Oracle Agile Engineering Data Management

Security Guide

Release e6.2.0.0

E52560-02

September 2015

ORACLE

Oracle Agile Engineering Data Management/Security Guide for Agile, Release e6.2.0.0

E52560-02

Primary Author:

Contributing Author:

Contributor:

Contents

3 Wallets

4 Encryption

5 Authentication

6 Agile e6 Database User and Privileges

7 Securing Ports

8 Securing the Database

9 Additional Security Relevant Information

Preface

Agile PLM is a comprehensive enterprise PLM solution for managing your product value chain.

Audience

This document is intended for administrators and users of the Agile PLM products.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Related Documents

Oracle's Agile PLM documentation set includes Adobe® Acrobat PDF files. The Oracle Technology Network (OTN) website http://www.oracle.com/technetwork/documentation/agile-085940.html contains the latest versions of the Agile PLM PDF files. You can view or download these manuals from the Web site, or you can ask your Agile administrator if there is an Agile PLM Documentation folder available on your network from which you can access the Agile PLM documentation (PDF) files.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.

Convention	Meaning
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Overview of Agile e6 System

Agile Engineering Data Management (Agile EDM) is a Product Lifecycle Management solution that enables the engineering industry to manage its complete lifecycle of product development activities in a secure and collaborative application environment.

This document provides an overview of the Agile e6 system and discusses the security objectives and security architecture of Agile e6 modules. It also explains how to install and use the Agile e6, release e6.2.0.0, system securely. It includes specific information on how to enable security features, such as SSL, as well as more open ended discussions of the security implications of configuration choices.

Note: For detailed information about the Agile e6 system architecture please refer to the Architecture Guide for Agile e6.2.0.0.

Agile e6 System Services

Some responsibilities of the application server process have been assigned to dedicated services, being able to service several client processes in parallel. These are:

- File Management Services

  The File Management Services manages the files and attachments transaction and storage services, thus facilitating the check-in and check-out functionality provided by the Document Management System in the Agile e6 system.

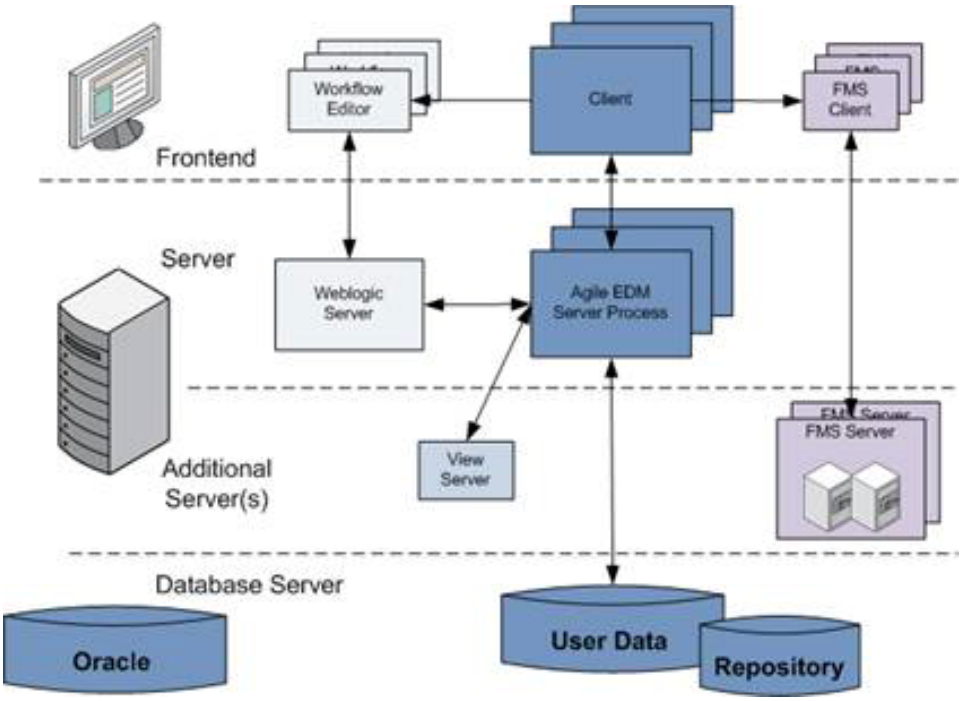
- Business Services

  The Business Services provides Agile e6 functionalities for Workflow Management, Product Configurator, and Permission Manager.

- Technical Services

  Technical Services encompass Java Client WebStart deployment, Java Client HTTPS support, Web Presentation Service, Web-Fileservice, Web Services, and Administration Client.

The Business Services, as well as the Technical Services, run on top of the Oracle WebLogic Application Server.



Agile e6 System Components

Server Side Components

The EDM Server components can reside on the same server where the EDM Server processes are executed (this is recommended for Business Services) or can reside on any other computer in the network (especially for the File Management Service - FMS).

The entire communication is based on TCP/IP. Only the unprivileged ports (above 1024) are used. Once a TCP/IP connection is established the port will not be changed dynamically.

- File Management Services
- ViewServer - (external) component of AutoVue Viewer that is used to view and redline documents (Office documents, 2D/3D-CAD Models)
- LDAP Server - (external) component (e.g. Oracle Identity Management Suite) to provide centralized store for managing user/password
- Kerberos Server - (external) component (e.g. Microsoft Active Directory) to provide centralized store for managing user/password and Single Sign-On (SSO) capabilities.
- Batch Client - component to run Agile e6 batch processes
- Java Daemon
- FMS Daemon
- PLM-API Proxy
- Web Services

Client Side Components

- Workflow Editor - to model and view workflows
- Office Suite - to check-in/check-out documents from/to Microsoft Office
  - Installed COM components

    The Office Suite uses the Addin-Express Framework for the Office Suite Addin module. The Addin-Express Frameworks installs several COM/.NET components. The following table lists the most important ones.

GUID	Class	Description
22C77E65-9597-4867-A5C2-EAF9078A4274	AddinExpress.MSO.ADXAddinInstaller	Addin-Express Addin Installer
8373210C-24AF-4561-9AE9-C829C4922415	AddinExpress.MSO.ADXAddinModule	Addin-Express Addin Loader
A767C1EC-710D-4A35-9A20-4AF4EDBB8BC0	GDMAddin2010.AddinModule	Office Suite Addin for Office 2010
A39F6C0D-4C64-4677-8751-0F82319B0635	GDMAddin2013.AddinModule	Office Suite Addin for Office 2013
FDB523C7-C0D9-47FB-A99C-EBAFF34E1F5F	GDMTools.clsConvertBookmark	Bookmark Conversion Toolkit
5E5BF33F-C063-47B1-AD65-CFBCA4CA6B65	GDMTools.clsConvertPDF	PDF Conversion Toolkit
6197B81A-16EC-4B42-905D-0B01DB8FA2A7	GDMTools.clsCreateZipSfx	ZIP Toolkit
05490640-BED4-42C4-9457-0A3B2C7F545E	GDMTools.clsCustomProperties	Custom Document Properties Toolkit
D628C366-F8FE-451E-9FCE-F9129C3002CF	GDMTools.clsDocumentProperties	Build-in Document Properties Toolkit
DF297172-733A-4223-BCDB-40A527C18AAB	GDMTools.clsFiles	File System Toolkit
9DCE1876-0F39-4DAF-B25C-3BF7EC23F19F	GDMTools.clsOffice	Basic Office Application Toolkit
E698A962-26B8-4BC3-A758-1F4C8497E5A6	GDMTools.clsOpenSave	File Dialog Toolkit
D10542A9-CDC3-4B8D-AAB3-3492EB4EAD24	GDMTools.clsProgressBar	Progress Report Toolkit
51971F0C-B912-457E-A86C-02DC135CA539	GDMTools.clsPrtDlg	Printer Dialog Toolkit
618B44CD-124B-4CEB-BB87-12C2844BE1E7	GDMTools.clsRefreshProperties	Refresh Document Properties Toolkit
9A07D5D6-71D6-4D95-9BD9-253E97487774	GDMTools.clsRegistry	Windows Registry Toolkit
46CADF67-6CAE-4D4F-8D2E-95005A06AB72	GDMTools.clsRegistryOffice	Advanced Windows Registry Toolkit
D4058C47-99D8-476E-BBB8-41CAA2425DA7	GDMTools.clsShell32	Basic Icon Toolkit
C189BD7F-B462-417E-AFA9-BC3A34FC22E0	GDMTools.clsTasks	Basic Control Toolkit

GUID	Class	Description
80172A67-C811-42C5-9D1A-E81FDFEE6A6C	GDMTools.clsVersion	Office Suite Version
ED25543D-F624-4314-A637-1E3DCDBDC375	GDMTools.clsVsImg32L	Basic Image Toolkit
13D600B4-28B1-4063-9E3D-E045D1494BB5	GDMTools.clsWinSys	Basic Window Control Toolkit

– Office Addin Registry Settings - example

Key	ADXStartMode	LoadBehavior
HKEY_CURRENT_USER\Software\Microsoft\Office\Excel\AddIns\GDMAddin2013.AddinModule	NORMAL	3 (start on startup)
HKEY_CURRENT_USER\Software\Microsoft\Office\PowerPoint\AddIns\GDMAddin2013.AddinModule	NORMAL	3 (start on startup)
HKEY_CURRENT_USER\Software\Microsoft\Office\Word\Addins\GDMAddin2013.AddinModule	NORMAL	3 (start on startup)
HKEY_CURRENT_USER\Software\Microsoft\Visio\AddIns\GDMAddin2013.AddinModule	NORMAL	3 (start on startup)

Security Objectives of Agile e6 System

- Providing Basic Security Services
- Supporting Standards
- Deployment and Configuration Flexibility
- Scalability and Predictability

System-wide Advice

Some advice applies to the entire system and the infrastructure in which it operates.

- Keep the software up to date
- Restrict network access to critical services
- Follow the principle of least privilege
- Monitor system activity
- Keep up to date on latest security information

2

Secure Environment - HTTP(S) Support

Note: Local FMS is not working in Secure Environment. Only Web Fileservice can be used in Secure Environment.

Prerequisites

Before starting to setup a secure environment, make sure your standard installation works without issues.

Note: For information about the installation, please refer to the Server Installation Guide on Windows and UNIX for Agile e6.2.0.0, or Client Installation Guide on Windows for Agile e6.2.0.0.

In the secure environment we will make some modifications to a standard environment. To configure the secure environment, you need a certificate from a trusted certificate authority.

In this example scenario a self signed certificate is used. The example scenario uses the Windows operating system. Self signed certificates require adding security exceptions to browsers and Java virtual machines to accept these certificates. Certificates from a trusted certificate authority do not need these security exceptions.

WARNING: We DO NOT recommend the use of Self Signed Certificates in production environments.

Setup of a secure productive environment needs expertise in network security setup. The following example shows a simple setup showing the Agile e6 requirements. In productive use customers need to extended the setup with their networking security infrastructure requirements.

Secure External Communication

In Agile e6 system, it is possible to setup external access to the Agile e6 environment with the Java Client or a web browser using the Web Client. It is possible to setup the complete external communication over Internet using the HTTP/S protocol. Internal intranet communication will still be via HTTP and RPC calls.

Note: For Agile e6, the Java Client is the main client.

This section describes how to setup a secure environment for this use case.

Secure external communication means the communication in the internet from a Java and Web Client over the internet to a server in the DMZ which acts as the End Point for the Java and Web Client. In case of Agile e6, the End Point is Apache HTTPD, configured as a reverse proxy in the DMZ.

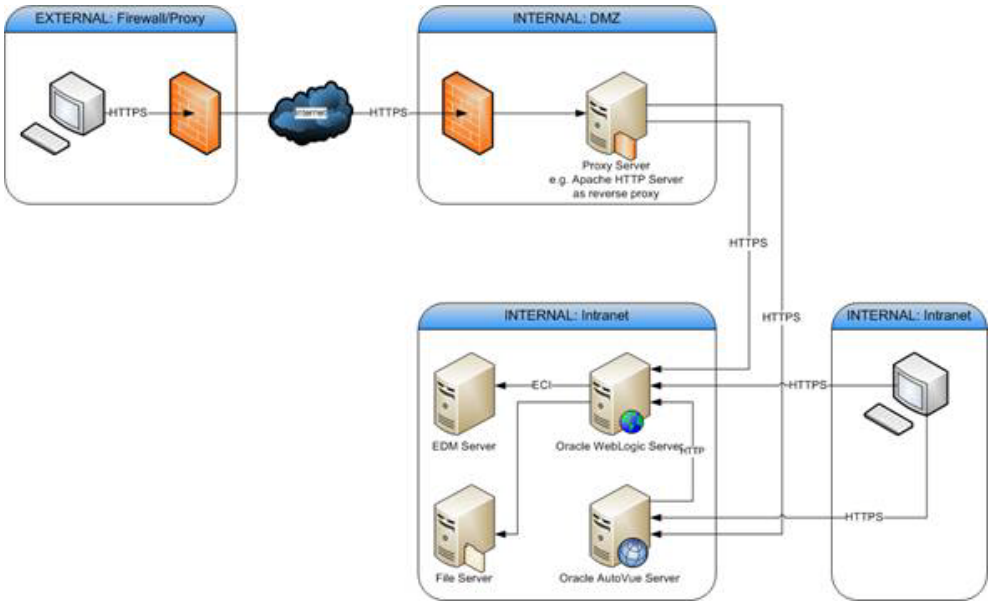
A certificate needs to be installed for Apache in the DMZ in this scenario.

