# Oracle® Enterprise Manager

Application Configuration Console Installation Guide

Release 5.3.2

**E14652-02**

September 2009

ORACLE®

Oracle Enterprise Manager Application Configuration Console Installation Guide Release 5.3.2

E14652-02



Primary Author: James Garrison

# Contents

## List of Figures

## List of Tables

# Preface

This document describes the installation and setup of Application Configuration Console. It defines the system requirements for the product, and includes instructions on validating and setting up the database.

## Audience

This document is intended for administrators and other privileged users who are responsible for installing, configuring, and upgrading Application Configuration Console.

## Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible to all users, including users that are disabled. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at http://www.oracle.com/accessibility/.

### Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

### Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

### Deaf/Hard of Hearing Access to Oracle Support Services

To reach Oracle Support Services, use a telecommunications relay service (TRS) to call Oracle Support at 1.800.223.1711. An Oracle Support Services engineer will handle technical issues and provide customer support according to the Oracle service request process. Information about TRS is available at http://www.fcc.gov/cgb/consumerfacts/trs.html, and a list of phone numbers is available at http://www.fcc.gov/cgb/dro/trsphonebk.html.

## Related Documents

For more information, see the following documents in the Application Configuration Console documentation set:

- *Release Notes*
- *Getting Started*
- *PCI Compliance*
- *Command Line Interface Reference*
- *Performance and Tuning Guide*

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

# 1 Installation Outline

This chapter outlines the major steps required to install the components of Application Configuration Console, including some pre-installation requirements.

> **Note:** Application Configuration Console uses a standalone Subversion (SVN) server for version control. A fresh Application Configuration Console installation on Windows or Linux sets up the SVN server instance automatically. On UNIX, however, you must already have Subversion installed and an SVN repository in place prior to installing Application Configuration Console. If SVN is not in place, contact your UNIX system administrator to ensure proper setup.

## 1.1 Before You Install

Prior to installing the components, you must choose the authentication service that you will use, and possibly configure that service to support Application Configuration Console.

Review Chapter 2 to ensure that you comply with the hardware and software (including third-party) requirements to install, set up, and run Application Configuration Console.

## 1.2 Installation Sequence

The following list outlines the steps to installation:

1. **Validate and Set Up the Database**

   First, you must run the provided database validation script to ensure that your database is ready for Application Configuration Console installation. Then run a SQL*Plus script to configure the database for use by Application Configuration Console. You must run this script on the database server; it cannot be run remotely. See Chapter 3 for more information.

2. **Install the Application Configuration Console Server**

   When you have completed setting up the database, follow the instructions in Chapter 4 to install the Application Configuration Console Server on a Windows or Linux system.

3. **Install Application Configuration Console Clients**

   You must install the Client software (Chapter 6) on all systems that you want to use to load and modify configuration data. Users also can access Application Configuration Console data through Web Reports, a browser-based reporting facility. Users who access Web Reports do not need the Client.

4. **Install Automation Modules**

   If you have purchased any automation modules with Application Configuration Console, you install them (Chapter 7) after completing the Server and Client installations.

5. **Install the Windows Resource Extensions**

   If you have purchased the Windows Resource Extensions (WRE) with Application Configuration Console, you install it (Chapter 8) after completing the Server and Client installations.

Following installation, if you want to redeploy servers, follow the instructions in Appendix D.

# 2

# System Requirements

This chapter details the system requirements to install and effectively use Application Configuration Console. System requirements are deemed to be the minimum and recommended software and hardware specifications, including all third-party applications required to support Application Configuration Console.

## 2.1 Application Configuration Console Server

Table 2–1 lists the system requirements necessary to support Application Configuration Console's Core Server, also known as the primary server, and the tracking and Web Reports servers, also known as the secondary servers. An installation of the Core Server includes separate deployments of the primary server and secondary servers. These secondary servers are candidates for redeployment to other instances of a Tomcat JVM on separate hosts, in which case Table 2–2 lists the system requirements necessary to support them.

