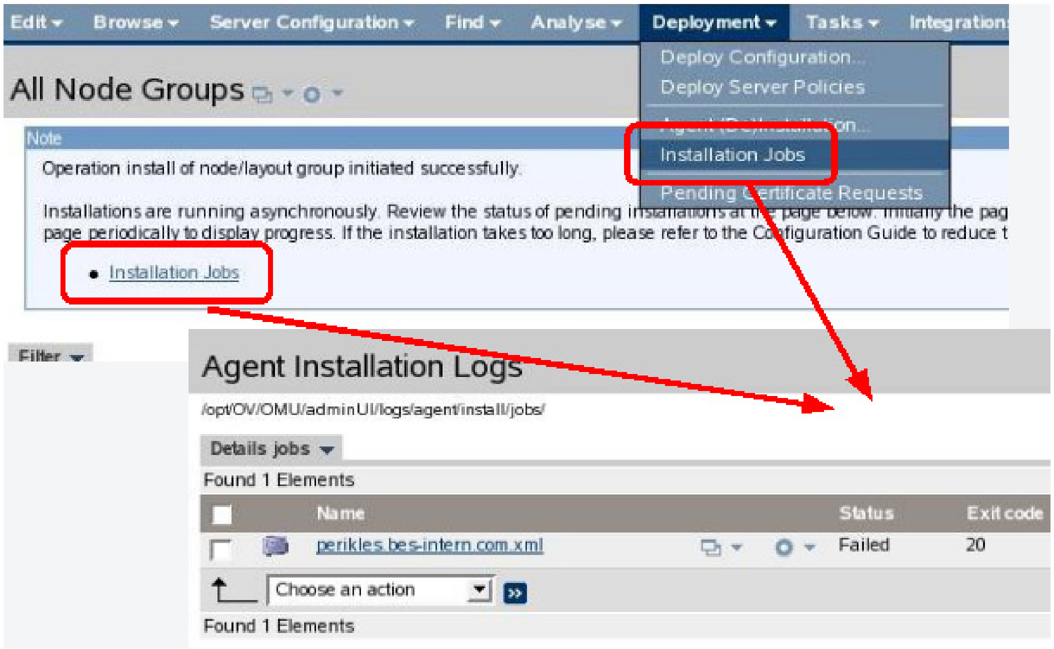
Messages precheck performed:ok

Main Installation Phase

After clicking the "Install ..." button, Administration UI will generate a parameter file for *each* selected node and execute the `inst.sh` script with these parameter files. The actual installation may take a long time. Therefore, Administration UI performs this operation asynchronously in the background in order to avoid communication timeouts.

The output of the `inst.sh` script will be captured and stored in the Administration UI BackEnd module (on the HPOM server) and can be reviewed later by following the link as shown in Figure 20 on page 82 below.

Figure 20 Installation Phase



All Node Groups

Note

Operation install of node/layout group initiated successfully.

Installations are running asynchronously. Review the status of pending installations at the page below. Initially the pag
page periodically to display progress. If the installation takes too long, please refer to the Configuration Guide to reduce t

- Installation Jobs

Agent Installation Logs

/opt/OV/OMU/adminUI/logs/agent/install/jobs/

Found 1 Elements

□	Name	Status	Exit code
□	perikles.bes-intern.com.xml	Failed	20

Choose an action

Found 1 Elements

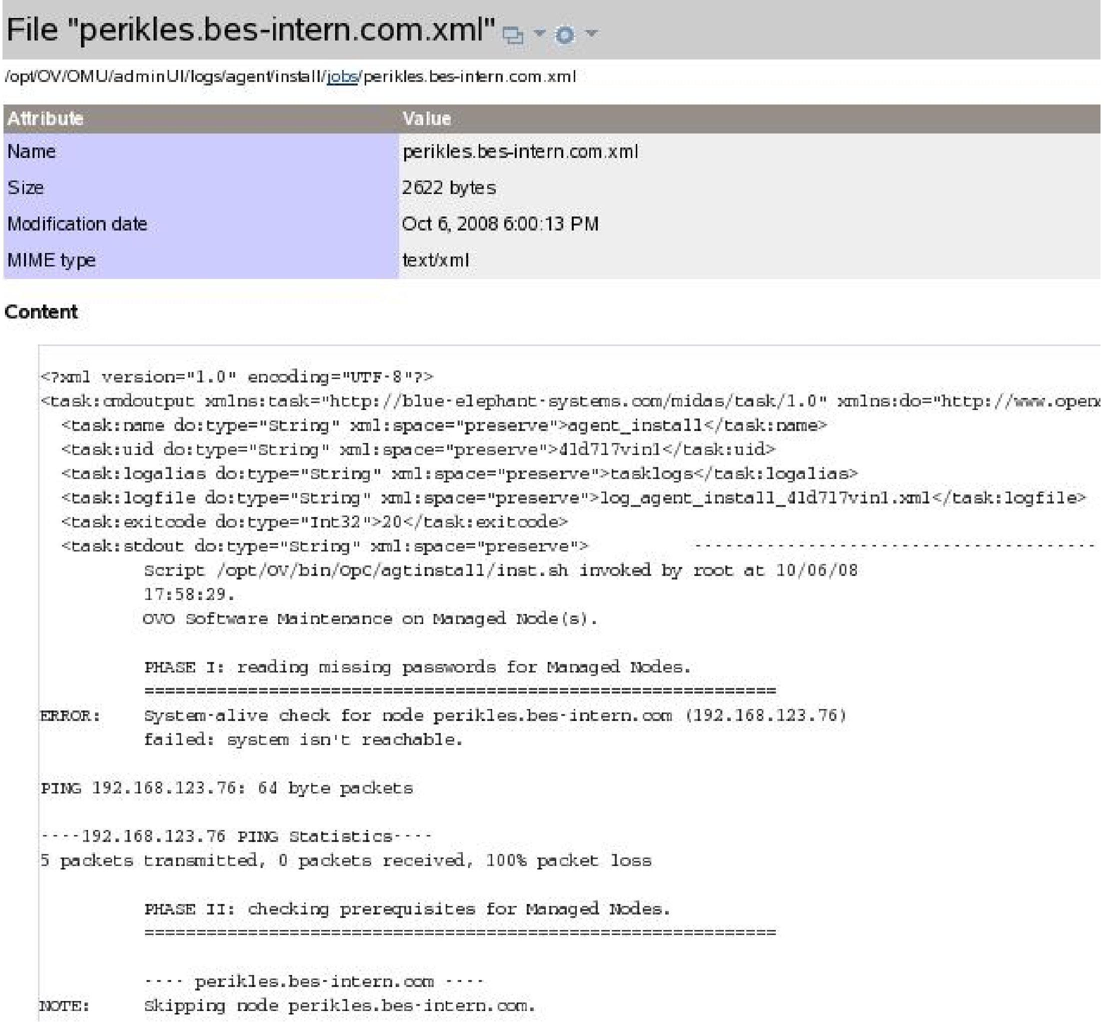
The installation takes place in a sequence (one node after the other as `inst.sh` cannot run concurrently multiple times) and may take a long time. Hence, the installation logs will appear one after another over time. Please refresh the log file view every now and then.

Installation Log

The content of the installation log is written by HPOM `inst.sh`. Review this output whether the actual agent installation was successful or not (Example Figure 21 on page 83). In the example the node “perikles” isn’t reachable.

In the Administration UI overview page, an exit code of zero is considered as successful.

Figure 21 Installation Log



File "perikles.bes-intern.com.xml"

/opt/OV/OMU/adminUI/logs/agent/install/jobs/perikles.bes-intern.com.xml

Attribute	Value
Name	perikles.bes-intern.com.xml
Size	2622 bytes
Modification date	Oct 6, 2008 6:00:13 PM
MIME type	text/xml

Content

```
<?xml version="1.0" encoding="UTF-8"?>
<task:cmdoutput xmlns:task="http://blue-elephant-systems.com/midas/task/1.0" xmlns:do="http://www.open
  <task:name do:type="String" xml:space="preserve">agent_install</task:name>
  <task:uid do:type="String" xml:space="preserve">41d717vin1</task:uid>
  <task:logalias do:type="String" xml:space="preserve">tasklogs</task:logalias>
  <task:logfile do:type="String" xml:space="preserve">log_agent_install_41d717vin1.xml</task:logfile>
  <task:exitcode do:type="Int32">20</task:exitcode>
  <task:stdout do:type="String" xml:space="preserve">          --------------------------------------
          Script /opt/OV/bin/OpC/agtinstall/inst.sh invoked by root at 10/06/08
          17:58:29.
          OVO Software Maintenance on Managed Node(s).

          PHASE I: reading missing passwords for Managed Nodes.
          ============================================================
ERROR:    System-alive check for node perikles.bes-intern.com (192.168.123.76)
          failed: system isn't reachable.

PING 192.168.123.76: 64 byte packets

----192.168.123.76 PING Statistics----
5 packets transmitted, 0 packets received, 100% packet loss

          PHASE II: checking prerequisites for Managed Nodes.
          ============================================================

          ---- perikles.bes-intern.com ----
NOTE:     Skipping node perikles.bes-intern.com.
```

Details About Pre-Installation Analysis

The flowchart below describes the steps performed during the prerequisites analysis for each node. The resulting states and methods are explained further down below.

The commands stated are executed as sub-processes of the Administration UI BackEnd server on the HPOM server. The output of the command can be viewed in the prerequisite analysis results list.

The SSH and ICMP ping commands are partially configurable through the Administration UI property file (its contents will be explained later):

```
# /opt/OV/OMU/adminUI/conf/ovoinstall.properties
```

Essentially these are the same tests which are also performed by the `inst.sh` script before actually starting the agent installation.

Method - Installation Method

When the prerequisites analysis is finished, the resulting installation method is shown. Please consult the flowchart (Figure 22 on page 85) indicating under which circumstances the corresponding method has been determined.