Secure internal communication is only possible by enabling PLM-API for the Java Client like for the external secure communication; otherwise the communication from the Java Client to the EDM Server process is not secure.

Using the Web Client with HTTP/S and Web Fileservice with HTTP/S protocol is also secure.

The communication between the servers in the "secured" LAN is unencrypted. Customers need to take care that network traffic in this area cannot be read by unauthorized persons.

Following graphic illustrates an example communication.



As shown in the graphic above, the external Java and Web Client connect to the Proxy Server over HTTP/S. The internal Java and Web Client connect directly to the Oracle WebLogic server.

The proxy server is configured to pass the incoming requests from the internet to the end points on the Oracle WebLogic Server in the intranet.

The end points in the intranet are:

- PLM-API for Java Client HTTP support
- Java Client WebStart for Java Client download
- Web Presentation Service for the Web Client
- Web Fileservice for file transfer over HTTP
- AutoVue proxy for AutoVue HTTP support
- AutoVue viewer deployment for Applet download

- Core Web Service for business communication
- Streaming File Services for file transfer over Web Services

For this scenario, you have to adapt the Java Client to use HTTP/S protocol and configure the proxy server to pass the incoming requests and accept HTTP/S connections.

> **Note:** Setting up the firewalls or other DMZ infrastructure is not part of this document.

Setup Apache HTTPD as SSL Reverse Proxy

This document describes only the minimal Apache configuration needed, which is not sufficient for securing production environments. Refer to the Apache HTTP Server Version 2.4 Documentation at http://httpd.apache.org/docs/2.4/

This scenario is based on Apache HTTP Server (HTTPD) 2.4.x.

The next steps require an installed Apache, which does not need to be fully configured.

> **Note:** On Windows it is not recommended to install the Apache HTTPD Server below "*C:\Program Files"* or "*C:\Program Files (x86)"* because these directories require special permissions to write files by the HTTPD server. We use C:\App\Apache24 for this description.

Tokens in the Configuration Files

Replace the following tokens with the values in the following configuration examples.

Token	Description
<proxy_server>	The hostname of the Apache HTTPD server
<wl_server>	The hostname of the Oracle WebLogic server
<view_proxy>	The hostname of the Oracle AutoVue server

Changes in <apache_home>/conf/httpd.conf

1. For the Modules, enable or add the following lines:

```
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_connect_module modules/mod_proxy_connect.so
LoadModule proxy_http_module modules/mod_proxy_http.so
LoadModule substitute_module modules/mod_substitue.so
LoadModule ssl_module modules/mod_ssl.so
```

2. For the Reverse Proxy, add a new section:

```
<IfModule proxy_module>
   ProxyRequests Off
   SSLProxyEngine On
   # JNLP
   <Proxy "*.jnlp">
      SetOutputFilter SUBSTITUTE
      Substitute s#https://<wl_server>:7104/Jacc#https://<proxy_server>/Jacc#i
      Substitute s#https://<wl_server>:7104/JVue#https://<proxy_server>/JVue#i
   </Proxy>
```

```
   # Proxies
   ProxyPass /autovueproxy https://<view_
proxy>:8443/VueServlet/servlet/VueServlet
   ProxyPassReverse /autovueproxy https://<view_
proxy>:8443/VueServlet/servlet/VueServlet
   ProxyPass /JVue https://<view_proxy>:7104/JVue
   ProxyPassReverse /JVue https://<view_proxy>:7104/JVue
   ProxyPass /plmapi https://<wl_server>:7104/plm-api-axis/services
   ProxyPassReverse /plmapi https://<wl_server>:7104/plm-api-axis/services
   ProxyPass /Jacc https://<wl_server>:7104/Jacc
   ProxyPassReverse /Jacc https://<wl_server>:7104/Jacc
   ProxyPass /FileService https://<wl_server>:7104/FileService
   ProxyPassReverse /FileService https://<wl_server>:7104/FileService
   ProxyPass /AgilePlmWps https://<wl_server>:7104/AgilePlmWps
   ProxyPassReverse /AgilePlmWps https://<wl_server>:7104/AgilePlmWps
   ProxyPass /StreamingFileService https://<wl_
server>:7104/StreamingFileService
   ProxyPassReverse /StreamingFileService https://<wl_
server>:7104/StreamingFileService
   ProxyPass /CoreServices https://<wl_server>:7108/CoreServices
   ProxyPassReverse /CoreServices https://<wl_server>:7108/CoreServices
</IfModule>
```

3. For SSL, activate either one of the following lines:

   Include conf/extra/http-sni.conf

   Include conf/extra/http-ssl.conf

Changes in <apache_home>/conf/extra/http-sni.conf or <apache_ home>/conf/extra/http-ssl.conf

1. Modify one of the SSL configuration files which you activated in the previous step.
2. Add the hostname of your proxy server in the SSL configuration file to avoid getting an error while using the Java Client Webstart.
3. The following entries have to be added in the section <VirtualHost *:443>.
   - ServerName <proxy server:ssl port>

     The ServerName directive sets the request scheme, hostname, and port that the server uses to identify itself.
   - SSLCertificateFile <full qualified filename>

     This directive points to the PEM-encoded Certificate file for the server and optionally also to the corresponding RSA or DSA Private Key file for it (contained in the same file).
   - SSLCertificateKeyFile <full qualified filename>

     This directive points to the PEM-encoded Private Key file for the server. If the Private Key is not combined with the Certificate in the `SSLCertificateFile`, use this additional directive to point to the file with the stand-alone Private Key.
   - SSLCertificateChainFile <full qualified filename>

     This directive sets the optional all-in-one file where you can assemble the certificates of Certification Authorities (CA) which form the certificate chain of the server certificate. This starts with the issuing CA certificate of the server certificate and can range up to the root CA certificate. Such a file is simply the

concatenation of the various PEM-encoded CA Certificate files, usually in certificate chain order.

Note: For more information about SSL please refer to the Apache mod_ssl documentation.

Check Your Configuration

- The reverse proxy is accessible and the default Apache website appears:

  https://<proxy_server>

- Following links must be accessible without error:

  https://<proxy_server>/FileService

  https://<proxy_server>/plmapi

  https://<proxy_server>/Jacc

  https://<proxy_server>/AgilePlmWps

- Following links must be accessible without error if you use AutoVue:

  https://<proxy_server>/autovueproxy

  https://<proxy_server>/

  https://<proxy_server>/JVue/jvue.jar

  https://<proxy_server>/autovuewrapper/agile-jvue-wrapper.jar

- Following links must be accessible without error if you use Web Services:

  https://<proxy_server>/CoreServices/BusinessObjectService?WSDL

  https://<proxy_server>/StreamingFileService/DocumentFileService?WSDL

Setup HTTPS on the Oracle WebLogic Servers

The demonstration digital certificates, private keys, and trusted CA certificates should be used in a development environment only. They should NOT be used in a production environment.

They are provided by default during WebLogic installation/domain setup.

For a production environment follow the steps in the WebLogic documentation to use non-demonstration digital certificates, private keys, and trusted CA certificates instead.

Note: A demonstration certificate of the WebLogic server will not be accepted by the Apache HTTPD reverse proxy. This will usually result in the proxy error "Error during SSL Handshake with remote server" while accessing the WebLogic server over the proxy server.

Setup the Java Client/Java Virtual Machine

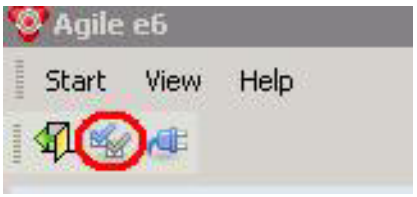
To enable HTTP/S support in the Java Client, you have to activate/change the HTTP/S support setting.

1. Download Java Client WebStart.
2. Open https://<proxy_server>/Jacc and select the download link.

3. After the download of the application, a warning appears that the digital signature of the application cannot be verified. If you trust the publisher, select "Always trust content of this publisher", and run the application.

   The Java Client starts.

4. Open the Java Client Preferences mask.



5. Enable the HTTP/S support and change/add the Server URL and Service name.

- For external clients to:

  Server URL: https://<fully qualified proxy_server name>

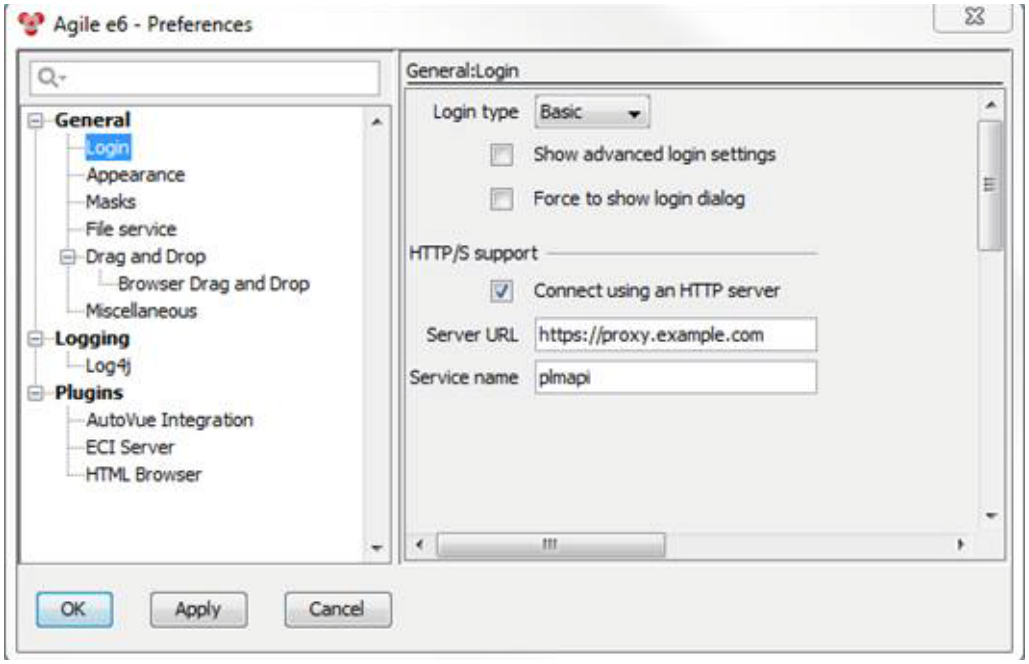
  Service name: plmapi

- For internal clients to:

  Server URL: https://<fully qualified wl_server name>:<SSL port>

  Service name: plm-api-axis/services

  The following screen shows an example configuration for an external client:



Java Client with Proxy

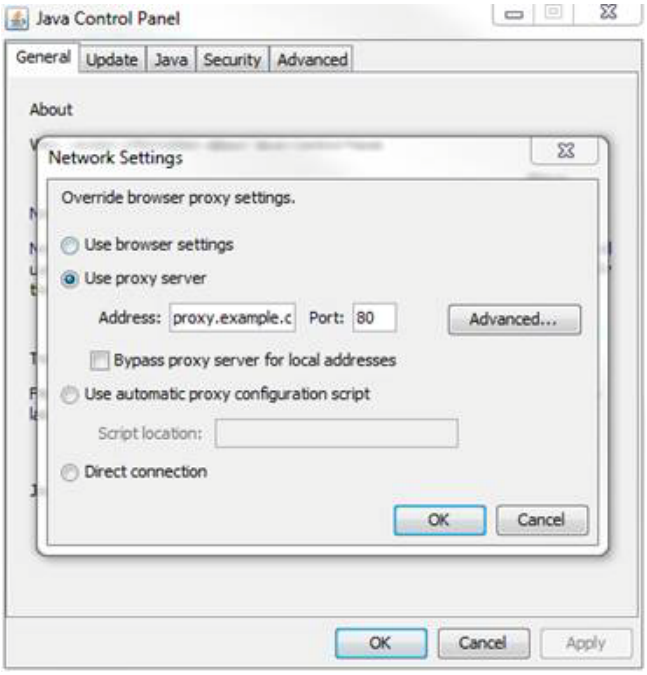
By default, the Java Client uses the proxy configuration of your Java environment. This is configured on the client side in the Java Control Panel.

To configure the proxy settings:

1. Open MS Windows Control Panel.
2. Execute Java.

   The Java Control Panel opens.
3. In the General tab, open the Network Settings….

> **Note:** Depending on your proxy configuration, one of the proxy settings has to be selected.