***Table 2–1 System Requirements for the Core Server***

| Core Server (Primary Server) | | | |
|---|---|---|---|
| Operating System | Windows Server 2003 | Red Hat Enterprise Linux (4.0, 5.0)<br>SUSE Linux Enterprise Server (10) | Solaris (9, 10) |
| Processor | 3 GHz dual processor | | UltraSPARC IV or greater, 1 GHz dual processor |
| Installed Memory | 2 GB (minimum)<br>3 GB (recommended) | | |
| Available Memory | 2 GB (minimum)<br>2.8 GB (recommended) | | |
| Disk Space | Minimum 40 GB (500 MB for installation, plus space for repository) | | |
| Third-Party Software | See Table 2–5 | | |

## 2.2 Application Configuration Console Secondary Servers

Redeployment of secondary servers, which is optional, has implications for performance and scalability. It reduces the load on the primary server so that response times are faster for other tasks, such as compare and provision operations. You can also deploy multiple secondary tracking servers, based on the original tracking server deployment, to improve throughput of tracking operations.

Table 2–2 lists the system requirements necessary to support secondary server instances. The requirements listed in Table 1 are sufficient to support both the primary and secondary servers.

***Table 2–2 System Requirements for the Secondary Servers***

| Secondary Servers | | | |
|---|---|---|---|
| Operating System | Windows Server 2003 | Red Hat Enterprise Linux (4.0, 5.0)<br>SUSE Linux Enterprise Server (10) | Solaris (9, 10) |
| Processor | 3 GHz dual processor | | UltraSPARC IV or greater, 1 GHz dual processor |
| Installed Memory | 2 GB (minimum)<br>3 GB (recommended) | | |
| Available Memory | 2 GB (minimum)<br>2.8 GB (recommended) | | |
| Disk Space | 500 MB | | |
| Third-Party Software | See Table 2–5 | | |
| Additional Requirements | OS must match the primary server OS<br>Secondary server directory structure must match that of the primary server (for example: `/user/oracle/oacc/server`) | | |

## 2.3 Oracle Database Server

Table 2–3 lists the system requirements necessary to support the Oracle database server needed to run Application Configuration Console.

***Table 2–3 System Requirements for the Oracle Database Server***

| Oracle Database Server | |
|---|---|
| Supported RDBMS versions | 10g (10.2.0.3 - Standard or Enterprise Edition)<br>11g recommended (11.1.0.1 - Standard or Enterprise Edition) |
| Operating System | See Oracle documentation |
| Processor | 3 GHz dual processor |
| Installed Memory | 2 GB (minimum)<br>3 GB (recommended) |
| Available Memory | 1.8 GB (minimum)<br>2.8 GB (recommended |

*Table 2–3 (Cont.) System Requirements for the Oracle Database Server*

| Oracle Database Server | |
|---|---|
| Disk Space | 5 GB (minimum)<br>10 GB (recommended) |
| Oracle Settings | SGA settings should be as follows:<br>■ Shared pool-512MB<br>■ Buffer cache-512MB<br>■ Large pool-8MB<br>■ Java pool-256MB minimum; 1GB recommended<br>Aggregate PGA target-512MB |
| Additional Requirements | The Oracle installation must have embedded Java support (JServer option); this is built in to 11g. |

## 2.4 Application Configuration Console Client

Table 2–4 lists the system requirements necessary to support the Client.

*Table 2–4 System Requirements for the Client*

| Application Configuration Console Client | |
|---|---|
| Operating System | Windows XP Professional<br>Windows 2003<br>Vista |
| Processor | 2 GHz |
| Installed Memory | 1 GB (minimum)<br>1.5 GB (recommended) |
| Available Memory | 768 MB (minimum)<br>1 GB (recommended |
| Disk Space | 300 MB |
| Third-party Software | See Table 2–5 |

## 2.5 Third-Party Software

Table 2–5 lists the third-party software requirements for Application Configuration Console.