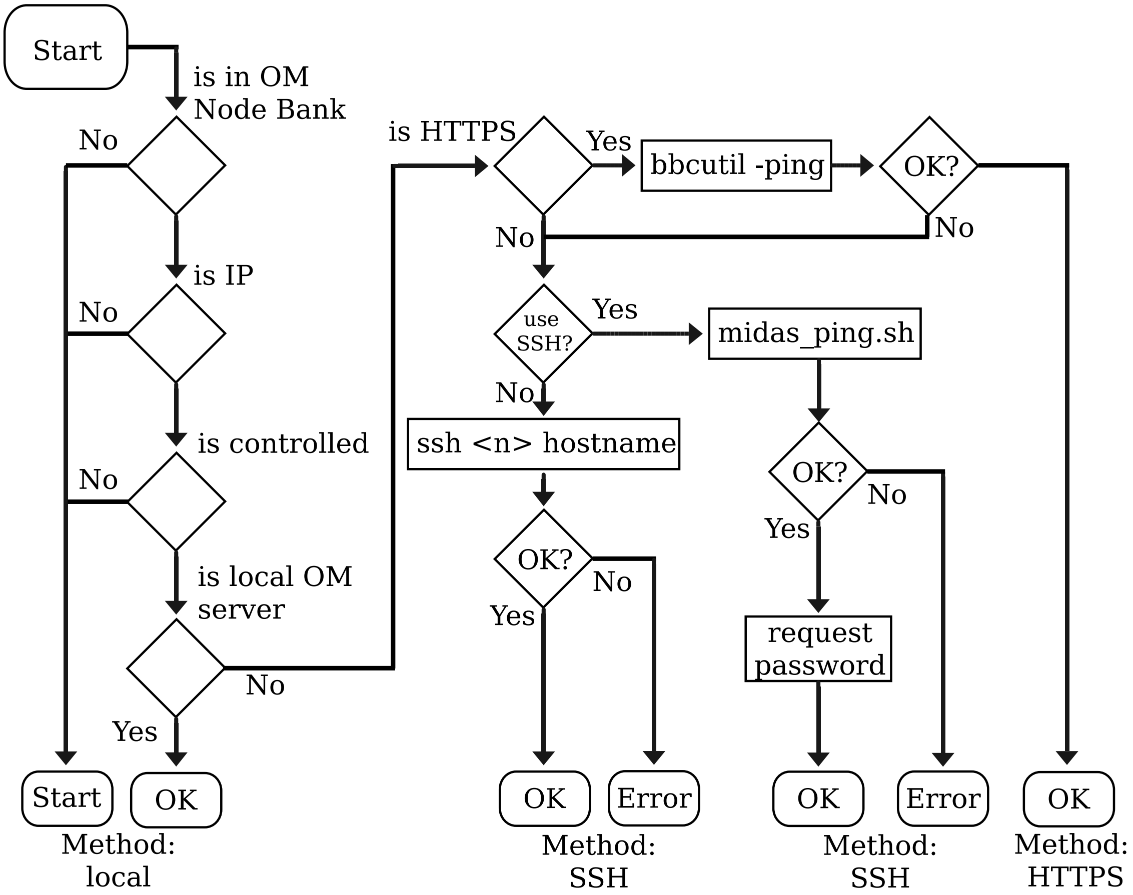
The "standard" method represents the "classic" FTP/rexec based communication, which actually has not been verified – it only remains as the only possibly method after all other options have turned out as inappropriate. This standard method actually may also involve a rhosts-based RCP.

The status and method values do not guarantee, that the installation will succeed but rather represent a best-effort prerequisites check to identify typical problems at the very beginning.

Possible values:

- "***local***" – Applies to local HPOM server only.
- "***HTTPS***" - A remote HTTPS HPOM agent could be contacted and the HPOM built-in HTTPS communication will be used.
- "***SSH***" - SSH installation is configured for this node.
- "***standard***" - Traditional FTP/rexec communication will be attempted.
- "***invalid***" - An error occurred, check “comment” column.

Figure 22 Pre-Installation Analysis Flowchart



Comment - States / Error Codes

- Possible error codes:
  - "***nosuchnode***" - Node is not found at all or not the IP.
  - "***notcontrolled***" - Node is a member of HPOM NodeBank but it is not set as “controlled”.
  - "***sshnopubkey***" - SSH installation requested, but key-based SSH failed (exit non-zero).
  - "***pingfailed***" - Node is not reachable via ICMP.
- Click on the “**?**” question mark to obtain more details.
- Not an error is:
  - "***passwdrequired***" - FTP/rexec method detected, needs password.

Details will be discussed in the next sections.

Details SSH

- SSH is purely non-interactive.
    - Public-key based access to target node must be possible!
        - There is no way to pass the password to the SSH command.
        - Key may have a passphrase. If this is the case it must be registered with SSH-add to environment of Administration UI server.
        - An automatic initial host key acceptance is possible.
- Local user is root, remote user configurable as HPOM node attribute.
- Administration UI DEBUG mode also runs SSH in verbose mode, see Administration UI logs for output.

Unlike FTP or RCP/REXEC, SSH is still considered as an acceptable communication method and is therefore particularly important as an agent installation method. Therefore, here some specific details.

There is no way to pass a password to the SSH command via command line or parameters. Because of that, the only possible authentication method is public-key based. This requires that

- The host key of the target node must be present in the SSH "known_hosts" file locally (on the HPOM server). To accomplish this, either
    - Manually execute the SSH command and confirm the host key.
    - Configure the SSH command to automatically accept new host keys. This configuration can be done globally, per user (root) or for Administration UI only. The last is recommended and explained later.
- The public key of the local (HPOM server) user "root" must be added to the SSH "authorized_keys“ file of the installation user configured for that node (usually root) on the target node.

The local key of the user root on the HPOM server may be protected by a passphrase. To make this passphrase available to the Administration UI BackEnd server process (which will effectively start the SSH command as sub-process), use the SSH-add command to register the passphrase with the SSH-agent. The Administration UI BackEnd server process must inherit some environment variables pointing it to the SSH-agent. This can be accomplished by starting an SSH-agent process during OS boot (or any time before starting the Administration UI BackEnd server) and write the printed environment variables into the file `.../ssh.env`, for example:

```
# ssh-agent -s > /opt/OV/OMU/adminUI/ssh.env
# cat /opt/OV/OMU/adminUI/ssh.env
SSH_AUTH_SOCK=/tmp/ssh-DJBhmN5478/agent.5478; export SSH_AUTH_SOCK;
SSH_AGENT_PID=5479; export SSH_AGENT_PID;
```

Then, at any time later on the command line, pass on the passphrase to the ssh-agent:

```
# ssh-add
Enter passphrase for /.root/.ssh/id_dsa:
Identity added: /.root/.ssh/id_dsa (/.root/.ssh/id_dsa)
```

In case of problems, review the following section which gives some troubleshooting instructions.

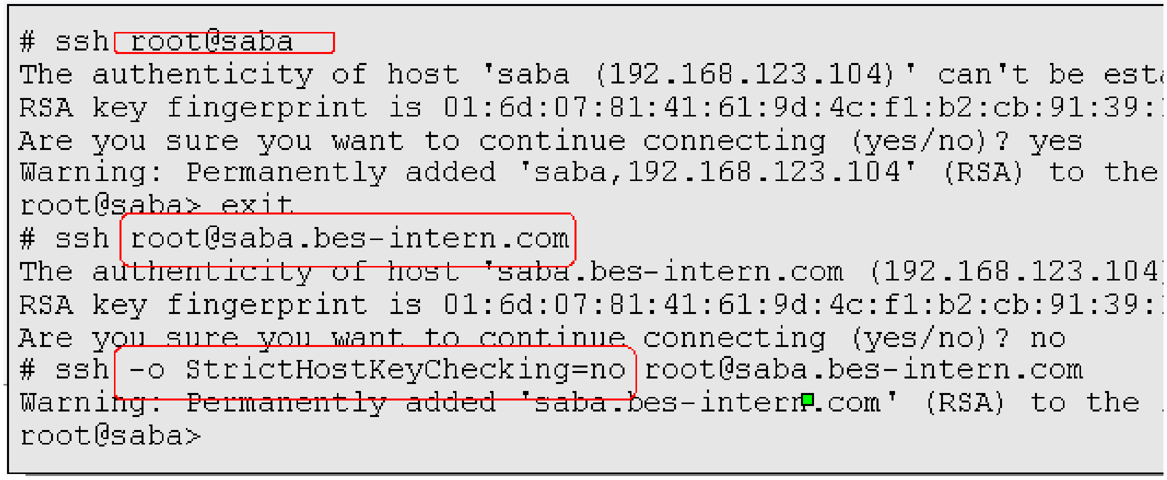
Troubleshooting SSH

Please remember the following SSH points:

- SSH key must be locally stored in ~root/.ssh/known_hosts.
  - — Associated with hostname as typed in with SSH command.
  - — Short and long host names NOT identical (neither IP address).
- Check configurable with SSH option.

SSH expects that the host key of the remote host has to be confirmed once – normally on the command line this looks as shown in the example below (Figure 23 on page 87).

Figure 23



```
# ssh root@saba
The authenticity of host 'saba (192.168.123.104)' can't be esta
RSA key fingerprint is 01:6d:07:81:41:61:9d:4c:f1:b2:cb:91:39:
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'saba,192.168.123.104' (RSA) to the
root@saba> exit
# ssh root@saba.bes-intern.com
The authenticity of host 'saba.bes-intern.com (192.168.123.104
RSA key fingerprint is 01:6d:07:81:41:61:9d:4c:f1:b2:cb:91:39:
Are you sure you want to continue connecting (yes/no)? no
# ssh -o StrictHostKeyChecking=no root@saba.bes-intern.com
Warning: Permanently added 'saba.bes-intern.com' (RSA) to the
root@saba>
```

Since Administration UI does not run interactively, nobody can confirm the host key. Thus, the host key confirmation has to be accomplished some other way. There are two possibilities:

- Somebody has to manually do a `# ssh root@targetnode` on the command line and confirm the remote host key. This has to be done once before attempting the Administration UI-based agent installation. Watch out for the problem with the host names illustrated in the example above: different host names representing the same node are not considered as identical by the SSH command!
- Administration UI can be configured to automatically accept remote host keys by always specifying the `StrictHostKeyChecking=no` option. Instructions how to configure this are presented later.