Setup Web Fileservice

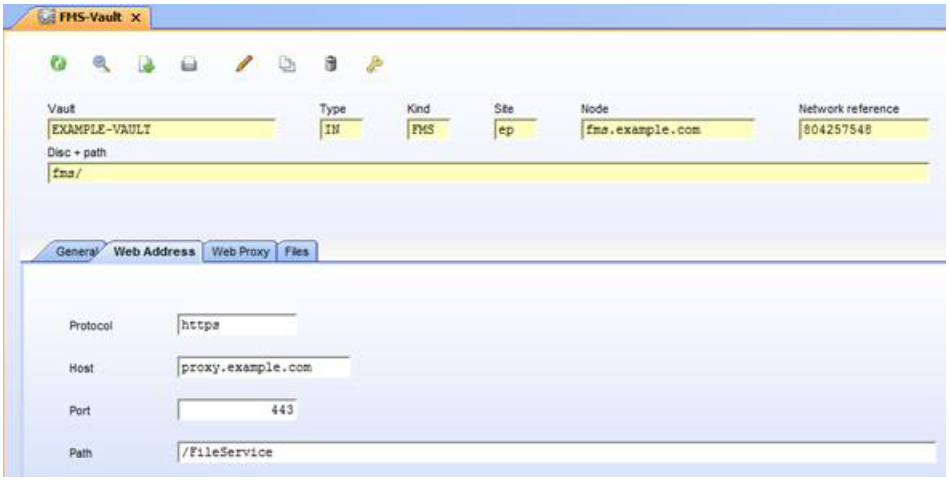
To enable HTTP/S support for the Web Fileservice, the web address in the vault configuration has to be changed.

> **Note:** As you can see in the following screenshot, you can only define one web address per vault. This means the web address is the same for internal and external Java/Web Client. If you want to use this vault for internal and external Java/Web Clients, the <proxy_server> must be reached under his name from external and also from internal.

1. Start the Agile e6 Java Client with a manager user
2. Select Manager > File Management > Vaults.

```
Protocol: https
Host: <proxy server>
Port: 443
Path: /FileService
```

The following screenshot shows the configuration:



Setup AutoVue

General setup of AutoVue is described in the "AutoVue Integration Installation and Administration Guide for Agile e6.2.0.0".

Note: SSL has to be activated for the servlet container of the AutoVue deployment.

Note: Perform these steps only if you are using AutoVue.

1. Setup the AutoVue tunneling servlet that it can be accessed via HTTP/S.

   For example, tomcat is configured with HTTP/S.
2. Start an Agile e6 Java Client with a manager user.
3. Select System > AutoVue > Configuration.
4. To enable secure communication, the following values must be changed:
   - EDB-PVM-AV-PROXY:

     Value (for example): https://<proxy_server>/autovueproxy

     Description: The URL where the AutoVue tunneling servlet can be accessed via HTTP/S.
   - EDB-PVM-AV-USE-PROXY:

     Value: true

     Description: Default is set to TRUE to use HTTP/S communication.
   - EDB-PVM-AV-DMS:

     Value (for example): https://<weblogic_server>:<weblogic port>/VueLink/Vuelink

     Description: The Oracle Agile DMS servlet address, where the AutoVue server can access the DMS servlet.

Setup Web Client

No configuration changes have to be done if you want to use the Web Client with HTTP/S.

Note: Only use the HTTP/S protocol and port in your browser.

In the dump, the Web Fileservice adaptations also have to be done (see above) to use file checkin/out in the Web Client with HTTP/S. Proxy configuration will be used from the browser.

Setting up WPS to use HTTPS

Obtain the appropriate certificate for your organization. This must be placed within your installation, and referenced from the file server_web.xml.

In the following you can find an example of the SSL section in the server_web.xml file.

Change the value of the "keystore" parameter to match the location of the certificate.

```
<Connector className="org.apache.tomcat.service.PoolTcpConnector">
     <Parameter name="handler"
         value="org.apache.tomcat.service.http.HttpConnectionHandler"/>
     <Parameter name="port"
         value="8443"/>
     <Parameter name="socketFactory"
         value="org.apache.tomcat.net.SSLSocketFactory" />
     <Parameter name="keystore"
         value="[example: c:/<Agile_root>/tomcat/conf/keystore]" />
     <Parameter name="keypass" value="password"/>
 </Connector>
```

Deactivate Web Client

If users do not use the Web Client, we recommend to deactivate the Web Client.

To deactivate, follow these steps:

1. Open the WebLogic installation console.
2. Login with your WebLogic administrator user.
3. Open the Deployments.
4. Select the WebPresentationService.
5. Click Stop.

   This opens a selection menu. We recommend to use one of the following two options:

   - When work completes
   - Force Stop Now

Change Lightweight Report URL

The attribute Report_Service_URL has to be changed in all application configuration files in <ep_root>/init/<application>.xml.

1. Search the following text:

   ```
   <PLMPresentationServices Report_Service_URL="…"/>
   ```

2. Adapt the Report_Service_URL to:

   `https://<proxy_server>/AgilePlmWps/reporter/report`

   If you change this line, the change is valid for the complete application. An internal client will also use this URL.

Setup Workflow Mailing

The workflow mailing and the server side external mailing provide SSL and authentication for mailing.

In the file ABS_<application>.ini, these configuration parameters can be found:

Configuration Parameter	Description
PORT=	Standard SMTP (mail server) port for SSL = 587, without SSL = 25
ConnectTimeOut=60	Value in seconds; default 1 minute=60 seconds
useSSL=false	SSL can be used without additional authentication. If this entry is deleted, the default is true.
useAuthentication=false	If this entry is deleted, the default is true. This means active SSL is used and SMTPUserName and SMTPPassword must be set!
SMTPUserName=	The user name, account name of the SMTP account. Note: Always use<account-name>@<domain-name>
SMTPPassword=	Password encrypted with epkeytool) Note: Unencrypted passwords are never accepted, even if authentication is switched off!

Note: These configuration parameters can be filled during installation, but can be changed afterwards. To make changes active, please redeploy the business service.

Note: Empty password is possible as long as authentication is off.

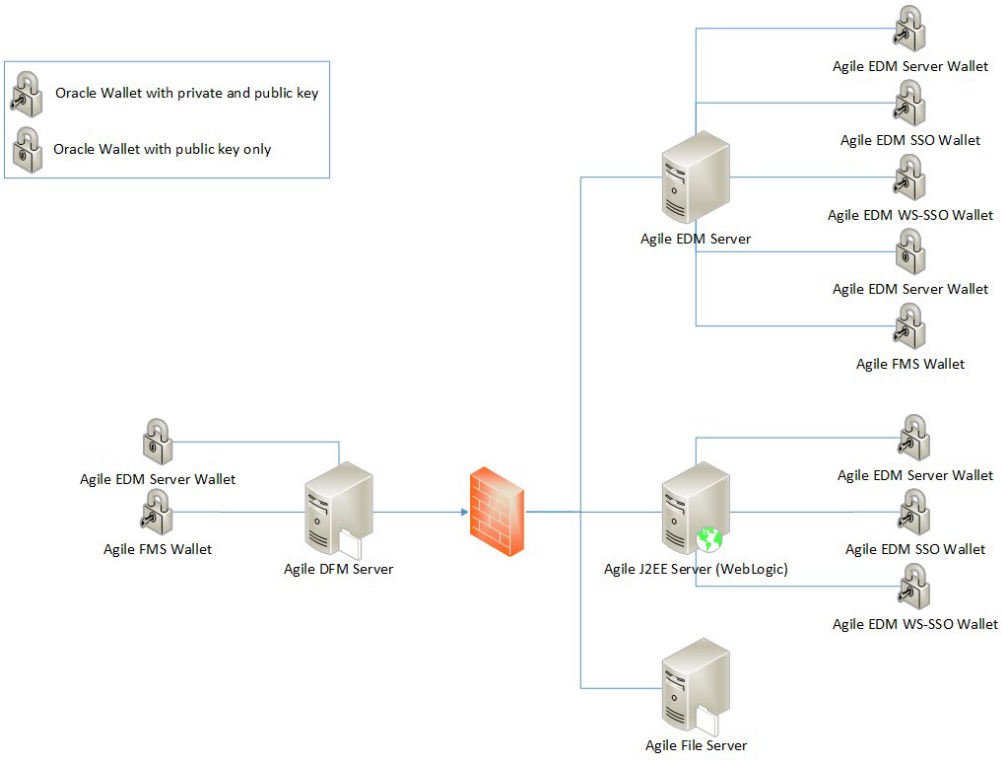
3

Wallets

To improve the security of passwords and tickets, Oracle wallets are used and can be customer specific. The wallets are created by the Agile e6 installer during the installation.

Overview Wallet Infrastructure

The following graphic depicts all wallets created during the installation and deployed to the different components.



- Agile EDM Server Wallet

  This Oracle wallet is the main wallet of the Agile e6 installation. This wallet is generated during the installation of the Agile EDM Server.

  There are two types of this Oracle wallet:

  – Private Wallet

    The private wallet includes the private and public key.

This wallet is used by the Agile EDM server installation and by the Agile J2EE server installation.

* The Agile EDM server uses this wallet to encrypt the database password in the configuration file.

  In addition, the wallet is used to create FMS tokens and tickets.

  The Java Daemon also uses this wallet to secure the Java Daemon administration password.

* The Business services (deployed in WebLogic) use the wallet to encrypt the mail SMTP password in the configuration file.

– Public Wallet

  The public wallet which only contains the public key.

  This wallet is used for the DFM installation on remote locations to verify tokens and tickets generated by the Agile EDM Server.

- Agile EDM SSO Wallet

  This Oracle wallet is used to create a trusted relationship between the Agile EDM Server which delegates the SSO authentication to the Business service.

- Agile EDM WS-SSO Wallet

  This Oracle wallet is used to verify WS-SSO tickets created by the Core WebService during login to the Agile EDM Server.

- Agile FMS Wallet

  This Oracle wallet is locally generated for each DFM location. The FMS Java Daemon uses this wallet to secure the access to the internal FMS interface.

The Agile e6 installer creates 2 ZIP files.

- The first ZIP file contains the private wallets and has to be used during the installation of additional Agile EDM Servers or J2EE servers.
- The second ZIP file contains the public Agile EDM Server wallet. This package has to be used on remote DFM locations.

Manual Creation of Wallets

There are some use cases which make it necessary to create a Oracle wallet manually.

- Update of the Oracle wallet infrastructure for security reasons
- The needed oracle wallet is not created during the installation, for the following components you need to create the Oracle wallet manually:
  – Batch Client
  – OfficeSuite PDF generator service
  – AutoVue Offline Metafile cache service
  – EIP

To create a new wallet, use the epkeytool, which is available in the e6 server installation. This tool allows you to create Oracle wallets for the following components:

- **adminclient**, used by the Admin client which is used to administrate your applications

- **batch**, used by the Batch Client, OfficeSuite PDF generator service and the AutoVue Offline Metafile cache service
- **eip**, used by the EIP
- **fms**, used by the FMS Java daemon
- **server**, used by the e6 server
- **sso**, used by Java Client SSO
- **ws**, used for WebService SSO
- **upt**, used by the upgrade tool

The epkeytool needs to know the location of the wallet root path.

The standard root path is %ep_root%/init/wallet.

Note: The Oracle wallet must be protected so that only the services that are using the wallet can access it! No one else has access to the wallet.

Note: The epkeytool never overwrites an existing Oracle wallet. If you want to update your Oracle wallet infrastructure, then you must first move the old wallets out of the root wallet location.

The following sections show how you can create and update your Oracle wallet structure for the different components.

Admin Client

The initial Oracle wallet is created during the installation. Each admin client installation has a unique Oracle wallet.

The Oracle wallet root path for the admin client is located at

Windows: %ALLUSERSPROFILE%\agile\installer\6.2.0\wallets

UNIX: ${HOME}/.agile/installer/6.2.0/wallets

To update the wallets, move the existing wallets from that location and then call the epkeytool to create a new one.

```
epkeytool -w adminclient -c -r C:/ProgramData/agile/installer/6.2.0/wallets
```

Output:

```
Created private wallet with type ADMINCLIENT at
C:/ProgramData/agile/installer/6.2.0/wallets/adminclient
--- Content of wallet at
C:\ProgramData\agile\installer\6.2.0\wallets\private\adminclient:
Requested Certificates:
User Certificates:
Subject:        C=US,ST=California,L=Redwood City,O=Oracle,OU=Agile
PLM,CN=EDMADMINCLIENT
Trusted Certificates:
Subject:        C=US,ST=California,L=Redwood City,O=Oracle,OU=Agile
```

```
PLM,CN=EDMADMINCLIENT
```

> **Note:** After updating the Oracle wallet for the admin client, then you must re-encrypt the admin user password. See the Administration Guide for information about how to change the password.

Batch

If there is more than one batch client installed on your server, then:

- You can choose to create a separate Oracle wallet for each batch client.

  If you use a separate Oracle wallet for each batch client installation, then use the following path as the wallet root location:

  ```
  %batch_root%/init/wallet
  ```

- Or, the batch clients can share the same Oracle wallet.

  If the batch clients share the same Oracle wallet, then use a dedicated path which can be accessed by all batch client installations.

The batch client installation package contains a script to simplify the usage of the epkeytool.