***Table 2–5 Application Configuration Console Third-Party Software Requirements***

| This Component | Requires These Third-Party Components |
|---|---|
| Primary Server | ■ Sun Microsystems JDK 1.6.0_01*<br>■ Directory service (the Core Server must be able to connect to a directory service); LDAP 2.0 or later, or Microsoft Windows Active Directory 2000 or 2003<br>■ Mail Server (Core Server must be able to connect to the mail server)<br>■ Apache Tomcat 6.0.14 (embedded with the software)<br>■ SVN 1.4.4 server and repository (embedded with the software, except for the Solaris version)<br>■ Java CIFS Client Library (JCIFS) Version 1.3.12 if you want UNC Support<br>■ Business Intelligence and Reporting Tools (BIRT) Runtime Version 2.5.0 (embedded with Web Reports) |
| Secondary Servers | If you follow the instructions in Appendix D on redeploying secondary servers, everything you need in terms of third-party software will be in place in the copied directory structure. The redeployments call in to services such as SVN, LDAP or AD, and e-mail, so there is no physical requirement on the secondary server host to support these services. |
| Client | ■ Sun Microsystems JDK1.6.0_01*<br>■ Eclipse Runtime Platform 3.3.x (embedded with the software)<br>■ Microsoft Internet Explorer 7 or 8 (for Web Reports)<br>■ Mozilla Firefox 2 or 3 (for Web Reports) |
| WebLogic Automation Module | WebLogic Server (8, 9, or 10) installed on the Core Server |
| WebSphere Automation Module | WebSphere Deployment Manager (6, 6.1, or 7) installed on the Core Server |
| Windows Resource Extensions | OpenSSH for Windows 3.8.1p1-1 installed on the machine hosting the WRE Proxy Service (or an equivalent secure connection, properly configured) |
| *And all subsequent patch releases issued for 1.6.0 | |

## 2.6 Requirements for Remotely Managed Systems

As part of your planning and preparation for Application Configuration Console installation, you should also ensure that support for remotely managed systems (also known as managed servers/target systems) is in place.

Application Configuration Console requires FTP or SSH access to the remote systems that it will manage. The configured user accounts on the target systems will require read and write permissions on files and directories that contain configuration data to be imported into Integrity.

Integrity offers the following **Authentication Types** support for SSH:

- Username and password
- SSH certificates (public and private keys)
- Secure provision (`sudo` and `su` to write content out to the file system)

Oracle recommends use of SSH certificates which requires that you create a dedicated user account for the application on the Application Configuration Console Server and all target systems.

Also note the following SSH/SFTP requirements for target systems:

- If using SSH certificates, you must use OpenSSH to generate certificates.
- SSH support requires that the SFTP subsystem be enabled on the targeted systems. The SSH configuration file (`/etc/ssh/sshd_config`) should contain a line similar to the following:

  ```
  subsystem sftp /usr/lib/ssh/sftp-server
  ```

# 3

# Validate and Set Up the Database

First, you must run the provided database validation script to ensure that your database is ready for Application Configuration Console installation. Then configure the database for use by Application Configuration Console by running a SQL configuration script.

> **Note:** Run these scripts on the database server; do not run them remotely.

## 3.1 System Requirements

Application Configuration Console has specific hardware and software requirements for the database host, as well as recommended initial configuration settings for optimal performance. Ensure that you comply with the requirements and recommendations as stated in Table 2–3 in Chapter 2.

## 3.2 Oracle Database Input Validation Procedure

You must run a script to perform database input validation. This script must be run on the Oracle server; it cannot be run remotely.

1. From the Application Configuration Console installation CD, extract `oracle_db_setup.zip` to a temporary directory on the Oracle server, such as `C:\oaccdbInstall`. *The path to the unzip directory cannot contain spaces*.

   Before proceeding, ensure that your environment complies with the following requirements:

   – The `$ORACLE_HOME/bin` directory must be in the user's PATH in order for the validation script to perform all testing successfully.