Locking

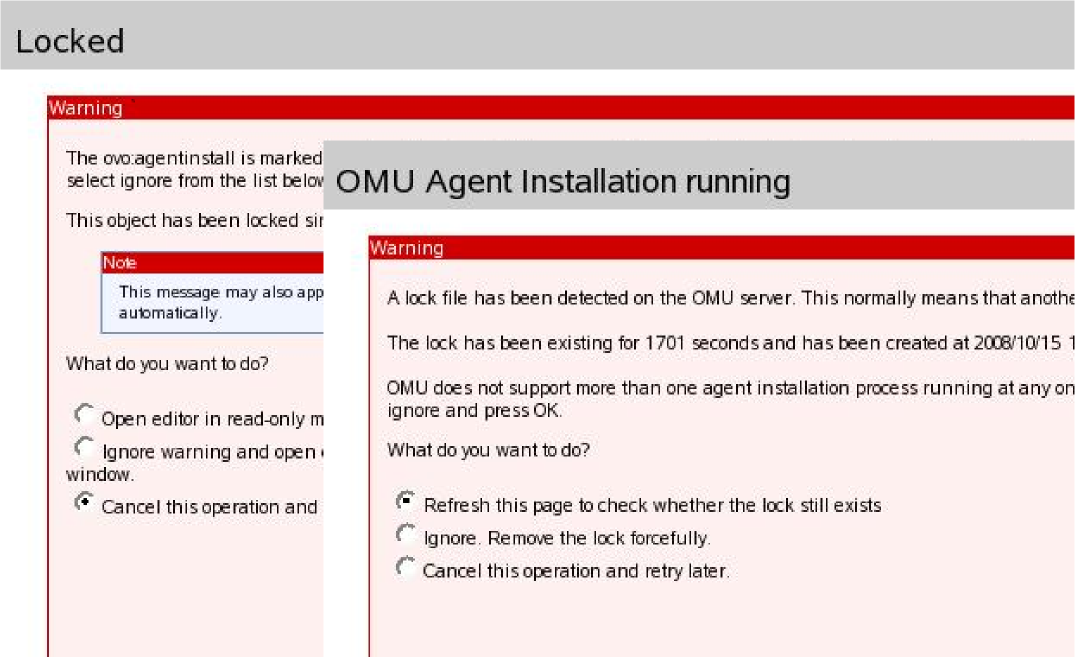
- Concurrent usage is not possible (see Figure 24 on page 88)
  - Administration UI lock protects deployment view
  - HPOM `inst.sh` locks execution of script

The regular Administration UI locking mechanism is implemented to prevent multiple users from opening the agent installation view at the same time. In addition, the HPOM `inst.sh` script maintains a lock file which guarantees, that only *one* instance of `inst.sh` executes at a time.

The HPOM lock is normally removed by `inst.sh` unless the `inst.sh` script has been killed with SIGKILL (signal 9) or has crashed – then the lock file remains and subsequent executions of `inst.sh` will fail because of the (wrong) assumption that `inst.sh` is still running. To remedy such a situation, Administration UI presents the view shown in the example below (Figure 24 on page 88) and offers the user three choices:

- **Cancel** entirely
- **Refresh** to retry testing the HPOM lock file. For example, if the `inst.sh` script has been started 5 min. ago, this option might be appropriate.
- **Ignore**/Break lock and continue. This may be appropriate if the lock is older than a certain time (lets say 1h) and/or no `inst.sh` process is running anymore – then, most likely the lock just has not been cleaned up properly.

Figure 24 Locking



Locked

Warning

The ovo:agentinstall is marked
select ignore from the list below

This object has been locked si

Note

This message may also app
automatically.

What do you want to do?

○ Open editor in read-only m
○ Ignore warning and open
window.
● Cancel this operation and

OMU Agent Installation running

Warning

A lock file has been detected on the OMU server. This normally means that anothe

The lock has been existing for 1701 seconds and has been created at 2008/10/15 1

OMU does not support more than one agent installation process running at any on
ignore and press OK.

What do you want to do?

● Refresh this page to check whether the lock still exists
○ Ignore. Remove the lock forcefully.
○ Cancel this operation and retry later.

Configuration & Tuning

To adapt the behavior of SSH and ICMP ping executed during the prerequisites analysis phase, configure the properties as shown below.

- SSH and ICMP ping are configurable. In order to do so, please edit:

```
# /opt/OV/OMU/adminUI/conf/ovoinstall.properties

[...]
# Local SSH command to check SSH access in pre-install phase:
# ovoinstall.sshCmd=ssh

# Command executed remotely by SSH:
# ovoinstall.sshCmdRem=hostname

# Local command to ping node in pre-install phase:
# ovoinstall.pingCmd=midas_ping.sh
```

The "sshCmd" property can be used to configure the exact command executed to test SSH connectivity. For example, a full path may be specified or additional options. The latter may be necessary to enable automatic host key acceptance – this can be accomplished by configuring:

```
ovoinstall.sshCmd = ssh -o StrictHostKeyChecking=no
```

The property "sshCmdRem" can be used to change the command executed remotely. By default this is hostname, but any other remote command can be configured like echo 123 or date, etc. The only requirement is that upon success, the exit code of the command must be zero.

During runtime, the actual SSH command is:

```
<sshCmd> <debug options> -l root -o BatchMode=yes <target node> <sshCmdRem>
```

where <sshCmd> and <sshCmdRem> are substituted by the properties specified above, `<debug options>` is empty normally or "-v -v -v" in DEBUG mode, and `<target node>` is the primary host name (as configured in HPOM) of the target node.

To enable ICMP please remove the # in front of this line, so it becomes:

```
ovoinstall.pingCmd=midas_ping.sh
```

- The ICMP ping behavior itself can be customized in the following file:

```
/opt/OV/OMU/adminUI/bin/midas_ping.sh
```

For example, directly exit 0 for no ping at all. The ICMP test will be done normally by executing the command `midas_ping.sh` with the target host name. The `midas_ping.sh` command is a small wrapper script to deal with platform-specific flavors of the ping command. If needed, the script can be customized to change the ping behavior. For example, if no ICMP ping is desired at all, just make the script exit zero directly.

Troubleshooting

- In the prerequisites analysis phase please:
  - Review output of pre-check command (click “?” question mark).
  - Set Administration UI to DEBUG mode (especially for SSH).
- During the agent installation phase please:
  - Review generated parameter file.
  - Review installation log file (output of `inst.sh`).
  - Run inst.sh manually.

In case of problems, perform the steps described above. Depending on the installation phase, the appropriate steps differ.

Note that the actually installation always invokes the `inst.sh` script. If the prerequisites analysis has succeeded, a parameter file will be generated and passed on to the `inst.sh` script. If the actual installation still fails, the HPOM `inst.sh` script should print diagnostic output which will be captured and can be reviewed in the installation log files.

Setting the prerequisites analysis phase to DEBUG mode can be done by editing:

```
/opt/OV/OMU/adminUI/conf/log4j.xml
```

Change the log level to `DEBUG` in the following section:

```
<logger name="com.bes.ovo.comp.install" additivity="false">
 <level value="DEBUG"/>
 <appender-ref ref="ovo"/>
</logger>
```

No restart of the Administration UI software is necessary. 30-40 sec. after the modification, DEBUG logging will be enabled. Then, after testing, review the log file

```
/opt/OV/OMU/adminUI/log/ovo.log
```

Particularly for SSH-based nodes, the SSH command executed during the pre-install phase, will run in verbose mode and generate detailed diagnostic output.

6 Troubleshooting

Overview

The information in this chapter lists some of the problems that you might face and explains how you can use the available tools to access the information you need to start the process of fixing the problems that occur. This section covers the following topics:

- General Procedures on page 92
- Display-Related Problems on page 93
- Using Log Files on page 94
- Viewing Raw XML Data on page 96
- Troubleshooting Commands on page 98
- Communication Problems on page 106
- Authentication Problems on page 107
- Checking Process Status on page 110

General Procedures

In general, whenever experiencing problems, think about the following (some of that appears obvious but sometimes it may be helpful to remember the obvious things):

- Describe the problem as precise as possible:
  - Error reports such as “Cannot edit a policy” are not very helpful.
  - Provide screenshots, shell output data, log files, or the policy!
  - Provide a support.zip file (see below for details).
- Think about differences!
  - to other instances (“It works for all items, except this”). What is special about the non-working item?
  - “Yesterday it used to work”. So, what has changed since then?
- Try to determine whether the problem is in HPOM or in Administration UI:
  - Can you perform the same operation using native HPOM tools?

    `opcragt, opchbp, ...`
  - Are there any related entries in the HPOM error log files?

    HPOM >= 8.x: `/var/opt/OV/log/System.txt`
  - Use HPOM tracing. Most Administration UI operations are performed by calling the HPOM API. This can be traced using the regular HPOM tracing facility. The trace area of most interest is opc.api (HPOM 8 XPL tracing).

    Possibly it may be needed to restart the Administration UI application to be traced.
  - Review the Oracle database. All HPOM configuration data exists in the central Oracle database. The sqlplus command may yield hints about how the data looks like in the database.

    DO NOT modify anything this way!
- Verify the OS resources:
  - Insufficient disk space or full kernel tables are common causes. Such problems may also be logged in syslog (Unix).

Also, most efficient troubleshooting strategies involve:

- First, perform simple steps to positively include or exclude the most likely causes.
  - Commands, which can be typed in quickly and provide fast results:

    These are especially “`adminui clean`” & “`adminui analyze`”.
  - Error log files.
- More powerful capabilities like tracing usually involve more effort (configuring trace level, possibly restart services, review trace data, ...).
- If you have some suspicion, try to be certain (also determining, that something is NOT the problem, may help).

Always supply the support.zip file (see below: Packing up Support Data on page 99) to the support team.

Display-Related Problems

Since this is a web-based tool that uses a browser to connect to a server, it is unlikely that you will encounter problems displaying the GUI. However, there are one or two exceptions, notably on UNIX operating systems, where you might need to investigate further, for example:

- You are trying to re-direct the display of the web browser between UNIX hosts.
- You are installing the software on a UNIX host.