Use the batchkeytool to Create the Oracle Wallet

1. Create a wallet root directory. (default location: %BATCHCLIENT_ROOT%\init\wallet)

   The default location is:

   ```
   %BATCHCLIENT_ROOT%\init\wallet
   ```

2. Protect the created wallet root directory against unauthorized file access!
3. Prepare the batchkeytool.cmd script (`%BATCHCLIENT_ROOT%\axalant\cmd`) and configure the JAVA_HOME setting.
4. Call the batchkeytool to create the Oracle wallet:

   ```
   batchkeytool -c
   ```

   The following output should appear:

   ```
   Created private wallet with type BATCH at
   D:\dev\batchclient\axalant\cmd\..\..\init\wallet\private\batch

   --- Content of wallet at
   D:\dev\batchclient\axalant\cmd\..\..\init\wallet\private\batch:
   Requested Certificates:
   User Certificates:
   Subject:        C=US,ST=California,L=Redwood City,O=Oracle,OU=Agile
   PLM,CN=EDMBATCH
   Trusted Certificates:
   Subject:        C=US,ST=California,L=Redwood City,O=Oracle,OU=Agile
   PLM,CN=EDMBATCH
   ```

The Oracle wallet is used by the batchkeytool to allow to encrypt user passwords, which must be used to store the password in the batch scenario properties file.

The batch client itself uses the Oracle wallet to decrypt the password to connect to the EDM server.

> **Note:** The password is encrypted during the login by using a secure generated session key which is protected by using the EDM server certificate.

Use the epkeytool to Create the Oracle Wallet for a Batch Client

Alternatively, you can also use the epkeytool to create the Oracle wallet for the batch client instead of using the batchkeytool.

> **Note:** The epkeytool is part of the e6 server.

1. Create a wallet root directory. (default location: %BATCHCLIENT_ROOT%\init\wallet)

   The default location is:

   ```
   %BATCHCLIENT_ROOT%\init\wallet
   ```

2. Protect the created wallet root directory against unauthorized file access!
3. Prepare the batchkeytool.cmd script (`%BATCHCLIENT_ROOT%\axalant\cmd`) and configure the JAVA_HOME setting.
4. Call the batchkeytool to create the Oracle wallet:

   ```
   epkeytool -w batch -c -r d:/dev/batchclient/init/wallet
   ```

   The following output should appear:

   ```
   Created private wallet with type BATCH at
   d:\dev\batchclient\init\wallet\private\batch

   --- Content of wallet at d:\dev\batchclient\init\wallet\private\batch:
   Requested Certificates:
   User Certificates:
   Subject:        C=US,ST=California,L=Redwood City,O=Oracle,OU=Agile
   PLM,CN=EDMBATCH
   Trusted Certificates:
   Subject:        C=US,ST=California,L=Redwood City,O=Oracle,OU=Agile
   PLM,CN=EDMBATCH
   Subject:        C=US,ST=California,L=Redwood City,O=Oracle,OU=Agile
   PLM,CN=EDMBATCH
   ```

5. After updating the Oracle wallet you must re-encrypt the password in the batch scenario properties file. See Chapter 4, "Encryption" for more information.

EIP

See the EIP documentation for information about how you can create or update the Oracle wallet and re-encrypt the password.

FMS

The create the Oracle wallet for the FMS Java Daemon call the epkeytool like this example:

```
epkeytool -w fms -c -r d:/plm/init/wallet
```

Output:

```
Created private wallet with type BATCH at d:\plm\init\wallet\private\batch
--- Content of wallet at d:\plm\init\wallet\private\batch:
Requested Certificates:
User Certificates:
Subject:        C=US,ST=California,L=Redwood City,O=Oracle,OU=Agile PLM,CN=EDMFMS
Trusted Certificates:
Subject:        C=US,ST=California,L=Redwood City,O=Oracle,OU=Agile PLM,CN=EDMFMS
```

The FMS Java Daemon does not encrypt any permanent passwords, the wallet is only used to secure dynamic sessions.

Server

To update the e6 server wallet is a bit more complex, because all e6 servers must use the same Oracle wallet and the public server wallet must also be deployed to all J2EE and DFM installations.

1. Create the Oracle wallets, like this example:

   ```
   epkeytool -w server -c -r d:/plm/init/wallet -p
   ```

   Output:

   ```
   Created private wallet with type SERVER at d:\plm\init\wallet\private\server
   --- Content of wallet at d:\plm\init\wallet\private\server:
   Requested Certificates:
   User Certificates:
   Subject:        C=US,ST=California,L=Redwood City,O=Oracle,OU=Agile PLM,CN=EDM
   Trusted Certificates:
   Subject:        C=US,ST=California,L=Redwood City,O=Oracle,OU=Agile PLM,CN=EDM
   Created public wallet with type SERVER at d:\plm\init\wallet\public\server
   --- Content of wallet at d:\plm\init\wallet\public\server:
   Requested Certificates:
   User Certificates:
   Trusted Certificates:
   Subject:        C=US,ST=California,L=Redwood City,O=Oracle,OU=Agile PLM,CN=EDM
   ```

   As you can see two wallets are created, one in the private folder and one in the public folder.

2. If your J2EE deployment is on the same server as the e6 server you can just re-deploy the BusinessService.

3. If you have a component based installation, then copy the private server wallet into the corresponding folder of your J2EE installation.

   Then re-deploy the BusinessService.

   Note: If you are using Workflow Mailing, you need to re-encrypt the password.

4. For DFM locations, just copy the Oracle wallet from the public folder to the corresponding folder in your DFM installation.

SSO

To update the Java Client SSO wallet, execute the following steps.

1. Create the Oracle wallet like this example:

```
epkeytool -w sso -c -r d:/plm/init/wallet
```

Output:

```
Created private wallet with type SSO at d:\plm\init\wallet\private\sso
--- Content of wallet at d:\plm\init\wallet\private\sso:
User Certificates:
Subject:        C=US,ST=California,L=Redwood City,O=Oracle,OU=Agile
PLM,CN=EDMSSO
Trusted Certificates:
Subject:        C=US,ST=California,L=Redwood City,O=Oracle,OU=Agile
PLM,CN=EDMSSO
```

2. If your J2EE deployment is on the same server as the e6 server, then you can just re-deploy the BusinessService.
3. If you have a component based installation, then copy the private server wallet into the corresponding folder of your J2EE installation.

   Then re-deploy the BusinessService.

The wallet is not used to encrypt permanent passwords so you have no follow up actions.

WebService SSO

To update theWebService SSO wallet, execute the following steps.

1. Create the Oracle wallet like this example:

```
epkeytool -w ws -c -r d:/plm/init/wallet
```

Output:

```
Created private wallet with type WEBSERVICE at d:\plm\init\wallet\private\ws
--- Content of wallet at d:\plm\init\wallet\private\ws:
Requested Certificates:
User Certificates:
Subject:        C=US,ST=California,L=Redwood City,O=Oracle,OU=Agile
PLM,CN=EDMWS
Trusted Certificates:
Subject:        C=US,ST=California,L=Redwood City,O=Oracle,OU=Agile
PLM,CN=EDMWS
```

2. If your J2EE deployment is on the same server as the e6 server. then you can just re-deploy the CoreWebServices

3. If you have a component based installation, then copy the private server wallet into the corresponding folder of your J2EE installation.

   Then re-deploy the CoreWebServices.

The wallet is not used to encrypt permanent passwords so you have no follow up actions.

Upgrade Tool

See the upgrade tool documentation for information about how you can create or update the Oracle wallet and re-encrypt the password.

4

Encryption

With Agile e6.2.0.0, the Advanced Encryption Standard (AES) is supported.

This encryption mechanism is used to encrypt the passwords within property files.

Encrypt Passwords

In some cases you may need to encrypt passwords manually.

To encrypt a password, you can use the epkeytool which is part of the Agile e6 installation.

The epkeytool can be started by calling:

```
%EP_ROOT%\axalant\cmd\epkeytool.bat -encryptpwd -keyStore file://<complete path to
the wallet which has to be used>/cwallet.sso -keyAlias orakey
```

> **Note:** Which wallet you have to use depends on which component should work with the encrypted password.
>
> The section "Manual Creation of Wallets" explains in detail how to create manually a wallet and the manual deployment of that wallet.

The epkeytool prompts for the password to encrypt, the output (encrypted password) will look similar to:

```
{PLM-AES-128}RSA-PUBLIC-BASE64:QjFurSOpjlhQER+wZFF7L/XgD1+npwlEBcK0DDpNeYJ8gbxhIxu
MZpZ4yEsuGuJQ5eZJiUHsHEW1X1pJddylUmrZm6rn+rx/BOfZlITnUvMpF93Ej11wdVu+DObmSazKD3v7r
pAwpKXsFMeiKCVVVF7g5C2k033/UZTCnoPUAtE={PLM-AES-128}CVVOULGVgv06h2FJCMrAGrvyEgCeV9
S0gZoTF4uCgL8=
```

Secured Components

For the following components you need to encrypt passwords manually:

- Batch Client
- OfficeSuite PDF Generator
- AutoVue Offline Metafile Cache

All these components are based on the Batch Client technology. For each scenario, the components have property files which contain the Batch user password.

Note: An Agile e6 batch user account must have limited access to the Agile e6 system and the installation directory needs to be secured to protect the properties files.

The Batch Clients do not support clear text passwords.

WebLogic Encryption

Passwords for WebLogic cannot be encrypted with the epkeytool. They have to be encrypted with the WebLogic server.

These passwords can only be encrypted with the WebLogic domain where they will be used. WebLogic passwords depend on a domain specific secret.

This means that the passwords in the batch installation properties file, which are WebLogic specific, cannot be stored encrypted when the WebLogic domains will be created with an Agile e6 batch installation.

It is possible for the (re)deployment of the Business Service to store the database password encrypted in the batch installation properties file. The following script can be used to generate an encrypted password:

`$ep_root/build/applicationServer/weblogic_121/scripts/<app_domain>/WLSencrpyt.`
All of the following passwords can only be used unencrypted for a batch installation:

- WebLogic Admin Password Installation Domain
- WebLogic Admin Password Application Domain
- PLM Authenticator Password

Agile e6 Encryption

The epkeytool is available directly from the installation package. The scripts for Windows and UNIX are located in the directory installer/tools/bin.

EDM Server

The following list shows all passwords that are encrypted with the epkeytool.

- Database Password in the ep_root/init/<env>.xml file
- Java Daemon Administration Password
- Unprivileged Windows User Password

  Local Windows User which is used by the following services:

  – Java

  – FMS

  – Java and Portmapper

File Server

Privileged Windows User Password that use Windows encryption mechanisms.

Local Windows User which is used by the following service:

- File Server

WebLogic

Mail Auth User Password for the Business Service in the WebLogic domain.

Batch Clients

- Batch user in properties files for standard Batch Client
- Batch user in properties files for Office Suite PDF generator
- Batch user in properties files for AutoVue Offline Metafile cache

Enterprise Integration Platform Encryption

The encryption tool is available directly from the installed package. The scripts for Windows and UNIX are located in the directory bin. Please refer to the Enterprise Integration Platform Administration Guide for more details.

5

Authentication

LDAP Support

LDAP (Lightweight Directory Access Protocol) is an application protocol for querying and modifying directory services running over TCP/IP.

With Agile e6.2.0.0, LDAP based authentication is supported. While an Agile e6 user is logging in to the Agile e6 system through any supported client, the password of that user will be verified with the one stored in the LDAP repository. There is no additional validation or storage of the password within EDM.

The communication between the EDM Server and the LDAP repository has to be set up. Each Agile e6 user, as configured in the Agile e6 database, has to be available in the LDAP repository in order to be authenticated at login.

> **Note:** Although LDAP support enables only ONE password for many different systems, it should not be confused with automatic Single-Sign-On (SSO) support. This would allow a user to log on automatically, without being asked to provide user login and password!

Prerequisites

- LDAP Server (Oracle Internet Directory / MS Active Directory / other LDAP server).
- Oracle LDAP Client (part of the Oracle client installation).
- LDAP configuration site-specific sites are created in DDM Site Vaults and can be used in all modules. To create a DDM Site Vaults refer to the Online Help > General Replication.

  > **Note:** For the module "General Replication", the license "Distributed Objects" is required and has to be active.

- An Agile e6 user name and an LDAP user name must have been created.

Note: The Agile e6 user name and the LDAP user name don't need to be identical. But it is required that the LDAP user name is configured in the Agile e6 system.

User Authentication via LDAP

An LDAP directory is often used to manage users and organization units in a central environment. Products like the Oracle Internet Directory are able to manage users, groups, and organization units in a standard LDAP environment and are compatible with the most other LDAP servers which are based on the LDAP standards.

Note: For more information on password policy for Oracle Internet Directory, refer to OID documentation on http://www.oracle.com/technetwork/documentation/oid-089101.html.