   – (**Linux/UNIX**) Grant the Oracle user read write access to the scripts directory of the unzipped `oracle_db_setup.zip`, for example, `chmod 777 /tmp/db/oracle/scripts`. Grant executable permissions to the `*.sh` scripts, for example `chmod +x /tmp/db/oracle/scripts/*.sh`. Sudo to the Oracle user: `sudo - oracle`.

2. From the command line on the Oracle server, change directory to the unzipped `db\oracle\scripts` directory, for example:

   ```
   cd C:\OaccdbInstall\db\oracle\scripts
   ```

3. Run the script appropriate to your platform:

```
validate_oracle_w32.bat (Windows)
validate_oracle_lin.sh (Linux)
validate_oracle_sol.sh (UNIX)
```

   To accept a default value for those prompts that display a default, press Enter.

4. Type `i` to perform preinstallation validation (`u` is for preupgrade validation, which is described in Appendix A).

```
Validation test scope? (i/u): i
```

5. Type the full path to the lib directory under the unzipped directory. The path must not contain lib, spaces, or a trailing (back)slash.

```
Path to the "lib" directory (no default): unzip_dir\db\oracle
```

6. Type the name of the Oracle Service that you want to use with Application Configuration Console. If you are using the Application Configuration Console-supplied database, take the default.

```
Oracle Service Name(default=OACCSERV): servicename
```

7. Type the Oracle System username and password at the prompts.

```
System Username (default=SYSTEM):            username
System Password (no default)                : password
```

8. Type an Oracle username and password for the Application Configuration Console user. Since these values are not available until you run the database setup procedure, no validation occurs for the values specified here. Note, however, that this username must be entered in all uppercase letters.

```
Application Configuration Console Username (default=OACCUSER): oaccusername
Application Configuration Console Password                   : oaccpassword
```

9. Type the name of the default tablespace to be used by Application Configuration Console.

```
Application Configuration Console Default Tablespace (default=USERS):
tablespace
```

10. Type the name of the temporary tablespace to be used by Application Configuration Console.

```
Application Configuration Console Temporary Tablespace (default=TEMP):
tablespace
```

11. In the directory from which you ran the script (the current directory), open the validate_oracle.out log file and look for the following entry at the end of the file:

```
SUCCESS.  You are ready to run the setup_db.sql script.
```

    If the log reports errors, do not continue with database setup. Notify your system administrator and contact Oracle Support.

## 3.3 Oracle Database Set Up Procedure

As you run the script to set up the database, specify values for the prompts based on what you entered during validation. You can find these values, excluding passwords, in the `validate_oracle.out` log file. If the file contains multiple entries, use the values at the end, as they are the most recent. Here is a sample output from the log file:

```
...
Oracle Service                                          : OACCSERV
System username                                         : SYSTEM
System password                                         : ******
Application Configuration Console username              : OACCUSER
Application Configuration Console password              : ******
Application Configuration Console default tablespace    : USERS
Application Configuration Console temporary tablespace : TEMP
...
Validating stored procedures directory: C:\OaccdbInstall\db\oracle
...
```

Perform database setup as follows:

1. From the command line on the unzipped `db\oracle\scripts` directory, start the SQL configuration with this command:

   ```
   $ sqlplus /nolog @setup_db.sql
   Please specify values for the following:
   ```

2. Type the full path to the lib directory under the unzipped directory. The path must not contain lib, spaces, or a trailing (back)slash.

   ```
   Path to the "lib" directory (no default): unzip_dir\db\oracle
   ```

3. Type the name of the Oracle Service that you want to use with Application Configuration Console. If you are using the supplied database, take the default.

   ```
   Oracle Service (default=OACCSERV): servicename
   ```

4. Type the Oracle System username and password at the prompts.

   ```
   System Username (default=SYSTEM)            : username
   System Password (no default)                : password
   ```

5. Type an Oracle username and password for the Application Configuration Console user. This username must be entered in all uppercase letters.