If you encounter display-related problems when trying to re-direct the display from one UNIX host to another or when installing the Administration UI software on a UNIX machine, check the following things, first:

- xhost settings:

  You need to allow remote X access on the host where the GUI is supposed to appear. To configure X access on a UNIX host, use the `xhost +` command.
- The DISPLAY variable:

  You need to set the DISPLAY variable on the UNIX host where the program that starts the display is running. Use the `export DISPLAY=<hostname>:0` command to set the display.

If the display settings are correctly set and the problem persists, check the following known problems:

- The LANG environment variable:

  A missing or incorrectly set language variable occasionally produces the following (or similar) error message when running a command on HP-UX systems:

  ```
  Warning: Missing charsets in String to FontSet conversion
  Warning: Unable to load any usable fontset
  ```

  If the language environment variable is either not (or incorrectly) set, try setting the language variable as part of the command:

  ```
  # LANG=C.iso88591 <the failing command>
  ```

Using Log Files

Administration UI writes detailed information about run-time operations to a number of different platform- and adapter-specific log files. The name of the log file and the information it contains varies according to the adapter writing the log file. By default, Administration UI stores server log files in the following location:

```
/opt/OV/OMU/adminUI/logs/*.log
```

The Administration UI WebApp component writes information to its own log files, which it stores in the following location:

```
/opt/OV/OMU/adminUI/webapps/<comp>/work/webapp/WEB-INF/logs/*.log
```

In the example above, the default path to the log files written by the Administration UI Web Application, <comp> stands for either one of the following values:

- `midas`

  The name of the logical Administration UI WebApp component
- `exist`

  The name of the built-in XML database

By default, Administration UI logs information about errors that occur during normal operation. However, if you are troubleshooting a particular problem and require more detailed, you can increase the log level. This can be done for each individual component for which you require more (or less) details. For information about the trace levels that are allowed and instructions on how to set the trace level, see chapter Log Files and Trace Levels on page 56.

The list below displays a list of the server log files created in a default set-up. Some log files belong to a specific adapter. If you are troubleshooting a problem related to a specific adapter, have a look at the log files that the adapter writes. For example, if you are investigating a problem relating to HPOM operations (e.g. list all policies, nodes is not working) which require Oracle access using the opc_op user, check the ovo.log first.

When analyzing startup problems the suggested route is: wrapper.log -> servicemix.log -> ovo.log -> midas.log.

File or directory	Purpose
access.log	Access log (IP and which pages were accessed)
agent/	Agent installation logs
ant.log	Log for the internal ant tasks
audit/	Directory holding daily auditing log files (no rollback, no cleanup). With log level INFO one line per transaction. With log level DEBUG everything is logged.
backend.log	9661 connector, only logged to with DEBUG enabled
dead.log	All illegal requests are logged here
events/	not used
exist.log	exist database
file.log	not used

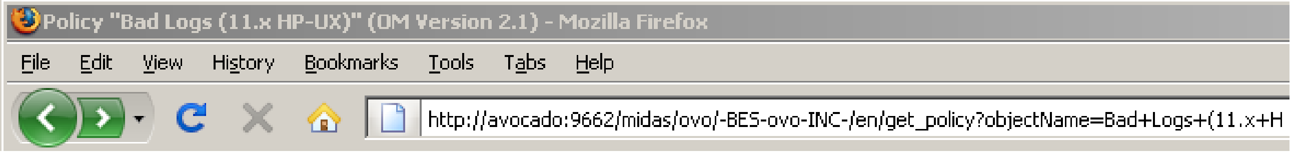
File or directory	**Purpose**
license.log	not used
lock.log	Log showing if a lock has occurred on an configuration item (WebApp module)
memory.log	memory consumption
midas.log	default log if no other special log exists
nnm.log	not used
ovo.log	Logging for HPOM and Oracle (opc_op)
ovoadmin.log	not used
package.log	not used
performance.log	for support purposes
request.log	for support purposes
requests	Directory with the request logs
results/	not used
search.log	not used
servicemix.log	if an adapter doesn't start etc check this log first
sync.log	not used
task	Directory containing individual log files written during the execution of tasks (commands, downloads)
task.log	general log whether a task has been run
threadinfo.log	for support purposes
usermgmt.log	User log when AD or LDAP is used for user
vcs.log	not used
velocity.log	used internally
web.log	WebApp log
wrapper.log	log for wrapper module
xmldb.log	XML DB database log

Viewing Raw XML Data

In case of problems displaying data correctly in the web browser, the raw XML data can be displayed as well. Displaying raw XML data can help you to determine whether the actual data is working or not or if the presentation in the Administration UI isn't simply working (stylesheets).

In order to do this the default browser URL needs to be modified.

Figure 25 Unmodified URL

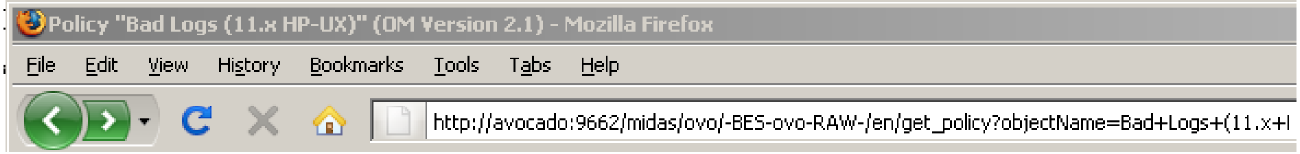


Policy "Bad Logs (11.x HP-UX)" (OM Version 2.1) - Mozilla Firefox

http://avocado:9662/midas/ovo/-BES-ovo-INC-/en/get_policy?objectName=Bad+Logs+(11.x+H

To view raw XML data, modify the standard URL as follows:
The string “`.../-BES-ovo-INC-/...`” (see Figure 25 on page 96) needs to be **modified and changed to** “`.../-BES-ovo-RAW-/...`”, so it looks like (Figure 26 on page 96) this:

Figure 26 Modified URL 1

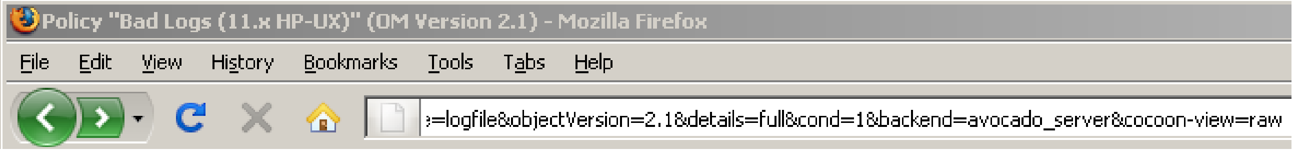


Policy "Bad Logs (11.x HP-UX)" (OM Version 2.1) - Mozilla Firefox

http://avocado:9662/midas/ovo/-BES-ovo-RAW-/en/get_policy?objectName=Bad+Logs+(11.x+I

Furthermore at the end of the URL add “`&cocoon-view=raw`”, see Figure 27 on page 96:

Figure 27 Modified URL 2

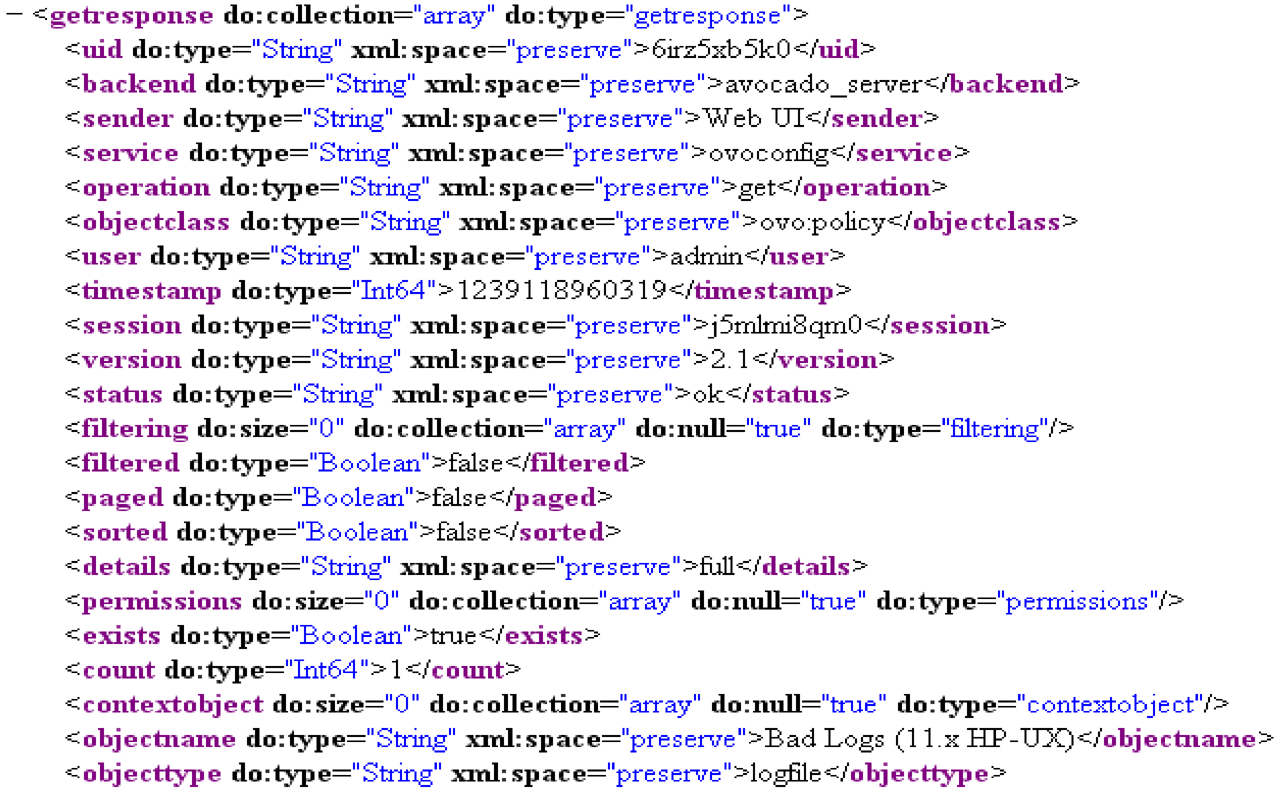


Policy "Bad Logs (11.x HP-UX)" (OM Version 2.1) - Mozilla Firefox

:=logfile&objectVersion=2.1&details=full&cond=1&backend=avocado_server&cocoon-view=raw

Enabling raw XML-data output produces output similar to that illustrated in Figure 28 on page 97: Raw XML Data Output.