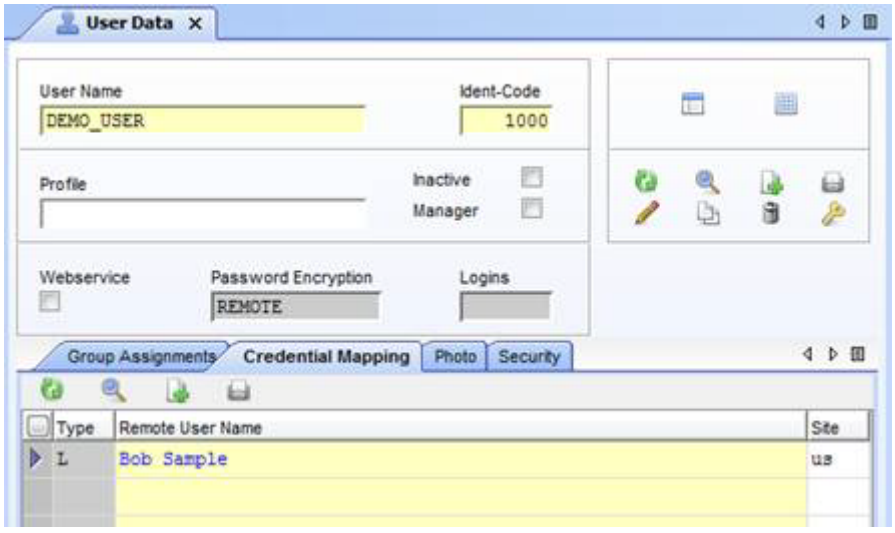
With Agile e6.2.0.0, the LDAP authentication mechanism supports the authentication of an Agile e6 user password against an LDAP repository.

The LDAP for Agile e6.2.0.0 uses the Base-DN for a direct access path to authenticate the user. LDAP integration does not support relative search paths.

Note: Security features like changing the password, locking/unlocking accounts etc. and any policies like password length etc. are managed in the external LDAP system and not in the Agile e6 system.

Setup an LDAP User

By mapping an Agile e6 user to a Remote Credential, you can change the used login mechanism of an Agile e6 user.



Typically, an Agile e6 user has a different user name in the LDAP repository, therefore a remote user name field is supported to map the user names. To support

multi-domains in LDAP, the administrator can link each user to the site specific LDAP configuration.

For more information about the Credential Mapping see the Online Help for DataView > Enhanced User Management.

The LDAP system takes care of the password policies (expiration and format).

Note: The enhanced security module and the possibility to change the password within Agile e6 are deactivated for LDAP users.

Configuration Parameter

The LDAP configuration used by the Agile e6 system is stored in the database (T_CFG_DAT) as configuration parameters.

Configuration Parameter	Default Value	Description
EDB-LDAP-HOST	<LDAP-host>	LDAP host name
EDB-LDAP-HOST-<number>	<LDAP-host>	Backup LDAP host name (optional). See chapter "Support Backup LDAP Server for Fail Over" below for further information.
EDB-LDAP-PORT	-	LDAP service port (=default port depends on encryption mode)
EDB-LDAP-BASEDN	cn=users, dc=example, dc=com	Base DN of the user group
EDB-LDAP-RDN	cn	Relative DN of the user (optional)
EDB-LDAP-ENCRYPTION	Yes	LDAP encryption mode (yes=SSL, no=unsecure)
EDB-LDAP-WALLET-LOCATION	-	Path to the Oracle Wallet which contains the LDAP server certificate (UNIX only)

The configuration entries are site specific to support multi-domains. For each site, different LDAP settings can be configured.

Note: Only default and site-specific LDAP configuration parameters are supported. Language-specific parameters are not supported.

Secure LDAP Connection

Note: LDAP supports SSL and it is recommended to enable SSL for the LDAP connection.

To secure the LDAP connection, the LDAP server certificate must be installed on the EDM Server. The mechanism to install the LDAP server certificate is different on Windows and UNIX.

Note: The following information applies to self signed as well as to official certificates.

- Windows

  The LDAP certificate has to be installed in the Windows Certificate Store.

- UNIX

  The LDAP certificate has to be imported into the Oracle wallet. The location of the Oracle wallet is configured in the database in the LDAP configuration parameter.

Support Oracle Wallet to store the LDAP Server Certificate

SSL Encryption on UNIX

1. Create an Oracle Wallet by using the Oracle Wallet Manager.

   This is part of the database client installation.
2. Import the LDAP server certificate as trusted certificate into the Oracle Wallet.
3. Activate the auto login feature of the Oracle Wallet.
4. Save the Oracle Wallet to a secure location, e.g. $ep_root (example /app/plm6).
5. Login in to Agile e6 with a manager user.
6. Configure the LDAP server configuration parameters to use the Windows active directory.

   Example:

   - EDB-LDAP-HOST: ldap.example.com
   - EDB-LDAP-BASEDN: CN=Users,DC=example,DC=com
   - EDB-LDAP-ENCRYPTION: yes
   - EDB-LDAP-WALLET-LOCATION: file:/app/plm6
7. Configure an Agile e6 user to use the LDAP to login.
8. Map Agile e6 user to an LDAP user.
9. Login to Agile e6 by using the Agile e6 user which is configured to use LDAP.

Import LDAP Server Certificate on Windows

SSL Encryption on Windows

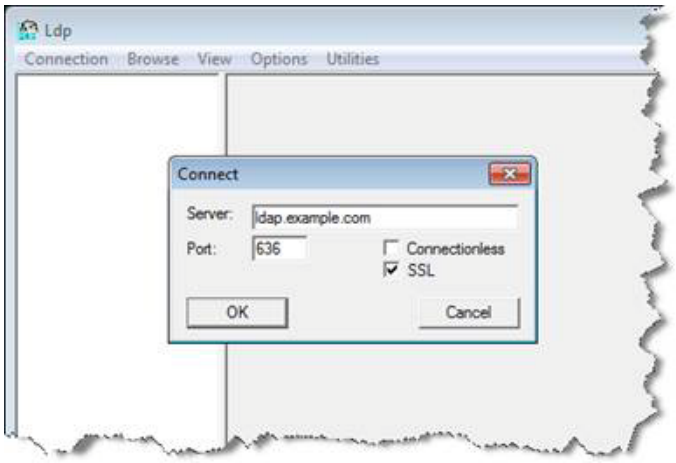
1. Logon on Windows EDM Server with the runtime account of the Agile e6 system (e.g. axalanrt user)
2. Normally you get two certificates from your LDAP system administrator which have to be imported into your EDM Server.

   - The Certificate Authority (CA) Certificate must be imported into "Trusted Root Certificate Authorities" group.
   - The LDAP Server Certificate has to be imported into the "Other People" group.

Verify LDAP Environment

Microsoft provides a program to test an LDAP connection. The program is called ldp.exe and can be downloaded from Microsoft Support.

1. Open ldp.exe.
2. Enter your server name.
3. Enter your port number.
4. If required, activate SSL.
5. Click OK.



The output should look like this for a successful test:

```
ld = ldap_sslinit("ldap.example.com", 636, 1);
Error <0x0> = ldap_set_option(hLdap, LDAP_OPT_PROTOCOL_VERSION, LDAP_VERSION3);
Error <0x0> = ldap_connect(hLdap, NULL);
Error <0x0> = ldap_get_option(hLdap,LDAP_OPT_SSL,(void*)&lv);
Host supports SSL, SSL cipher strength = 128 bits
Established connection to ldap.example.com.
Retrieving base DSA information...
Result <0>: (null)
Matched DNs:
```

Support Backup LDAP Server for Fail Over

To setup a backup LDAP server:

1. To start, use the same configuration as described above.
2. To configure a backup LDAP server, add a new configuration parameter with EDB-LDAP-HOST-1.

> **Note:** You can configure an unlimited amount of backup LDAP server (EDB-LDAP-HOST-1, EDB-LDAP-HOST-2, ...).

Java Client Single Sign-On (SSO)

The Java Client SSO feature bases on Kerberos. The Java Client uses the Windows logon Ticket Granting Ticket (TGT) to request a service ticket for the Agile e6 system.

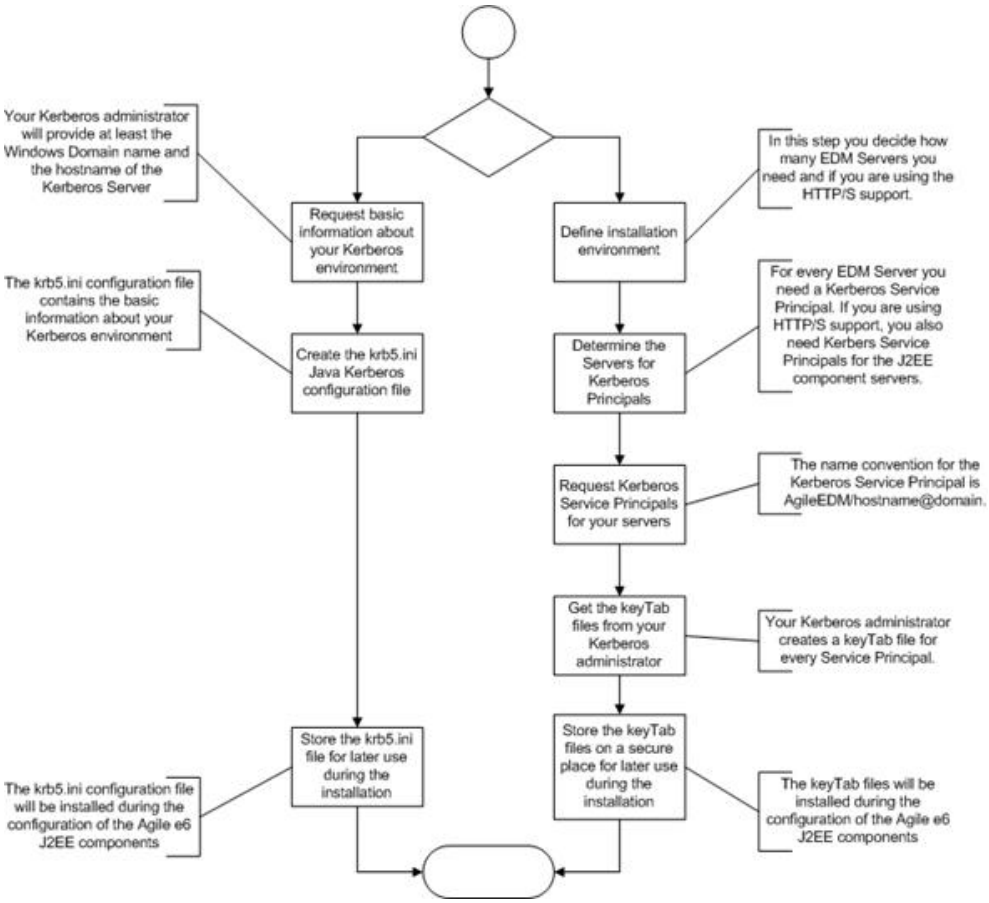
The Agile e6 user has to be mapped to the Windows domain user so that the Agile e6 system can automatically log into the Windows domain user with its mapped default Agile e6 user account.

For more information about user credential mapping refer to the Online Help DataView > Enhanced User Management.

The following sections explain how you can configure your Agile e6 system to support the Kerberos based SSO feature.

Kerberos Prerequisites

The graphic shows an overview of all tasks to prepare the customer infrastructure to use the Kerberos integration.Tasks on the left side show System Administrator tasks, while tasks on the right side are executed by an Agile e6 Administrator.



All examples are based on a Microsoft Windows Domain environment with a domain login and a Domain Controller with an Active Directory.

Some of the tasks are manual steps for installation and configuration of the Kerberos integration. The other tasks are administration tasks of the Windows domain.

In general, there are two tasks which must be completed before you can configure an SSO environment for the Java Client.

1. Collect the information of your company Kerberos infrastructure and create the krb5.ini configuration file for Java.
2. Request the Service Principals and get the corresponding keyTab files from your Kerberos administrator.

Kerberos Infrastructure

The Java Runtime environment of the Agile e6 J2EE components need some basic information about the company Kerberos infrastructure.

Request Basic Information About Your Kerberos Environment

For a standard configuration you need the following information:

- Name of your domain
- Host name of the Kerberos server

This information must be provided by your Kerberos administrator. In the most common cases you will have a Windows domain with a domain controller as Kerberos server.

Java Kerberos Configuration File

> **Note:** If the configuration file already exists, please modify it, otherwise create it new.

The Java Runtime uses the JAAS Kerberos module to verify the Kerberos service tickets provided during the login.

The krb5.ini configuration file describes the settings used by the JAAS Kerberos module. This configuration file is normally located at C:\Windows on a Windows system.

```
[libdefaults]
default_realm = EXAMPLE.COM
dns_lookup_kdc = false
dns_lookup_realm = false
ticket_lifetime = 600
default_tgs_enctypes = aes128-cts des3-cbc-sha1 rc4-hmac des-cbc-md5 des-cbc-crc
default_tkt_enctypes = aes128-cts des3-cbc-sha1 rc4-hmac des-cbc-md5 des-cbc-crc
[realms]
EXAMPLE.COM = {
 kdc = kdc.example.com
 default_domain = EXAMPLE.COM
}
[appdefaults]
autologin = true
forward = true
forwardable = true
encrypt = true
```

Here a short description of the most important settings:

- default_realm

  Default realm which should be used. This is the name of the Windows domain used for login.

- kdc

  The realm definition with the name of the Kerberos server (kdc) and the domain name. The domain name must be in uppercase! The kdc is normally the domain controller.

- default_domain

  In this example the Windows domain is EXAMPLE.COM and the domain controller has the hostname "kdc.example.com."

The configuration file for a Windows domain can use the DNS lookup to locate the KDC for the Windows domain. So it is not necessary to configure the hostname of the KDC, the system finds the KDC for the domain automatically.