   ```
   Application Configuration Console Username (default=OACCUSER): oaccusername
   Application Configuration Console Password                   : oaccpassword
   ```

6. Type the name of the default tablespace to be used by Application Configuration Console.

   ```
   Application Configuration Console Default Tablespace (default=USERS):
   tablespace
   ```

7. Type the name of the temporary tablespace to be used by Application Configuration Console.

   ```
   Application Configuration Console Temporary Tablespace (default=TEMP):
   tablespace
   ```

You should see "Connected" followed by a long sequence of SQL configuration commands and their results. You can safely ignore any error messages about tables or views that don't exist.

This procedure creates the Application Configuration Console-specific Oracle username (OACCUSER). You must provide the same username and password during Application Configuration Console (Core) Server installation.

You should optimize the database after you have installed the Core Server and loaded some data, as explained in "Optimize Database Performance" on page 29.

Note the existence of the setup_db.out file in the unzipped scripts directory (C:\OaccdbInstall\db\oracle\scripts). This file contains information that can be useful to technical support in troubleshooting situations. In fact, you can check the file to ensure successful setup. Specifically, you want to look for lines similar to the following:

```
SQL> @@test_stored_procs.sql;
SQL> select MV_UTIL_PKG.GET_RANDOM_STRING() from MV_ATOM
  2  /

MV_UTIL_PKG.GET_RANDOM_STRING()
--------------------------------------------------------------------------------
Stored procedure setup successful.

SQL>
SQL> select MV_UTIL_PKG.GET_STRING('Stored procedure setup successful.') from MV_
ATOM
  2  /

MV_UTIL_PKG.GET_STRING('STOREDPROCEDURESETUPSUCCESSFUL.')
--------------------------------------------------------------------------------
Stored procedure setup successful.
```

If you don't see the Stored procedure setup successful line, one thing you might try to resolve the issue is to run the query below to verify the correct ownership of some of the database objects used by Application Configuration Console.

1. Connect to the database as system/system with this command:

   ```
   $ sqlplus system/system@oaccserv
   ```

2. Run the following query:

   ```
   select substr(owner,1,15) "OWNER",
   substr(dbms_java.longname(object_name),1,50) "OBJECT_NAME",
   status from all_objects where object_type='JAVA CLASS' and
   dbms_java.longname(object_name) like '%DocumentBuilder%';
   ```

3. The results should be:

   ```
   OWNER      OBJECT_NAME                                      STATUS
   ------------------------------------------------------------
   SYS        javax/xml/parsers/DocumentBuilder         VALID
   SYS        oracle/xml/jaxp/JXDocumentBuilder         VALID
   SYS        oracle/xml/jaxp/JXDocumentBuilderFactory  VALID
   SYS        oracle/xml/parser/v2/XSLDocumentBuilder   VALID
   SYS        javax/xml/parsers/DocumentBuilderFactory  VALID
   SYS        oracle/xml/parser/v2/DocumentBuilder      VALID
   ```

4. Look in the OBJECT_NAME column for any org/apache/xerces/... objects. If there are any, note the owner of that object and run this dropjava command to remove the object from the database:

   ```
   dropjava -user <owner from query>/<password> -synonym <path to xerces.jar>
   ```

5. Rerun the query from Step 2 and verify that there are no `org/apache/xerces/...` objects in the results.

## 3.4 A Note on Database Backups

Oracle expects that you have your own policies on backing up the database, including how you perform an export and import of the database. The information provided here pertains to steps you should take prior to a database import.

The scripts referenced below are bundled with the database setup package (and the database upgrade package). Run the scripts as the SYSTEM USER.