Figure 28 Raw XML Data Output



```
- <getresponse do:collection="array" do:type="getresponse">
    <uid do:type="String" xml:space="preserve">6irz5xb5k0</uid>
    <backend do:type="String" xml:space="preserve">avocado_server</backend>
    <sender do:type="String" xml:space="preserve">Web UI</sender>
    <service do:type="String" xml:space="preserve">ovoconfig</service>
    <operation do:type="String" xml:space="preserve">get</operation>
    <objectclass do:type="String" xml:space="preserve">ovo:policy</objectclass>
    <user do:type="String" xml:space="preserve">admin</user>
    <timestamp do:type="Int64">1239118960319</timestamp>
    <session do:type="String" xml:space="preserve">j5mlmi8qm0</session>
    <version do:type="String" xml:space="preserve">2.1</version>
    <status do:type="String" xml:space="preserve">ok</status>
    <filtering do:size="0" do:collection="array" do:null="true" do:type="filtering"/>
    <filtered do:type="Boolean">false</filtered>
    <paged do:type="Boolean">false</paged>
    <sorted do:type="Boolean">false</sorted>
    <details do:type="String" xml:space="preserve">full</details>
    <permissions do:size="0" do:collection="array" do:null="true" do:type="permissions"/>
    <exists do:type="Boolean">true</exists>
    <count do:type="Int64">1</count>
    <contextobject do:size="0" do:collection="array" do:null="true" do:type="contextobject"/>
    <objectname do:type="String" xml:space="preserve">Bad Logs (11.x HP-UX)</objectname>
    <objecttype do:type="String" xml:space="preserve">logfile</objecttype>
```

Such a result would indicate that the query itself is working. Therefore, if the result in the graphical web interface isn't returning a proper result and webpage this would indicate that perhaps there is a problem with one of the stylesheets which are used to render the web interface or result output.

Troubleshooting Commands

This section describes advanced options for the `./adminui` command. Please note, that you do not need these advanced options for normal, every-day maintenance. However, in troubleshooting situations, these commands might be helpful. Check with product support before using them. The information in this section covers the following topics:

- Packing up Support Data on page 99
- Clean Restart on page 100
- Running ANT Tasks on page 100
- Accessing the XML Database on page 101
- Checking Component Status on page 104
- Re-initialize the XML Database on page 105

Packing up Support Data

The support sub-command is an important support tool. Use it if you run into any problems with Administration UI and want to contact Support. With the support sub-command it is possible to quickly collect all required log & configuration files. The file location will be inside

```
/opt/OV/OMU/adminUI/
```

Example:

```
# /opt/OV/OMU/adminUI/adminui support
```

The shell output should look like this:

```
[root@deli:/opt/OV/OMU/adminui] ./adminui support
[...]
support.zip:
     [echo] collecting support information ...
     [echo] collecting version info ...
     [echo] collecting installed files ...
     [echo] collecting Java properties ...
[propertyfile] Creating new property file:
[...]
intern.checksum_check:
     [echo] checking checksums ...
     [echo] creating support zip ...
      [zip] Building zip: /opt/OV/OMU/adminUI/
support_20090325162209.zip
[echo] cleaning up ...
[echo] send the file /opt/OV/OMU/adminUI/support_20090325162209.zip
to support
BUILD SUCCESSFUL
Total time: 2 minutes 30 seconds
```

At the end of the output you see the support filename's name and location.

The file name will contain the date and time when the command has been executed.

The zip file contains:

- All core configuration files from `/opt/OV/OMU/adminUI/conf`
- All core log files from `/opt/OV/OMU/adminUI/logs`
- All WebApp component log and configuration files from `/opt/OV/OMU/adminUI/webapp/midas/work/webapp/WEB-INF`
- Furthermore some environment variables are collected. Example: output from `./adminui analyze`, `uname -a`, `listener.ora`, etc., information helping Support troubleshooting the reported problem.

The product support team will usually ask you to send this file via email as a first step to investigate a problem.

Clean Restart

Quite often the clean command is helpful in solving any existing problems, especially when a file corruption exists that prevent one or more modules to start up successfully.
Solution: Perform a “clean” restart of the application itself, using the following command:

```
# /opt/OV/OMU/adminUI/adminui clean
# /opt/OV/OMU/adminUI/adminui start
```

This will restart the application performing a cleanup of all log and run-time files, forcing the application to unpack all necessary run-time files again.

> Please make sure no other users are logged in and working inside the Administration UI before you restart the application like this! Otherwise their current work might be lost.

Running ANT Tasks

Using the `ant` sub-command, integrated ANT tasks can be started similarly to what happens inside Administration UI generally.

However, none of these tasks are for normal customer use. If necessary Support will supply the correct syntax and how-to information.

Accessing the XML Database

Generally it should not be necessary in a day-to-day operation to directly access the XML database. This is generally only needed if e.g. all users, user groups or user roles need to be downloaded in the raw format.

It is strongly recommended not to modify anything unless instructed by Support. Two possibilities exist to access the XML database of Administration UI:

- using the embedded console (not recommended)
- using the HTTP interface via your standard browser

Embedded Console

The `xmldb` sub-command starts a graphical XML DB management console.

> If used remotely, please export the X DISPLAY of the HPOM server to your own workstation before using the `xmldb` sub-command.

After running

```
# /opt/OV/OMU/adminUI/adminui xmldb
```

a screen appears as illustrated in Figure 29 on page 101: XML Database Login Screen

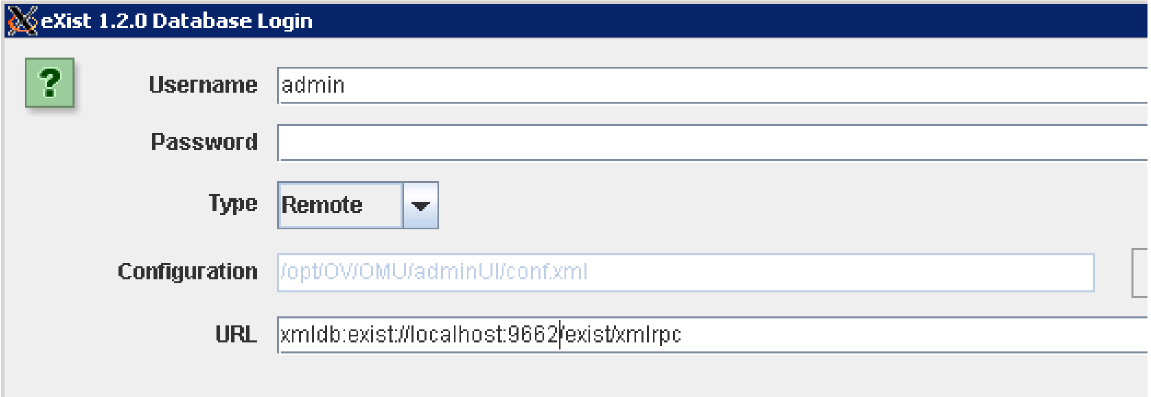
Specify the correct URL (use localhost if started on the HPOM system itself and the correct port; 9662 by default). Example:

```
xmldb:exist://localhost:9662/exist/xmlrpc
```

The default access information is:

- Username: `admin`
- Password: `admin`

Figure 29 XML Database Login Screen



eXist 1.2.0 Database Login

Username: admin

Password:

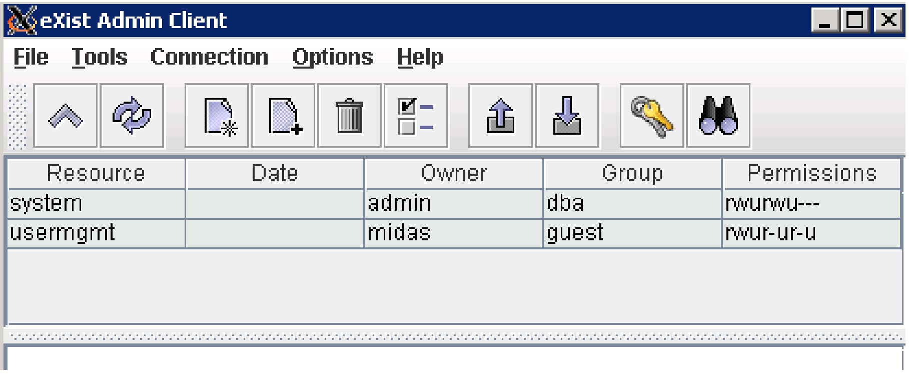
Type: Remote

Configuration: /opt/OV/OMU/adminUI/conf.xml

URL: xmldb:exist://localhost:9662/exist/xmlrpc

After a successful login the actual GUI appears (Figure 30 on page 102):

Figure 30 XML Console



Resource	Date	Owner	Group	Permissions
system		admin	dba	rwurwu---
usermgmt		midas	guest	rwur-ur-u