For more information about creating a new configuration file, please refer to the following on-line documentations:

- Kerberos Requirements

  http://docs.oracle.com/javase/7/docs/technotes/guides/security/jgss/tutorials/KerberosReq.html

- How to Manually Configure a Kerberos Client

  http://docs.oracle.com/cd/E19253-01/816-4557/setup-341/index.html

Store the Java Kerberos Configuration File

To store the configuration file krb5.ini for later usage during the Agile e6 J2EE components configuration, we recommend storing it in a secret location.

When installing the configuration file krb5.ini, it can be stored in two different places:

- C:\Windows

  The default location.

- Application specific

  To minimize possible side effects with other Java application, it is possible to install the Java Kerberos configuration file application specific.

How this can be achieved is described in section Agile e6 J2EE Components Configuration.

Service Principals and keyTab Files

The Kerberos authentication for services works with so called Service Principal Names (SPN). These are special service accounts on the Kerberos server.

The Kerberos administrator needs the hostnames where the service is running and the service name to create such a Service Principal Name.

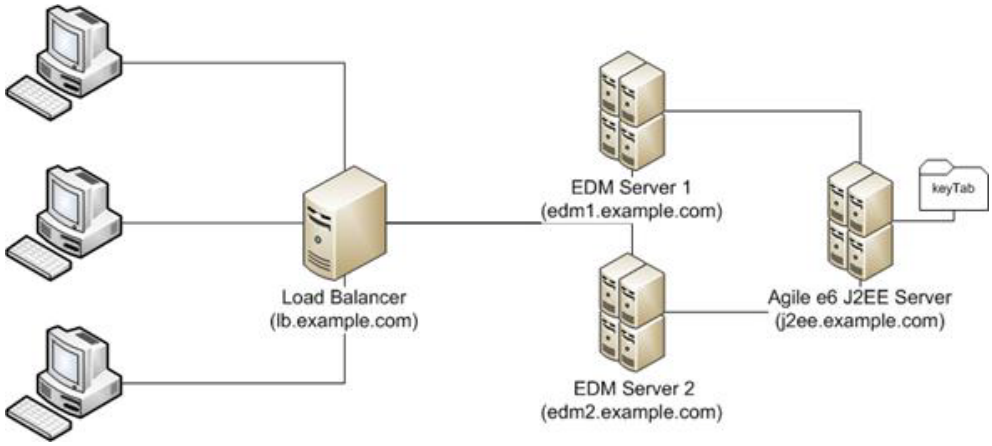
In case of Agile e6, the SPN pattern is AgileEDM/hostname@DOMAIN.

Define Installation Environment

As the used environment defines for which server(s) you need to request a Service Principal Name, the Agile e6 environment needs to be defined first.

The following picture shows some common installation bases which demonstrate how to identify the servers which need the Service Principal Name.

Example Configurations - Load Balancer

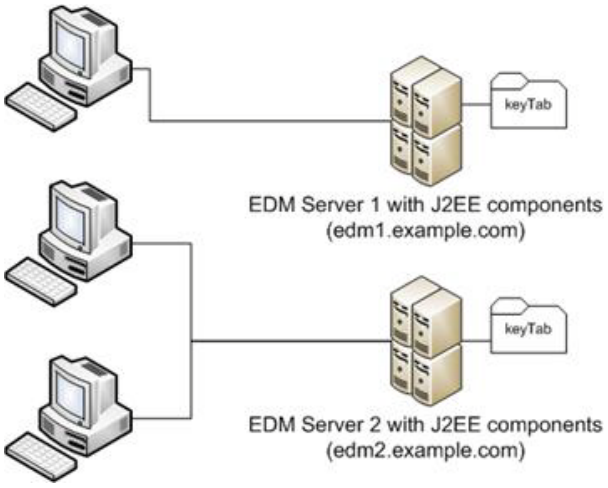


In this example, the different clients connect to a load balancer server which routes the clients to different EDM Servers. The Agile e6 J2EE components are deployed on an extra server which is used by the different EDM servers.

In such a configuration, the load balancer is the entry point for the different clients which are running the Agile e6 Java Client. Therefore, the Service Principal has to be created for the load balancer server and not for the EDM servers which are behind the load balancer.

The keyTab for the load balancer has to be installed on the Agile e6 J2EE server for the verification of the Kerberos service tickets.

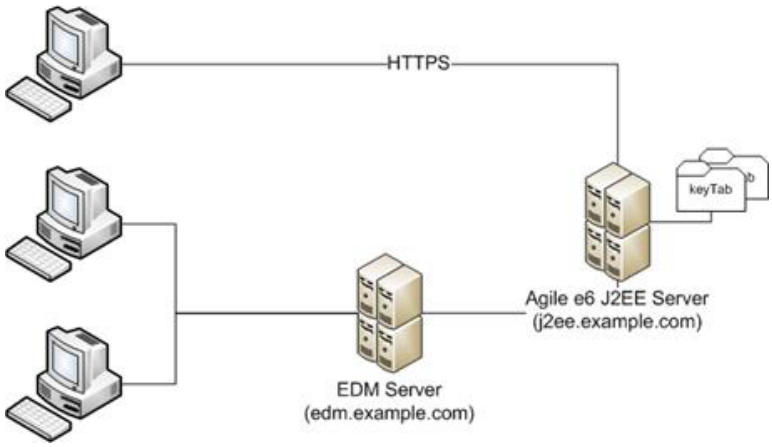
Example: Different EDM Servers



In this configuration, some client machines are connecting to the EDM Server 1 and the other client machines are using the EDM Server 2, the Agile e6 J2EE components are deployed on these servers, too.

In this use case we need two Service Principals, one for each EDM server and the keyTab files have to be installed on corresponding Agile e6 J2EE components deployment.

Example: Different protocols to access the Agile e6 Server



This environment has one EDM Server which can be accessed in different ways. Some client machines are using the socket based ECI protocol to access the EDM Server directly.

Other client machines are connecting via HTTP/S (PLM-API) to the Agile e6 J2EE server to login into Agile e6.

In this scenario we need two Service Principals, one for the EDM server, and one for the Agile e6 J2EE server. Both keyTab files have to be installed on the Agile e6 J2EE server.

Determine the Servers for Kerberos Principals

As you have seen in the examples, not all servers need a Service Principal Name. It depends on how the Java Clients are connected to the Agile e6 system.

In general, the server which is used for the login (hostname to access the Agile e6 system) needs the Service Principal Name and the keyTab needs to be installed on the corresponding Agile e6 J2EE components deployment.

Required Service Principal Names for the Examples

In the examples, the Windows domain is EXAMPLE.COM, therefore the Service Principal names (SPN) are:

- Load Balancer Example
  - AgileEDM/lb.example.com@EXAMPLE.COM
- Different Agile e6 Servers Example
  - AgileEDM/edm1.example.com@EXAMPLE.COM
  - AgileEDM/edm2.example.com@EXAMPLE.COM
- Different Protocols to access the Agile e6 Server Example
  - AgileEDM/edm.example.com@EXAMPLE.COM
  - AgileEDM/j2ee.example.com@EXAMPLE.COM

Request Kerberos Service Principals for Your Servers

The Active Directory administrator has to create a Service Principal for the Login Server.

To achieve this; the administrator needs to create a mapping user which will be used to map to the Service Principal.

If the customer infrastructure needs more than one Service Principal Name (one for each login server), it is recommended to add the server name to the mapping user name.

For example, you want to create a mapping user to map to a Service Principal Name for the EDM Server, which is running on edm.example.com.

1. Create a mapping user like e.g. AgileEDM_edm.example.com.
2. You can see for which service on which host the mapping user is used.

The Service Principal Name itself has to be named with this pattern:

AgileEDM/hostname@DOMAIN in this example it would be AgileEDM/edm.example.com@EXAMPLE.COM.

Get the keyTab Files From Your Kerberos Administrator

By mapping the user account, created for the Service Principal Name, a keyTab file is written, which is needed by the Agile e6 J2EE components to verify Kerberos tickets for that Service Principal Name.

Create keyTab for Service Principal

With the ktpass util program from Mircosoft, you can map the mapping user to the Service Principal Name and export the key table which can be used by the the Agile e6 J2EE components to verify the Kerberos service tickets for that Service Principal Name.

```
ktpass /princ AgileEDM/edm.example.com@EXAMPLE.COM /mapuser
AgileEDM_edm.example.com@EXAMPLE.COM /pass mapuserpassword /out
krb5.keytab /crypto all /ptype KRB5_NT_PRINCIPAL /mapop set /kvno 0
Targeting domain controller: kdc.example.com
Successfully mapped AgileEDM/edm.example.com to
AgileEDM_edm.example.com.
Password succesfully set!
Key created.
Key created.
Key created.
Key created.
Key created.
Output keytab to krb5.keytab:
Keytab version: 0x502
keysize 71 AgileEDM/edm.example.com@EXAMPLE.COM ptype 1 (KRB5_NT_PRINCIPAL) vno 0
etype 0x1 (DES-CBC-CRC) keylength 8 (0x7f085b1f62498a08)
keysize 71 AgileEDM/edm.example.com@EXAMPLE.COM ptype 1 (KRB5_NT_PRINCIPAL) vno 0
etype 0x3 (DES-CBC-MD5) keylength 8 (0x7f085b1f62498a08)
keysize 79 AgileEDM/edm.example.com@EXAMPLE.COM ptype 1 (KRB5_NT_PRINCIPAL) vno 0
etype 0x17 (RC4-HMAC) keylength 16 (0x0d2c584ff099e7eda13b6f5312f14782)
keysize 95 AgileEDM/edm.example.com@EXAMPLE.COM ptype 1 (KRB5_NT_PRINCIPAL) vno 0
etype 0x12 (AES256-SHA1) keylength 32
(0x9d208948196e2d869d9d9648fc7f478aad2244940ab17344939d4f7b33abdea4)
keysize 79 AgileEDM/edm.example.com@EXAMPLE.COM ptype 1 (KRB5_NT_PRINCIPAL) vno 0
etype 0x11 (AES128-SHA1) keylength 16 (0x61150efafb64635fd247b677a18e4840)
```

In this example, the keytab file contains all supported encryption algorithms of the KDC.

For Windows Domain Controllers, the RC4-HMAC is the standard.

See MSDN help for more information.

Store the keyTab Files

To store the keyTab files for later usage during the Agile e6 J2EE components configuration, we recommend storing it securely. It may the best practice to name the keyTab file so that it is clear for which Service Principal Name it is used for.

For example, the keyTab for AgileEDM/edm.example.com@EXAMPLE.COM can be named like edm.example.com.keytab.

You can also use different sub directories to organize your keyTab files for the several login servers.

EDM Server Configuration

The EDM Server delegates the Kerberos login request to the security module, located within the Agile e6 J2EE components, which are deployed on the WebLogic server.

Agile e6 J2EE Components Configuration

The following graphic shows the different steps to configure the Agile e6 J2EE components for the Kerberos integration.

1. Create secured directory
2. Install Java Kerberos configuration file
3. Install keytab file(s)
4. Configure your Service Principal(s)
5. Populate Kerberos configuration to WebLogic Server
6. Install Oracle Wallet
7. Re-deploy
8. Restart domain

Create Secured Directory

The Agile e6 J2EE server needs several configuration files and keytab files exported from the Kerberos server (Active Directory).

These files contain security relevant information and must be protected.

As best practice, you can store the files within the WebLogic installation at the following location:

```
%DOMAIN_HOME%\..\..\security
```

Example:

Your domain directory of your Agile e6 J2EE server installation may be located at:

- D:\Oracle\Middleware_WLS1212\user_projects\domains\eSeries_domain

  Then you can create the secure directory at:

- D:\Oracle\Middleware_WLS1212\user_projects\security

Note: Make sure that the WebLogic runtime user has access only to this directory!

Do not create the secure directory within the eSeries_domain, because it may be deleted by the admin client.

Install Java Kerberos Configuration File

During the preparation of the Kerberos prerequisites, the Java Kerberos Configuration file (krb5.ini) was created.

1. Copy that file into the new secure directory.

Install keyTab File(s)

You got the keyTab file(s) from your Kerberos administrator.

1. Copy them into the new secure directory.

   You can copy them all into the secure directory with different file names or create a sub directory for each keyTab file.

In the following are some examples of secure directory configurations with more than one keyTab file.

All in one Directory

- krb5.ini
- edm1.example.com.keytab
- edm2.example.com.keytab
- j2ee.example.com.keytab
- jaas.conf

Sub Directories

- conf
  - krb5.ini
  - jaas.conf
- keytab
  - edm1.example.com
    - krb5.keytab
  - edm2.example.com
    - krb5.keytab
  - j2ee.example.com
    - krb5.keytab

Configure Your Service Principal Name(s)

In the example above, the jaas.conf configuration file is listed. This configuration file allows binding the Service Principal Names to their keyTab file.

In this section we will create that configuration file.