First, if you are importing to a host where an Application Configuration Console schema existed, you must drop the Application Configuration Console-specific user (`OACCUSER`). For example:

```
> sqlplus SYSTEM/SYSTEM@OACCSERV @drop_users.sql OACCUSER
```

Next, you must create the OACC USER, specifying the username and password, and the tablespace. For example:

```
> sqlplus SYSTEM/SYSTEM@OACCSERV @create_users.sql OACCUSER OACCUSER USER
```

Finally, you have to grant the required permissions to the OACC USER. The command must include the full Oracle `lib` path. For example:

```
> sqlplus SYSTEM/SYSTEM@OACCSERV @grant_permissions.sql OACCUSER install_
dir\db\oracle
```

Ensure that the path does not contain the word `lib`, any spaces, or a trailing (back) slash.

---

**Note:** The database and the SVN repository must always be in synch. Whenever you back up the database, be sure to back up the SVN repository. You can find the location of the repository in `server_modules_registry.xml`. See the "url" property value for the module `"com.mvalent.service.system.repository.version.impl.subversion.SvnSessionContext"`.

---

# 4

# Install the Application Configuration Console Server

This chapter tells you how to install the Application Configuration Console Server, also called the Core Server, on either a Windows or Linux server. Instructions in this chapter observe the following convention: `$OACC_INSTALL` refers to the Core Server-installed location, for example, `opt/oracle/oacc/server`.

> **Note:** If you are installing on a Linux server and you do not have physical access to the machine (or if it does not have a CD drive), follow the instructions for Linux Tarball installation in Chapter 5.

## 4.1 System Requirements

The Core Server has specific hardware and software requirements, including third-party software that must be in place prior to installation. Ensure that you comply with the requirements as stated in in Table 2–1 in Chapter 2.

## 4.2 Prerequisites

Before installing the Core Server, the applications and software listed below must be installed and accessible by the host system and you must have the information specified.

### 4.2.1 Oracle

You will need the following information about your database before installing the Core Server:

- Oracle Service Name (SID) (default=`OACCSERV`): _____
- Oracle System User Name (default=`SYSTEM`): _____
- Oracle System Password (default=`SYSTEM`): _____
- Database User Name for Core Server (default=`OACCUSER`): _____
- Database User Password for Core Server (default=`OACCUSER`): _____

Note that the Oracle SID must have a minimum block size of 8K to ensure successful installation of the Core Server. You can check the block size by logging into SQL*Plus as `sysdba` and executing the following command:

```
SQL> select value from v$parameter where name = 'db_block_size';
```

### 4.2.2 LDAP

If you are configuring the Core Server to use LDAP for user authentication, you must have this information about the LDAP system before installing:

- Bind URL: _____
- Group DN: _____
- Domain Name: _____

  (The fully-qualified domain name of the domain controller)
- Bind User Name: _____

  (used by the Core Server for group lookup operations)
- Bind User Password: _____

The Core Server has two built-in groups, Administrators and All Users. These groups are populated from LDAP groups. You can use existing groups but it is not recommended, as anyone that is a member of the group that is mapped to the All Users group will be able to log in. Two LDAP groups should be created to list Application Configuration Console users.

- Group name to map to All Users group: _____
- Group name to map to Administrators group: _____

The Core Server must be able to connect to the LDAP system. If there is a firewall between the two systems, be sure that the LDAP port (389 is the default) is open.

### 4.2.3 Mail Server

Application Configuration Console can send e-mail messages to alert users of configuration changes. You must have the following information regarding the mail server:

- Mail Transport Protocol: _____
- SMTP Mail Host Name: _____
- SMTP Mail Port: _____
- Authentication Required? Yes No
- If yes, mail system user and password for authentication: _____

### 4.2.4 Java Version

Note the following about the Java prerequisite:

- A working version of Java such as a JDK or JRE is required to run the installation program, and its location must be included in the PATH. If Java is not installed on the host system, you must use the JDK from the installation CD when installing. See the note in Step 1 for more details.
- If a JDK is already installed on the host system, the Core Server can use that installation, provided it is the correct version. Otherwise, you must select the option to install the JDK that ships with the Core Server.

## 4.3 Install the Application Configuration Console Server

The Core Server software comes complete in a ready-to-install executable file.

> **Note:** Do not run the installation program as root on Linux systems. Log in as a non-root user to run the installation program.