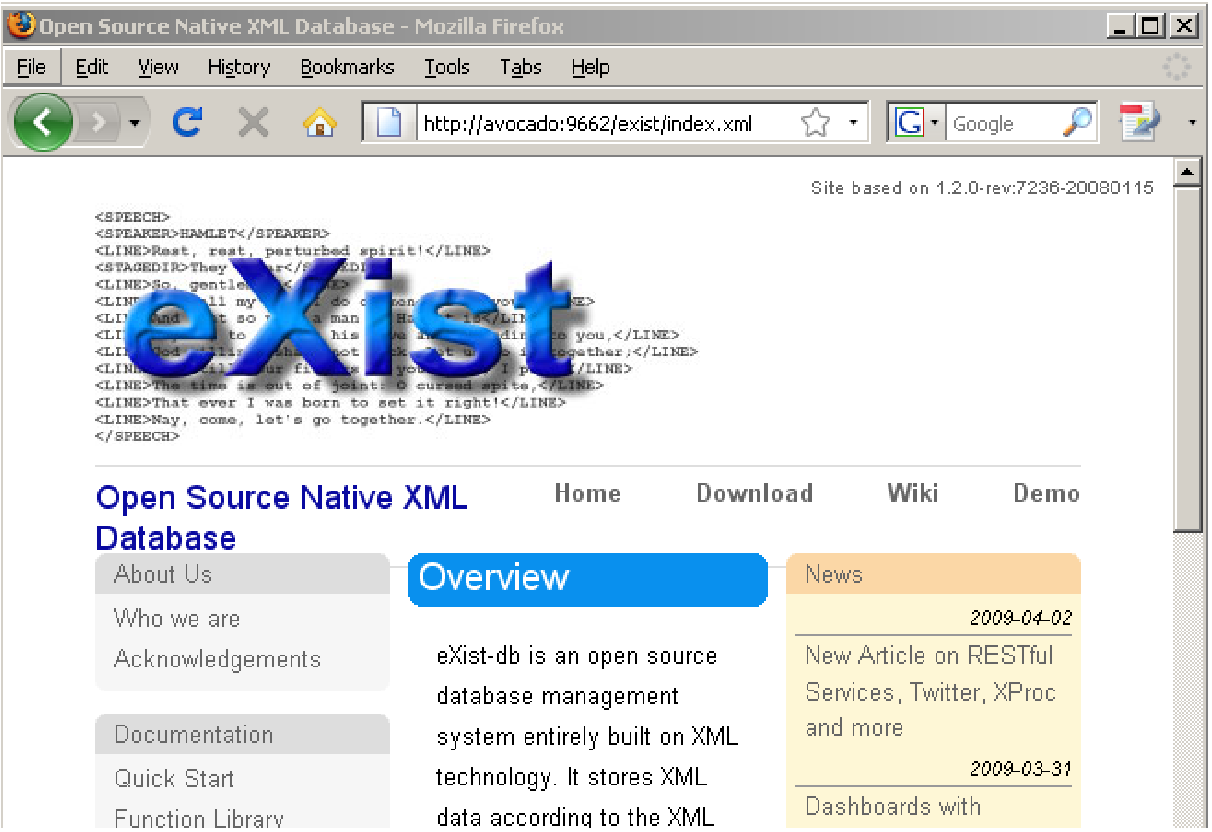
HTTP Access Using a Standard Browser

The XML DB also supports an HTML interface which can be accessed using the standard web browser. It is only necessary to add to the normal Administration UI URL `exist` at the end, so the URL looks like this:

```
http://<HP-OMU-address>:9662/exist
```

A screen similar to the one displayed in Figure 31 on page 102:

Figure 31 XML Database HTML Interface


Open Source Native XML Database - Mozilla Firefox
File
Edit
View
History
Bookmarks
Tools
Tabs
Help
Google
eXist
<SPEECH>
<SPEAKER>HAMLET</SPEAKER>
<LINE>Rest, rest, perturbed spirit!</LINE>
<LINE>The time is out of joint: O cursed spite,</LINE>
<LINE>That ever I was born to set it right!</LINE>
<LINE>Nay, come, let's go together.</LINE>
</SPEECH>
Home
Download
Wiki
Demo
About Us
Who we are
Acknowledgements
Documentation
Quick Start
Function Library
Overview
News

http://avocado:9662/exist/index.xml

Site based on 1.2.0-rev:7236-20080115

Open Source Native XML Database

eXist-db is an open source database management system entirely built on XML technology. It stores XML data according to the XML

2009-04-02

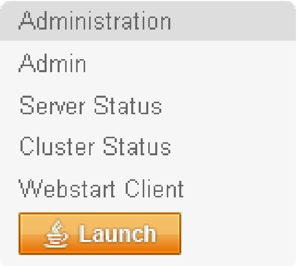
New Article on RESTful Services, Twitter, XProc and more

2009-03-31

Dashboards with

In order to log into the XML database please navigate on the left hand side to the section “Administration” and select “Admin”, see Figure 32 on page 103

Figure 32

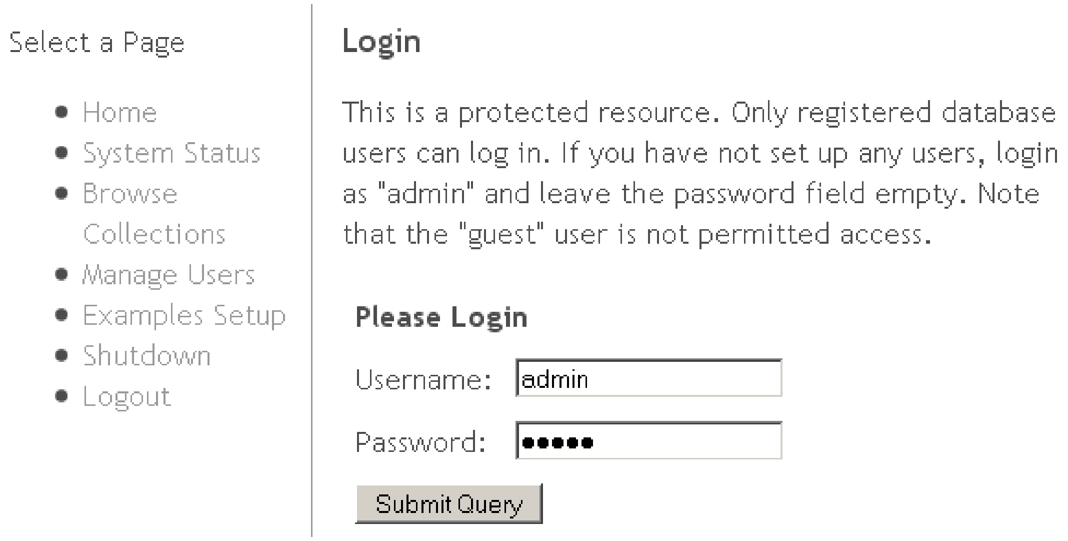


This will bring you to the actual login screen, see below Figure 33 on page 103.

The default login information is:

- Username: `admin`
- Password: `admin`

Figure 33 HTML Interface Login



Login

This is a protected resource. Only registered database users can log in. If you have not set up any users, login as "admin" and leave the password field empty. Note that the "guest" user is not permitted access.

Please Login

Username: admin

Password: ●●●●●

Again, if access to the XML database and any data is required by Support, a step-by-step instruction will be provided.

For further details, see: http://exist.sourceforge.net

Checking Component Status

The `servicemix` sub-command displays information about installed Servicemix JBI components (binding components and services) and deployed service assemblies with their contained service units during runtime.

With this command it can be checked whether all required Administration UI adaptors have been successfully started. Missing service assemblies indicate a problem with the corresponding adapter. In this case review the log files servicemix.log and midas.log for more details about failed adapters.

This sub-command is not needed during normal day-to-day operation but can be used for extended troubleshooting purposes. It is absolutely non-destructive and can be used without damaging anything. It is recommend to re-direct the output to a temporary file.

```
# /opt/OV/OMU/adminUI/adminui servicemix > /tmp/sm.status
```

The following example shows parts of the output of the `servicemix` sub-command:

```
servicemix:
list-binding-components:
[echo]  list-binding-components
[echo]  Prints information about the binding components installed in
servicemix.
[echo]     host=avocado
[echo]     port=9660
[...]
list-service-engines:
[echo]  list-service-engines
[echo]  Prints information about all of the Service Engines in
Servicemix.
[echo]     host=avocado
[echo]     port=9660
[...]
[jbi:list-service-engines]<component-info type='service-engine'
name='servicemix-eip' state='Started'>
[jbi:list-service-engines]   <component-info type='service-engine'
name='servicemix-lwcontainer' state='Started'>
[jbi:list-service-engines]   <component-info type='service-engine'
name='servicemix-script' state='Started'>
[...]
list-service-assemblies:
[echo]  list-service-assemblies
[echo]  list deployed Service Assemblies in Servicemix.
[echo]     host=avocado
[echo]     port=9660
[...]
[jbi:list-service-assemblies]<service-unit-info name=
'midas-ovoconfig' state='Started'
deployed-on='servicemix-lwcontainer'>
[jbi:list-service-assemblies]<description>MIDAS HPOM for UNIX
Configuration Adaptor</description>
[...]
```

The first blocks **list-binding-components** and **list-service-engines** must be always present including some additional details.

The last block **list-service-assemblies** shows all deployed service assemblies. This list must match the set of files in `/opt/OV/OMU/adminUI/deploy`. If a service assembly is present in the deploy directory but not listed in the output of

```
# /opt/OV/OMU/adminUI/adminui servicemix
```

as **Started**, it has failed to start. In this case inspect the log files for details. Particularly the file

```
/opt/OV/OMU/adminUI/logs/servicemix.log
```

will contain entries like the following:

```
ERROR - 2009-02-20 13:03:16,369 |
AutoDeploymentService.updateArchive(308) | Failed to update Service
Assembly: midas-wapam
java.lang.Exception: <?xml version="1.0" encoding="UTF-8"?>
[...]
nested exception is java.lang.UnsatisfiedLinkError: no jpam in
java.library.path</loc-message>
[...]
```

In this example, the adapter midas-wapam (the PAM authentication adapter) has failed to start because the native library libjpam.so could not be found.

Re-initialize the XML Database

The `init` sub-command clears and re-initializes the XML DB.

For details please see Re-initialization of XML DB on page 40

Communication Problems

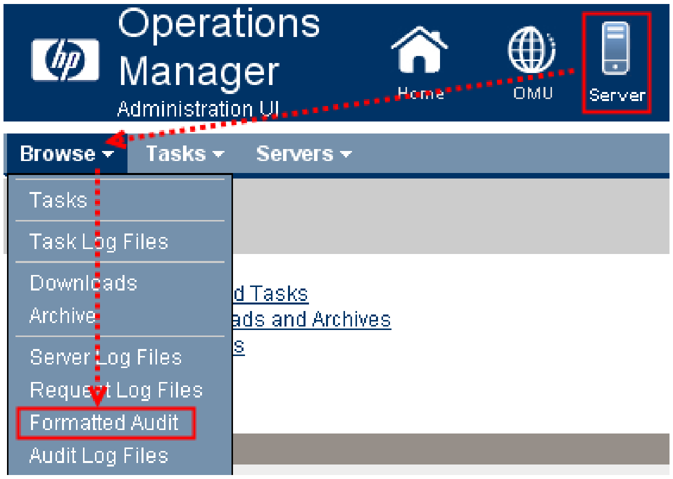
This section describes how to investigate problems relating to inter-component communication. For example, you can learn how to perform basic checks to ensure that name resolution works correctly or ensure more advanced features relating to HTTPS configuration work as expected. The information in this section covers the following areas:

- General Communication Problems on page 106

General Communication Problems

Always make sure first that general network connectivity exists between the involved systems. Since Administration UI is installed on the HPOM system, end-users must be able to reach and access the HPOM system from their workstations using their web browsers on the correct Web Application UI ports (default 9662 for http:// and 9663 for https:// requests), Figure 34 on page 106:

Figure 34



It is quite common that the end-user will be in a different network than the HPOM server is in. If a firewall exists the necessary ports of the Administration UI WebApp component must be open. Also must the hostname of the HPOM system be resolvable within the end-users network.