- Template

This template can be used to create the jaas.conf configuration file:

```
<hostname> {
  com.sun.security.auth.module.Krb5LoginModule required
  useKeyTab=true
  keyTab="<full path to the keytab file>"
  storeKey=true doNotPrompt=false
  isInitiator=false
  principal="<Service Principal>";
};
```

- Template for IBM AIX

  On IBM AIX the jaas.conf file looks different:

```
<hostname> {
  com.ibm.security.auth.module.Krb5LoginModule required
  credsType=acceptor
  useKeytab="<full path to the keytab file>"
  principal="<Service Principal>";
};
```

  For each Service Principal Name the jaas.conf file has to contain such a section.

  The common settings are pre-configured in this example, for detailed information for the possible settings please refer to the JAAS Login Configuration File documentation on docs.oracle.com.

You need to provide the following information:

- The Agile e6 J2EE component uses the hostname as key to find the entry in the configuration.
- The full path to the keyTab file.

  Note: The full path always has to use the UNIX style syntax "/"!

- The Service Principal Name without the domain.

Example: "Sub directories" example:

```
edm1.example.com {
  com.sun.security.auth.module.Krb5LoginModule required
  useKeyTab=true
  keyTab="D:/Oracle/Middleware_WLS1212/user_
projects/security/edm1.example.com/krb5.keytab"
  storeKey=true
  doNotPrompt=false
  isInitiator=false
  principal="AgileEDM/edm1.example.com";
};

edm2.example.com {
  com.sun.security.auth.module.Krb5LoginModule required
  useKeyTab=true
  keyTab="D:/Oracle/Middleware_WLS1212/user_
projects/security/edm2.example.com/krb5.keytab"
  storeKey=true
  doNotPrompt=false
  isInitiator=false
  principal="AgileEDM/edm2.example.com";
};
```

```
j2ee.example.com {
  com.sun.security.auth.module.Krb5LoginModule required
  useKeyTab=true
  keyTab="D:/Oracle/Middleware_WLS1212/user_
projects/security/j2ee.example.com/krb5.keytab"
  storeKey=true
  doNotPrompt=false
  isInitiator=false
  principal="AgileEDM/j2ee.example.com";
};
```

To create the file jaas.conf:

1. Open an editor.
2. For each Principal Service Name add such an entry.
3. Save it into your secure directory with the file name jaas.conf.

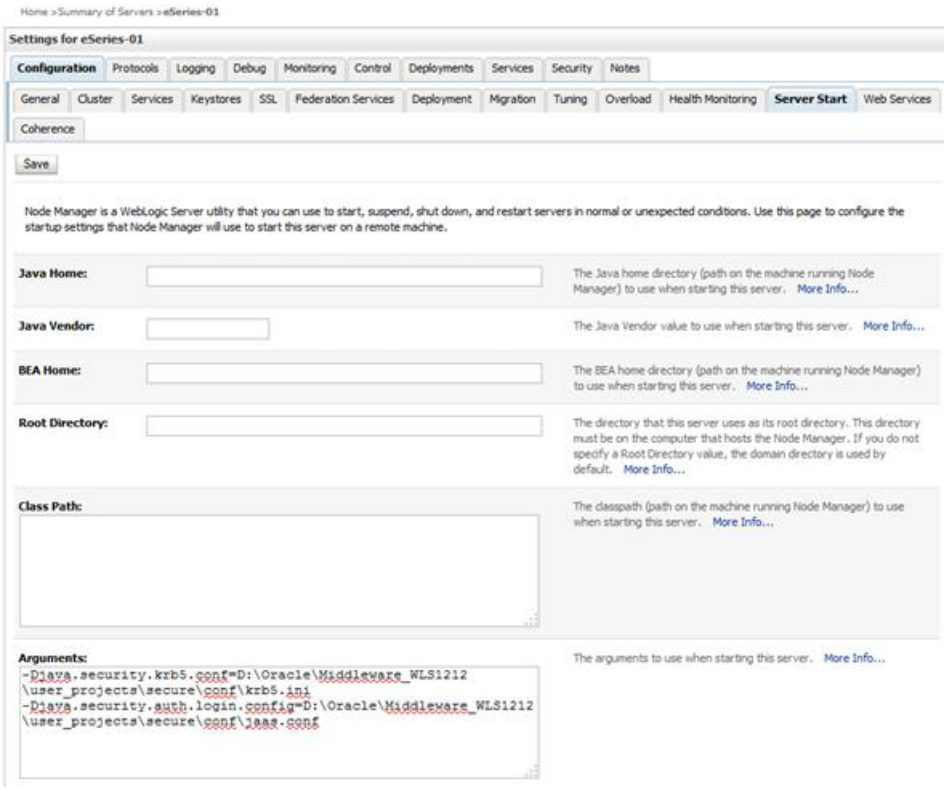
Populate Kerberos Configuration to WebLogic Server

The Kerberos configuration files must be populated to the application domain server of the Agile e6 J2EE server. The Java JAAS module supports 2 system properties which allow configuring the location of the configuration files.

```
-Djava.security.krb5.conf=<full path to the krb5.ini file>
-Djava.security.auth.login.config=<full path to the jaas.conf file>
```

Example:

1. The WebLogic console of the Agile e6 J2EE Server allows setting the security system properties.
2. Navigate via Environments -> Servers to the eSeries server of your application domain.
3. On the tab "Server Start" set the system properties in the Arguments text box.



Example: Script based Example

With the WSLT tool from WebLogic, you can set the system properties like this:

```
cd('/Servers/eSeries-01/ServerStart/eSeries-01'
cmo.setArguments('-Djava.security.krb5.conf=D:\Oracle\Middleware_WLS1212\user_
projects\security\conf\krb5.ini
-Djava.security.auth.login.config=D:\Oracle\Middleware_WLS1212\user_
projects\security\conf\jaas.conf'
activate()
```

Restart the Domain

1. Restart WebLogic to activate the Kerberos configuration.

Troubleshooting

The configuration of Kerberos is very complex. In case Kerberos does not work, here some hints to track down what could be wrong.

Possible Errors - The creation of the Kerberos Ticket failed

- If the user changes the login type to "Kerberos", the user name should be changed to the Kerberos user name automatically. If this does not happen, the Java Client cannot access the TGT of the Windows session.
- If the Java Client shows the error that the SSO ticket cannot be created, it may be a configuration issue within the krb5.ini file.
- Activate the Java Client logging and trace com.agile.security package to get more information.

Tracing

If you get an invalid user/password message, the Kerberos ticket could not be verified or the trusted relationship did not work.

To get more information, activate the EDM Server tracing.

1. Open or create a custom environment start-up script (<application>_cust.cmd) in the %ep_root%\init directory of your EDM Server installation.
2. Add to the custom environment start-up script the following setting:

   ```
   set EP_DEBUG=Main,Epq
   ```

   This activates the server trace including the SQL trace.

To activate the C++ trace:

1. Open the environment XML file <application>.xml at the same location as the start-up script.
2. Edit the TraceConfig attribute of the General section.

Set the name of the trace configuration file.

- The default trace configuration file is located in the %ep_root%\axalant\ini directory and is named as trace.cfg.
- Here an example for the use of the default trace configuration file.

  ```
  TraceConfig="F:D:\plm\axalant\ini\trace.cfg"
  ```

1. Open the trace.cfg and activate the following entry:

   ```
   com::agile::dtv debug
   ```

2. The EDM Server writes a log file into the tmp directory of your EDM Server installation.

Common reasons for an error

- User Mapping

  You can use the SQL statements within the log file to check if a valid user mapping was found.

  1. Search for SELECT on T_USERMAP.

     At least one record has to be found.

- Kerberos ticket verification

  If your Kerberos ticket cannot be verified, you will find the error message in the log file.

  1. Search for PlmLogin to find Kerberos ticket verification errors.

- Trust of relationship

  For errors in the trust of relationship, you will also find the error message in the EDM Server log file. In most cases the reason for errors here are access permission issues with the Oracle wallet files, or inconsistencies between the Oracle wallet provided to the EDM Server, and the Oracle wallet provided to the J2EE components.

Web Service SSO

For a detailed description for the configuration of the Web Services, see the Web Services Guide for Agile e6.2.0.0.

- The remote credential of type LDAP, used for the Web Service call, must have a valid default mapping to an Agile e6 user.
- The mapped Agile e6 user has to have the Web Service flag activated. The Web Service flag is activated in the Agile e6 user management.
- Configuration parameter EDB-WSI-URL must be set to the Web Service URL for the Agile e6 Core Web Service (https://<server>:<port>/CoreServices).
- Other configuration parameters for Web Service Configuration must be set dependent of the called Web Services.

  For more information about the user credential mapping refer to the Online Help DataView > Enhanced User Management.

WebLogic SAML Configuration

The Security Assertion Markup Language (SAML) enables cross-platform authentication between web applications, or Web Services running in a WebLogic server domain and web browsers, or other HTTP clients. WebLogic server supports single sign-on (SSO) based on SAML. When users are authenticated at one site that participates in a single sign-on (SSO) configuration, they are automatically authenticated at other sites in the SSO configuration and do not need to log in separately.

To use SSO based on SAML for Web Service calls, you have to configure a trusted relationship between the WebLogic domains and the application domain server of the Agile e6 J2EE server.

See the Oracle WebLogic Server documentation, "Understanding Security for Oracle WebLogic Server" how to configure SAML.

Agile e6 Web Service target resources that can be configured to accept authentications through SAML assertions:

```
target:*:/CoreServices/BusinessObjectService
target:*:/CoreServices/ConfigurationService
target:*:/CoreServices/DocumentManagementService
target:*:/CoreServices/MetadataService
target:*:/CoreServices/ECService
```

To be able to call the Web Services with SAML authentication, you have to adapt the file web.xml and file weblogic.xml:

1. Copy the files from directory %ep_root%\staging\product\application\<application_name>\WebServices\agile-ws-e6-jws-core.war\WEB-INF\

   to directory:

   ```
   %ep_root%\staging\custom\application\<application_
   name>\WebServices\agile-ws-e6-jws-core.war\WEB-INF\.
   ```
2. Remove in file web.xml sections:
   - <security-constraint>…. </ security-constraint>
   - <login-config>…</login-config>
   - <security-role> … </security-role>
3. Remove in file weblogic.xml section
   - <security-role-assignment> … </ security-role-assignment >

Note: After applying the SAML policy to a Web Service, you are no longer able to call this Web Service without a SAML authentication.

6

Agile e6 Database User and Privileges

The default installation assumes that the Agile e6 software is installed on dedicated servers where no other users have access to the installation.

Predefined Agile e6 User

User in table T_USER and T_GROUP and their group assignment:

User Name	Manager	Profile	Status	Assigned to Group
EDBKERNEL	Yes	MANAGER-PROFIL	Locked	DATAVIEW, DEMOEP, EDB
EDBCUSTO	Yes	MANAGER-PROFIL	Locked	DATAVIEW, EDB
DEMOEP	No	-	Locked	DEMOEP
DEMOEP_M	Yes	MANAGER-PROFIL	Locked	DATAVIEW, DEMOEP, EDB
EDB-RESERVED	Yes	MANAGER-PROFIL	Locked	EDB
DODEKERNEL	Yes	-	Locked	DODE-DEVELOPER
EDB-EIP	Yes	MANAGER-PROFIL	Locked	EDB
MANAGER	Yes	-	Active	DATAVIEW

The following example shows how a user can have several roles assigned that were defined in the so-called Job Functions.

User Name	Role	Job Function	Privileges
EDBCUSTO	EDB-ORGANIZATION-MNG	DefaultOrgMng	EDB-ORG-CPY Copy a company/department
EDBCUSTO	EDB-PROJECT-MNG	DefaultProjectMng	EDB-POS-DEL Delete a position or project team member

The following table shows assigned Job Functions to the user EDBCUSTO.

Note: EDBCUSTO is the only user with assigned Job Functions.

Job Function	Description
DefaultOrgMng	Enabling EDBCUSTO to initiate organizations
DefaultProjectMng	Enabling EDBCUSTO to initiate projects
DefaultRoleMng	Enabling EDBCUSTO to define roles, privileges and job functions
Txt-Manager-2	Manager for Txt-Management

Windows Users

The installation has to be started with a user who has Administration rights to create users and services. This user will be later referred to as the Installation User.

> **Note:** After the installation is done, this user should no longer have Administration rights because the AdminClient service has to run under this account to modify the existing installation. This task will not require Administration rights.

Depending on the installed components, there will be two users which will be created during the installation:

- The runtime user for the following services which requires no privileged permissions. This user will be referred to as the Runtime User:
  - FMS Java Daemon
  - Java Daemon
  - Portmapper
- The user running the File Server. This user requires Administrative rights to secure its own data directories. This user will be referred to as the File Server User.

UNIX Users

The installation on UNIX requires no special permissions during the installation and should be started as an unprivileged user.

> **Note:** Should not be started by the root user.

To secure the installation, there should be two user accounts created analog to the Windows users which will be created during the Agile e6 installation:

- Runtime User
- File Server User

Default Installation Permissions

This section describes the directory access permission after an installation.

> **Note:** No other users or groups have access permissions to these directories.