- Check whether name resolution works correctly:

  ```
  # nslookup <target-node>
  ```

  Make sure this yields the same results on both sides (unless there is a NAT router in between).

- Ping the target system

  ```
  # ping <target-node>
  ```

- Check the Administration UI WebApp port if it can be reached:

  ```
  # telnet <target-node> <port>
  ```

  for example:

  ```
  # telnet ios 9662
  Trying 192.168.123.113...
  ```

```
Connected to ios.
Escape character is '^]'.
^]
telnet> quit
Connection closed.
```

There will be no real communication possible but the telnet command must be able to at least establish a connection.

Authentication Problems

If you or another user cannot log into the Administration UI web interface, please check the following list:

- Is the Administration UI running on the desired system and does it use the correct ports? Does the web interface show up?

  Verify whether the MIDAS processes are running and the port configuration. Check whether the ports are allocated using the `netstat` or `lsof` commands.

  Also, if possible, start a web browser locally on the WebApp system and try to connect.

- Can you reach the HPOM system from the basic network perspective?

  Use the ping and telnet commands.

If the Administration UI web interface is displayed but the user login fails, it is possible to check:

- If the Administration UI has been restarted a few moments ago, all components might no be fully running yet. Although you can see the login web interface, the Administration UI XML user database will generally take longer to start.
  Solution: Therefore, please wait for another 30-120sec before you try to login again.

- Another possibility is that the user is in no user group or that this user group does not have a user role inside Administration UI assigned. See the User Guide for details.
  Solution: try another user login, e.g. the main Administration UI user “admin” in order to validate if the login process is generally broken or not.

- In case the previous tips do not help a file corruption might exist, preventing some modules to start up successfully.
  Solution: Perform a “clean” restart of the application itself, using the following command:

  ```
  # /opt/OV/OMU/adminUI/adminui clean
  # /opt/OV/OMU/adminUI/adminui start
  ```

  This will restart the application performing a cleanup of all log and run-time files, forcing the application to unpack all necessary run-time files again.

Please make sure no other users are logged in and working inside the Administration UI before you restart the application like this! Otherwise their current work might be lost.

- If a “clean” restart also does not help it might be possible that a rare case of corruption of the XML database exists.
  Solution: Please create a backup of the Administration UI configuration including the XML DB, using

```
# /opt/OV/OMU/adminUI/adminui save
```

Additionally also create a “support zip” containing all Administration UI configuration and log files. Use the following command:

```
 # /opt/OV/OMU/adminUI/adminui support
```

Please forward both zip file packages to Support explaining your problems so they can analyze the XML DB files for any problems.

Note: Details on the save and support command can be found here:
- Save Configuration - “./adminui save|reload” on page 28
- Collect Support Information - “./adminui support” on page 31

PAM Integration

For the PAM setup please see section: PAM integration on page 112

If experiencing problems with a configured PAM module, check the following:

- Are there any related entries in the log file

  `/opt/OV/OMU/adminUI/logs/usermgmt.log`

for example:

```
DEBUG - 2006-12-20 14:03:49,002 |
UserModelRequestTransformer.transform(?)|rewriting request to service
pam
DEBUG - 2006-12-20 14:03:49,087 | PamServer.authenticate(?) |
Authenticating user admin with PAM service midas ...
DEBUG - 2006-12-20 14:03:49,088 | Pam.authenticate(160)| Debug mode
active.
ERROR - 2006-12-20 14:03:51,126 | PamServer.authenticate(?) |
Authentication of user admin failed: Underlying authentication
service can not retrieve authentication information.
DEBUG - 2006-12-20 14:03:51,163 | UserMgmtFilter.onMessageExchange(?)
| clearing response { DOType: errorresponse
extra : com.bes.itm.comp.usermgmt.AuthenticationFailedException:
Could not authenticate user admin via PAM. PAM error: Underlying
authentication service can not retrieve authentication information.
```

- To enable additional tracing in the PAM adapter module, configure the following in `/opt/OV/OMU/adminUI/conf/log4j.xml`:

```
<logger name="net.sf.jpam" additivity="false">
 <level value="DEBUG"/>
 <appender-ref ref="usermgmt"/>
</logger>
```

This will make the PAM module write additional debug statements into the log file `/opt/OV/OMU/adminUI/logs/usermgmt.log`.

- Test the authentication method standalone

```
# cd /opt/OV/OMU/adminUI/
# export SHLIB_PATH=$SHLIB_PATH:./lib/midas
# ./jre/bin/java -cp ./lib/cli/midas_cli.jar:./work/
service-assemblies/midas-wapam/version_1/sus/servicemix-lwcontainer/
midas-pam/lib/jpam-0.5.jar:../lib/commons-logging-1.1.jar com/bes/
itm/comp/usermgmt/TestPam <user name> <password>
**** Starting ...
```

```
**** Authenticating user admin ...
**** Authentication done.
**** Success: false
**** Result: Underlying authentication service can not retrieve
authentication information.
**** Exit.
```

This test class performs a pure PAM authentication of <user name> with <password> and prints the results to stdout.

- Check the PAM configuration as described in chapter Authentication Software.
- Make sure that all dependencies of the native library `/opt/OV/OMU/adminUI/lib/midas/libjpam.so` are satisfied, for example:

```
# ldd < _HOME>/lib/midas/libjpam.so
  linux-gate.so.1 =>  (0xffffe000)
  libpam.so.0 => /lib/libpam.so.0 (0x40016000)
  libpam_misc.so.0 => /lib/libpam_misc.so.0 (0x40020000)
  libdl.so.2 => /lib/libdl.so.2 (0x40023000)
  libc.so.6 => /lib/tls/libc.so.6 (0x40027000)
  /lib/ld-linux.so.2 (0x80000000)
```

- Review the UNIX syslog log file, the native PAM library logs messages to syslog, for example:

```
# grep -i pam /var/log/messages
[...]
Dec  1 18:15:41 garlic midas.pam(pam_unix)[25305]: authentication
failure; logname= uid=0 euid=0 tty= ruser= rhost=  user=admin
Dec  1 18:17:40 garlic midas.pam(pam_unix)[25305]: authentication
failure; logname= uid=0 euid=0 tty= ruser= rhost=  user=admin
[...]
```

- To enable tracing the native PAM library, turn on debugging on syslog level (the steps needed may vary for the different UNIX flavors). The related service name is auth. To turn on tracing, configure syslogd as in the following example:

```
# touch /etc/pam_debug
# vi /etc/syslog.conf
auth.debug/tmp/pam_auth.log
[...]
```

  Make the `syslogd` process re-read its configuration, for example by executing:

```
# kill -HUP `cat /var/run/syslog.pid`
```

The resulting debug output looks as in the following example:

```
# tail -f /tmp/pam_auth.log
Dec 20 15:41:16 ios PAM: pam_start(midas admin)
Dec 20 15:41:16 ios PAM: pam_set_item(1)
Dec 20 15:41:16 ios PAM: pam_set_item(2)
Dec 20 15:41:16 ios PAM: pam_set_item(5)
Dec 20 15:41:16 ios PAM: pam_set_item(6)
Dec 20 15:41:16 ios PAM: pam_authenticate()
Dec 20 15:41:16 ios PAM: load_modules: /usr/lib/security/hpux32/
libpam_unix.so.1
Dec 20 15:41:16 ios PAM: load_function: successful load of
pam_sm_authenticate
```

```
Dec 20 15:41:16 ios PAM: pam_get_username(ux)
Dec 20 15:41:16 ios PAM: pam_mapping_in_use()
Dec 20 15:41:16 ios PAM: pam_set_item(6)
Dec 20 15:41:16 ios PAM: pam_acct_mgmt()
Dec 20 15:41:16 ios PAM: load_modules: /usr/lib/security/hpux32/
libpam_unix.so.1
Dec 20 15:41:16 ios PAM: load_function: successful load of
pam_sm_acct_mgmt
Dec 20 15:41:16 ios PAM: pam_get_username(ux)
Dec 20 15:41:16 ios PAM: pam_mapping_in_use()
Dec 20 15:41:16 ios PAM: pam_end(): status = Success
```

Checking Process Status

If the JRE process running Administration UI server crashes, collect the following data and send it to product support:

- The support.zip generated as described above
- The related `hs_err_pid<PID>.log` file

If the crash occurs regularly, also the core file written by the JRE may help. By default, creating core files on UNIX is disabled. To enable the creation of core files, edit:

```
# /opt/OV/OMU/adminUI/bin/server.sh
```

Please comment out the following line:

```
ulimit -c 0
```

Then, if the abort occurs again, save the core file for later evaluation. If a core file exists, it can be analyzed using the HPOM utility stacktrace, as shown in the following example (this example applies to an HP-UX Itanium system):

```
# /opt/OV/contrib/OpC/stacktrace /opt/midas31/core
                                 /opt/midas31/jre/bin/IA64N/java
```

This may particularly yield precise information where the problem occurred.

7 External Software

Overview

This section lists additional, external software products that are integrated in Administration UI and describes how you can configure the software to suit the demands of your environment. All the software products described in this section are optional unless you choose to install and configure functionality on which a Administration UI feature depends:

- Authentication Software on page 112

- DST – Daylight Saving Time Patches on page 117

Authentication Software

Authentication of Administration UI users happens inside the Administration UI WebApp server part, to which the user’s web browser connects.