Windows

Directory	Access Type	Access Users/Groups
%ALLUSERSPROFILE%\agile\installer\6.2.0	Full Access	Installation User Administrators Group
E6 Installation Destination (ep_root)	Full Access	Installation User Administrators Group Runtime User
File Server Destination	Full Access	Installation User Administrators Group File Server User
Enterprise Integration Platform Destination	Full Access	Installation User Administrators Group File Server User

UNIX

Directory	Access Type	Access Users/Groups
${HOME}/.agile/installer/6.2.0	Full Access	Installation User
E6 Installation Destination (ep_root)	Full Access	Installation User
File Server Destination	Full Access	Installation User
Enterprise Integration Platform Destination	Full Access	Installation User

Detailed Access Permissions

This section describes the minimum access permissions for specific users and directories.

Installation User

This user needs to have full access to the Agile e6 installation to administrate the installation, e.g. applying hot fixes, modifying or creating a new application.

> **Note:** The Agile e6 installation includes here the native EDM Server (ep_root), the File Server, and the WebLogic user domains.

This user needs to have exclusive full access to the following directories, too.

> **Note:** No additional users should have access to the following directories.

- Windows

  ```
  %ALLUSERSPROFILE%\agile
  ```

- UNIX

  ```
  ${HOME}/.agile
  ```

Runtime User

This user requires read only and execute permissions for the native EDM Server or dedicated DFM installation directory.

In addition, this user requires write and delete permissions for the following directories:

- Native EDM Server
  - ep_root/axalant/dmp
  - ep_root/tmp
  - ep_root/<application>/lck
- DFM location
  - <tomat_server_root>/logs
  - <tomat_server_root>/webapps
  - <tomat_server_root>/work
  - ep_root/tmp
- EIP Location
  - – <eip_root>/logs
  - – <eip_root>/tmp

File Server User

This user only requires full access to the File Server root directory and below it.

Example How to Use Strict Access Permissions

This section describes how to remove the access permissions for other users, and remove unneeded permissions for the runtime user.

Note: This also applies to the Enterprise Integration Platform installation location.

Windows

The Windows command icacls.exe can be used to add or remove access permissions to directories.

Execute the following commands in a command shell with the installation user.

1. Remove the administrator access.

Note: Replace <ep_root> with the path to the Agile e6 installation directory.

```
icacls.exe <ep_root> /remove:g BUILTIN\Administrators
icacls.exe %ALLUSERSPROFILE%\agile\installer\6.2.0 /remove:g
BUILTIN\Administrators
```

Note: The above command requires changing the Log On Account for the AdminClient service.

1. Start the Services Administration Configuration.
2. Open the properties of the Apache Tomcat AgileAdminClient service.
3. Switch to the tab Log On.
4. Change the local system account to this account, and fill in the data of your installation user.

2. Remove the Administrators group access for the File Server directory:

Note: Replace <fms_root> with the path to the File Server directory.

```
icacls.exe <fms_root> /remove:g BUILTIN\Administrators
```

3. Restrict the access permission for the runtime user.

Note: Replace <ep_root> with the path to the Agile e6 installation directory and replace <RUNTIME_USER> with the name of the runtime user. Replace <application> with the name of your Agile e6 application.

1. Remove the access permission for the Runtime User (<RUNTIME_USER>) first.

```
icacls.exe <ep_root> /remove:g <RUNTIME_USER>
```

2. Add the default read and execute permissions for the runtime user:

```
icacls.exe <ep_root> /grant <RUNTIME_USER>:(RX)
icacls.exe <ep_root> /grant <RUNTIME_USER>:(OI)(CI)(IO)(RX)
```

3. Add the full access permissions for the runtime user to a selected set of directories:

```
icacls.exe <ep_root>\axlant\dmp /grant <RUNTIME_USER>:(F)
icacls.exe <ep_root>\axalant\dmp /grant <RUNTIME_USER>:(OI)(CI)(IO)(F)
icacls.exe <ep_root>\tmp /grant <RUNTIME_USER>:(F)
icacls.exe <ep_root>\tmp /grant <RUNTIME_USER>:(OI)(CI)(IO)(F)
icacls.exe <ep_root>\<application>\lck /grant <RUNTIME_USER>:(F)
icacls.exe <ep_root>\<application>\lck /grant <RUNTIME_
USER>:(OI)(CI)(IO)(F)
```

Note: Permissions for additional applications which are created with the Administration Client or the batch installation need to be granted manually.

UNIX

There are different options to restrict the access, e.g. using ACL or UNIX groups. The following description is for UNIX groups.

Note: Replace <ep_root> with the path to the Agile e6 installation directory.

1. Stop any Agile e6 daemons.
2. Clean up all files in the following directory before changing the process owner from the installation to the runtime user:

```
rm <ep_root>/axalant/dmp/*
rm <ep_root>/tmp/*
rm <ep_root>/<application>/lck/*
```

3. Create a UNIX group, e.g. plmgrp.
4. Add the installation user to the new group from above.
5. Create a new UNIX user, e.g. plmrun and add this user to the newly created group.
6. Change the default group file/directory access permission of ep_root:

```
chgrp -R plmgrp <ep_root>
chmod -R g=rx <ep_root>
```

7. Add the full access permissions for the runtime user to a selected set of directories:

```
chmod -R g+w <ep_root>/axalant/dmp
chmod -R g+w <ep_root>/tmp
chmod -R g+w <ep_root>/<application>/lck
```

8. Now you can start the following daemons with the runtime user:
   - FMS Java Daemon (${ep_root}/axalant/scripts/fms_jade)
   - Java Daemon (${ep_root}/axalant/scripts/jade)

7

Securing Ports

The Internet Assigned Numbers Authority (IANA) administrates the port numbers in the range of 0 - 65,535.

When it comes to assigning port numbers for services that are not registered, only port numbers of the so-called dynamic (private) range of 49,152 - 65,535 should be assigned in order to meet minimum security requirements. However, conflicts with already installed applications can occur.

But in practice, numbers in the range from 0-1,023 are protected, and from 1,024 onwards can be used. For example, operating systems should only allow processes with appropriate privileges to open the server ports that are within the given range.

Prior to any installation, you should contact your system administrator in order to evaluate the ports that are already in use by the system and applications. Thus conflicts can be avoided when assigning ports that are used by Agile e6. Additionally, an existing firewall needs to be configured accordingly.

In case a system administrator is not available, a list of currently used TCP- and UDP -ports can be created with the command netstat -a. The RPC ports that are available through PortMapper can be determined using the command rpcinfo -p.

Range of Ports

The port numbers are divided into three ranges:

1. Well-known ports

   The well known ports are those from 0 - 1,023. DCCP well known ports should not be used without IANA registration. The registration procedure is defined in document RFC4340, section 19.9.

2. Registered ports

   The registered ports are those from 1,024 - 49,151. DCCP registered ports should not be used without IANA registration. The registration procedure is defined in document RFC4340, section 19.9.

3. Dynamic and/or private ports

   The dynamic and/or private ports are those from 49,152 - 65,535.

Note: Assignment of a port number does not in any way imply an endorsement of an application or product, and the fact that network traffic is flowing to or from a registered port does not mean that it is "good" traffic. Firewall and system administrators should choose how to configure their systems based on their knowledge of the traffic in question, not whether there is a port number registered or not.

Well Known Port Numbers

The well known ports are assigned by the IANA and on most systems can only be used by system (or root) processes or by programs executed by privileged users.

Ports are used in the TCP [RFC793] to name the ends of logical connections which carry long term conversations. For the purpose of providing services to unknown callers, a service contact port is defined. This list specifies the port used by the server process as its contact port. The contact port is sometimes called the "well-known port".

To the extent possible, the same port assignments are used with the UDP [RFC768]. The range for assigned ports managed by the IANA is 0-1,023.

Registered Port Numbers

The registered ports are listed by the IANA and on most systems can be used by ordinary user processes, or programs executed by ordinary users.

Ports are used in the TCP [RFC793] to name the ends of logical connections which carry long term conversations. For the purpose of providing services to unknown callers, a service contact port is defined. This list specifies the port used by the server process as its contact port.

The IANA registers uses of these ports as a convenience to the community. To the extent possible, these same port assignments are used with the UDP [RFC768]. The Registered Ports are in the range 1,024-49,151.

Dynamic and/or Private Ports

The Dynamic and/or Private Ports are those from 49,152 - 65,535.

Range of Values and Dependencies

Service	Ports (default value)	Dependencies
Sun Portmapper (RPC)	111	Always present under UNIX, under Windows a component of the Agile e6 delivery
Admin Server	■ HTTP (8030) ■ Shutdown Port (8006) ■ AJP 1.3 Port (8010)	
Java Daemon	■ StandardPort (16087) ■ AdminPort (16088), only local ■ RegistrationPort (16089), only local ■ One free port from the port range per application server (3000-4000)	

Service	Ports (default value)	Dependencies
FileServer	■ RPC port (804257548) ■ Web Fileservice (8088) ■ One free port per client connection	Sun Portmapper Web Presentation Service
e6 Server	Per session one port assigned from the daemon. Concerning security issues and firewall settings the port range used from the Java daemon to select a server port should be defined between 3000 and 4000. This is the default used during installation and is defined in file jade.ini with parameter named PortRange.	Sun Portmapper Business Service File Server
Web Presentation Service (Tomcat)	■ Ajp 1.3 Port (8009) ■ Shutdown Port (8005) ■ Web Client / Web Report Service (8088)	Java Daemon
Web Presentation Service (WebLogic)	Web Client / Web Report Service ■ HTTP (7103) ■ HTTPS (7104)	Java Daemon
Business Service	ECI Port (19997) One free port per connection to the EDM Server	Java Daemon EDM Server SMTP port ■ Unsecure mailing ■ 25 ■ Secure mailing (SSL) ■ 587
Java Client	ECI Topic (4444) Needs to be distinct for each client call and can be set with the start.	Java Daemon EDM Server
Core Web Services	Web Service ECI Port (19998) One free port per connection to the EDM Server	Java Daemon EDM Server Fileservice
Workflow Editor	Business Service (ECI Port)	
Office Suite	DDE/OLE/COM	EDM Server
EIP	■ Admin Port (9876) ■ Log Port (4445) ■ Web Server (8080) ■ synchronous: ECI Server Port (19997) Note Here exists a conflict with the standard ECI port of the Business Service.	Java Daemon EDM Server

8

Securing the Database

Default Setup

This section describes the default setup of the database.

DB Role AGILE_E_ROLE

The database role AGILE_E_ROLE is created once for the entire database. For every application, a database user will be created and the role AGILE_E_ROLE will be assigned to it.

The following privileges are assigned.

- Role - CONNECT

  This is a basic privilege from the database. For further information please refer to the Oracle Database documentation.

- System Privileges

  The following system privileges are assigned to AGILE_E_ROLE

  - CREATE TABLE
  - CREATE VIEW
  - CREATE SYNONYM
  - CREATE DATABASE LINK
  - CREATE SEQUENCE TO
  - GRANT ALTER SESSION
  - CREATE PROCEDURE
  - GRANT CREATE TRIGGER
  - GRANT READ,WRITE ON DIRECTORY ORA_DMP

- Quota Unlimited

  This privilege will be given to the database user for the following tablespaces.

  Note: The names for the tablespaces are default names and can be changed during the installation.

- DEFAULT TABLESPACE "EDB"

  This are Agile e6 specific tablespaces.

  * Data tablepsace: QUOTA UNLIMITED ON "EDB"
  * Index tablespace: QUOTA UNLIMITED ON "EDB_IDX"
  * Index tablespace for temporary objects: QUOTA UNLIMITED ON "EDB_TMPIDX"
  * Tablespace for blob data: QUOTA UNLIMITED ON "EDB_LOB"
  * Tablespace for temporary objects: QUOTA UNLIMITED ON "EDB_TMP"

- TEMPORARY TABLESPACE "TEMP"

  This is a database tablespace for sorting. It contains data from GLOBAL temporary tables

Advanced Setup

Additional modules are supported to setup a secure database. Agile e6 is certified for the following advanced security option.

Note: We recommend performing the setup of a secure database with the help of an Oracle security consultant.

- Transparent data encryption
  - For more information please refer to the Oracle Database 12c documentation, Securing Oracle Database, Advanced Security Guide - Using Transparent Data Encryption.
  - A guideline about how to move the database to a transparent data encryption can be found in the White Paper: Oracle Advanced Security TDE "OneCommand" for Oracle Agile e6.
- Database Vault
  - For more information please refer to the Oracle Database 12c documentation, Securing Oracle Database, Database Vault Administrator's Guide.
  - Example scripts can be found under DVB-for-Agile-e6.zip.

Note: Fur further information about these documents please refer to the Oracle Database Security documentation.

9

Additional Security Relevant Information

For further information please also refer to the following documentations.

Access Rights for User

- Online Help DataView
  - Section Password Protection
  - Section Access on Records
- Online Help Multi-Organization Access Rights
- Online Help Multi-Project Access Rights
- Online Help Role Concept

URL Linking Support

- Online Help Getting Started

Whitelist Mechanism for Masks

- Web Service Guide for Agile e6.2.0.0

Number Variant configuration for ECI Web Service Access

- Web Service Guide for Agile e6.2.0.0

Apache Tomcat Security

- http://tomcat.apache.org/tomcat-7.0-doc/security-howto.html

WebLogic Security

- http://docs.oracle.com/middleware/1212/wls/wls-secure.htm