With the current product version, Administration UI supports authentication using LDAP, LDAPS, Active Directory Server or any authentication service which can be integrated into PAM.

NOTE: Authentication covers only the process of validating the user account with a password. It does not include any authorization control (user's capabilities). Authorization is implemented exclusively in Administration UI by defining Administration UI user roles.

Therefore, whenever setting up a new Administration UI user, make sure that the account exists in both Administration UI and the external authentication system. Furthermore, make sure that the Administration UI user is member of at least one Administration UI group which has at least one Administration UI user role assigned.

To use an external authentication software like LDAP or PAM in Administration UI additional software (for example, the LDAP server) may be required on the Administration UI WebApp. Install and configure this software as needed. More details are presented below.

PAM integration

To authenticate Administration UI users through PAM (Pluggable Authentication Modules), no extra software is needed. Administration UI already includes the open-source module jpam (see `http://jpam.sourceforge.net` for details).

However, PAM is just an interface linking software providing authentication services (like LDAP, Kerberos, UNIX passwd) to consumer applications like Administration UI. Therefore, possibly software modules implementing the actual authentication service may be needed.

To configure PAM, perform the following steps:

1.Decide, which authentication method to use. If needed, install required software modules and configure them. Test the authentication service standalone, i.e. outside of the Administration UI context.

2.Configure all Administration UI user accounts in the authentication service.

3.Configure PAM to route Administration UI authentication requests to the desired authentication service. The PAM service name is **midas**.

4.Activate the external authentication service in the conf/auth.properties file:

```
# vi /opt/OV/OMU/adminUI/conf/auth.properties
# configuration properties for authentication and authorization
components
#auth-filter.enabled=false
caches.timeout=7200000
usermodel-router.authResource=file:conf/auth.xml
# eof
```

5.Switch Administration UI to PAM authentication by configuring the property in the `conf/auth.xml` file. The file has to look like this:

```
# vi /opt/OV/OMU/adminUI/conf/auth.xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN" "http://
www.springframework.org/dtd/spring-beans.dtd">
<beans>

  <bean id="targetServices" class="java.util.ArrayList">
    <constructor-arg>
      <list>
        <value>pam</value>
        <value>usermgmt</value>
      </list>
    </constructor-arg>
  </bean>

</beans>
```

6.Deploy the midas-wapam-sa.zip service assembly (command must be typed in one single line!):

```
# cp /opt/OV/OMU/adminUI/assemblies/midas-wapam-sa.zip \
     /opt/OV/OMU/adminUI/deploy
```

7.Restart the WebApp:

```
# /opt/OV/OMU/adminUI/adminui restart
```

Configuring the actual authentication software depends on this software itself and the OS the Administration UI WebApp is running on. In the following example, the default authentication mechanism on a Linux system is used.

Example: UNIX passwd

By default, the default authentication mechanism on Linux is UNIX passwd, which in turn is always available (normally) and there is no need to install and configure anything.

Create a user account and set the password, if not done yet, for example the user tge:

```
# useradd tge
# passwd tge
Changing password for tge.
New Password: *******
Reenter New Password: *******
Password changed.
```

Other UNIX-related parameters like home directory or shell are not needed for Administration UI.

Configure PAM to route Administration UI authentication requests to UNIX passwd by configuring the following in /etc/pam.conf:

```
[...]
midas    auth    required       libpam_unix.so.1
midas    account required       libpam_unix.so.1
[...]
```

For further details like advanced PAM capabilities (for example using multiple and/or optional authentication services) refer to the OS-specific PAM documentation.

LDAP integration

Administration UI supports user authentication through LDAP (Lightweight Directory Access Protocol). Administration UI includes the open-source component “Acegi Security System for Spring Project” (see `http://acegisecurity.org` for details).

New is LDAPS and the support of user search, which is needed for an Active Directory Server.

Currently, only basic authentication of user accounts is supported. No additional LDAP features like group membership etc. are used.

To configure LDAP authentication in Administration UI, perform the following steps:

1. Configure all Administration UI user accounts on the LDAP/Active Directory Server.

2. Configure the desired LDAP server in the Administration UI properties file `/opt/OV/OMU/adminUI/conf/ldap.properties` as shown in the following example:

```
# The LDAP URL
# Format: ldap://<host>:<port>/<base dn>
ldap.url=ldap://ldap-test:389/dc=bes-intern,dc=com
#ldap.url=ldaps://ldap-test:636/dc=bes-intern,dc=com
#ldap.url=ldaps://winsrv2008ad:636/dc=elephant-test,dc=org
```

For unencrypted access use `ldap.url=ldap://...` and for encrypted access use here as well `ldap.url=ldaps://...`
This applies both to a standard LDAP and an Active Directory Server.

In the next block enter the login credentials:

```
# Manager DN for login
ldap.managerDn=cn=Manager,dc=bes-intern,dc=com
#ldap.managerDn=cn=administrator,cn=users,dc=elephant-test,dc=org
# Manager password
ldap.managerPassword=******
```

For standard LDAP the default setting is `ldap.authenticationMode=BIND_WITH_DN`. Otherwise, no configuration changes are necessary here. Leave everything else commented out:

```
# The mode which is used for the authentication
# Allowed values are:
# BIND_WITH_DN:Use the authenticationDnPatterns for identifying a user
# USER_SEARCH : Use the authenticationSearchBase and
# authenticationSearchFilter for identifying a user
ldap.authenticationMode=BIND_WITH_DN
# The search base for searching users for authentication
# This property is used in combination with the ldap.authenticationSearchFilter
# and is used e.g. for a Active Directory search
#ldap.authenticationSearchBase=CN=Users
# The filter for searching users for authentication
# This property is used in combination with the ldap.authenticationSearchBase
# and is used e.g. for a Active Directory search
#ldap.authenticationSearchFilter=(sAMAccountName={0})
```

To use an Active Directory Server or user identification via user search, see the next page.

If you want to use an Active Directory Server or identify a user via user search, the following configuration has to be used. First of all the `ldap.authenticationMode` setting must be set to `USER_SEARCH`. Depending on the Active Directory Server configuration, the login name field also needs to be defined. In our example the field attribute is called `sAMAccountName`. Please note, that the `USER_SEARCH` function can also be used in LDAP, but generally the easier setup is done by using `BIND_WITH_DN` (see the previous page).

```
# The mode which is used for the authentication
# Allowed values are:
# BIND_WITH_DN : Use the authenticationDnPatterns for identifying a user
# USER_SEARCH : Use the authenticationSearchBase and
# authenticationSearchFilter for identifying a user
ldap.authenticationMode=USER_SEARCH
# The search base for searching users for authentication
# This property is used in combination with the ldap.authenticationSearchFilter
# and is used e.g. for a Active Directory search
ldap.authenticationSearchBase=CN=Users
# The filter for searching users for authentication
# This property is used in combination with the ldap.authenticationSearchBase
# and is used e.g. for a Active Directory search
ldap.authenticationSearchFilter=(sAMAccountName={0})
```

If the certificate originates from a proper third-party certification authority (like Verisign), no other change should be necessary (untested).

If a secure encrypted URL string is used, but without a certificate from a proper third-party certification authority, it is necessary to import the certificate also into the local Administration UI truststore. In order to do so the following two lines need to be enabled:

```
# The path to the truststore for trusted certificates for secure LDAP
ldap.truststore=conf/servicemix/truststore.jks

# The truststore password for secure LDAP
ldap.trustPassword=password
```

The import of the certificate (needs to be in the .cer format) is done via the following command (command must be typed in one single line!):

```
# /opt/OV/OMU/adminUI/jre/bin/keytool -import -alias ldapserver_a \
-keystore /opt/OV/OMU/adminUI/conf/servicemix/truststore_endpoint.jks \
 -file /tmp/ldap_server.cer
Enter keystore password: *******
[...]
Trust this certificate? [no]: yes
Certificate was added to keystore
```

The default password for the Administration UI truststore is: password

3. Activate the external authentication service in the `conf/auth.properties` file like this:

```
# vi /opt/OV/OMU/adminUI/conf/auth.properties

# configuration properties for authentication and authorization components
#auth-filter.enabled=false
caches.timeout=7200000
usermodel-router.authResource=file:conf/auth.xml
# eof
```

4. Switch Administration UI to LDAP authentication by configuring the `conf/auth.xml` as follows:

```
# vi /opt/OV/OMU/adminUI/conf/auth.xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN" "http://
www.springframework.org/dtd/spring-beans.dtd">
<beans>

 <bean id="targetServices" class="java.util.ArrayList">
 <constructor-arg>
      <list>
        <value>ldap</value>
        <value>usermgmt</value>
      </list>
    </constructor-arg>
  </bean>

</beans>
```

Whether LDAP, LDAPS or an Active Directory Server is used, leave the value set to “ldap” here.

5. Deploy the midas-waldap-sa.zip service assembly (enter the command in one single line!):

```
# cp /opt/OV/OMU/adminUI/assemblies/midas-waldap-sa.zip \
     /opt/OV/OMU/adminUI/deploy
```

6. Restart the WebApp:

```
# /opt/OV/OMU/adminUI/adminui clean
# /opt/OV/OMU/adminUI/adminui start
```

DST – Daylight Saving Time Patches

The Administration UI installer includes for convenience reasons a bundled JDK version 1.5, newer versions include 1.6.

‼ Please note, that for future JDK DST changes or JDK hotfixes you need to update Administration UI yourself. These JDK updates or hotfixes will not be included in any Administration UI patch.

For the JDK update Sun's tzupdater can be used for this purpose.

For the latest version please visit: http://java.sun.com/javase/downloads/

To check your existing Java version you can use the command below:

```
#/opt/OV/OMU/adminUI/jre/bin/java -version
```

In order to update your JDK/JRE image bundled in Administration UI after an installation, please perform the following steps:

1. stop Administration UI via:

```
#/opt/OV/OMU/adminUI/adminui stop
```

2. invoke the update tool via:

```
#/opt/OV/OMU/adminUI/jre/bin/java -jar tzupdater.jar -u -v
```

3. verify with:

```
#/opt/OV/OMU/adminUI/jre/bin/java -jar tzupdater.jar -t -v
```

4. start Administration UI via:

```
#/opt/OV/OMU/adminUI/adminui start
```

(intentionally left blank)