1. Double-click the installation file on the CD to launch the installation program.

   - For Windows servers:

     ```
     server\ApplicationConfigurationConsoleServer.exe
     ```

   - For Linux servers:

     ```
     ./server/ApplicationConfigurationConsoleServer.bin
     ```

   - If you don't have access to the server console, or for Linux servers that do not have XWindows installed, you can use a command prompt to run the installation program in text mode:

     ```
     ./server/ApplicationConfigurationConsoleServer.bin -console -is:javahome
     <Java16_Home>
     ```

     > **Note:** The Core Server installation program requires Java, either a JRE or JDK. If the required version is not installed on the host system, use these steps:

     **Windows**: An extracted instance of JDK is included on the installation CD. Run the installation program from the command line and specify that location as JAVA_HOME. For example, from the server directory on the CD:

     ```
     D:\server>ApplicationConfigurationConsoleServer.exe -is:javahome ..\jdk\jdk1.6
     ```

     **Linux**: The `jdk` directory on the installation CD includes an installable version of JDK. Install the JDK and then install the Core Server.

     Click **Next** on the Welcome panel to proceed with the installation.

2. Accept **New Installation** as the installation type and click **Next**.

3. Use the default installation location, or click **Browse** to navigate to a different location.

   Click **Next** on the Directory panel to proceed with the installation.

4. Ensure that you meet the stated third-party requirements. If necessary, download and extract the components to a temporary directory, before continuing.

   Click **Next** to proceed with the installation.

5. Click **Browse** to navigate to the Hibernate root directory (`hibernate-distribution-3.3.1.GA`). This should be under the temporary directory where you downloaded the software; for example, `/tmp/oaccDownloads`.

   Click **Next** to proceed with the installation.

6. Click Browse to navigate to the JCIFS .jar file (`jcifs-1.2.25.jar`). This should be under the temporary directory where you downloaded the software; for example, `/tmp/oaccDownloads`.

   Click **Next** to proceed with the installation.

7. The Core Server runs on an embedded Tomcat server. Enter the HTTP and SSL port numbers that this server should use. They must not conflict with any other port assignments on the host system. Write down these port numbers, as you will need them when you install the Application Configuration Console Clients.

   Click **Next** to proceed with the installation.

8. If you're installing on a Windows system, select whether to start the Tomcat server as a service.

   Click **Next** to proceed with the installation.

9. The Core Server requires a JDK. If you want to continue to use your existing version, select the second option on this panel and then specify the JDK installation directory on the follow-on panel. If you want to use a dedicated version with Application Configuration Console, select the first option to install it with the Core Server.

   Click **Next** to proceed with the installation.

10. Type the name of the server that hosts your database and the SID that will be used for the Application Configuration Console data.

    The Core Server requires a minimum Java heap size of 1024. More memory will improve performance when working with large data sets. This must be real memory on the host machine; not just virtual memory without real memory backup.

    Click **Next** to proceed with the installation.

11. The Core Server must access the database as a specific database user. Type the user name and password that were used with the database setup scripts. Type the port on the host system that the Core Server should use for database service. Also type the port on the host system to use for the Subversion (SVN) server.

    Click **Next** to proceed with the installation.

12. Specify the time zone where the database server exists. Select a value appropriate to the geographic region from the drop-down menu.

    **Note:** There are almost 400 time zones listed. If you are installing from the console, you may need to increase the screen buffer size to be able to scroll all of the values. Open the console window properties dialog, go to the **Layout** tab, and set the buffer height to 500.

    Click **Next** to proceed with the installation.

13. The Core Server uses an external authentication system for user authentication and authorization, either LDAP or Windows NT. Select the authentication method to use. Choose AD-JNDI if you are using Microsoft Active Directory, or MV-JNDI for all other LDAP servers.

    **Note:** If you choose Windows NT, you must log in to the Core Server in console mode. The Core Server cannot be run as a Windows service.

    Click **Next** to proceed with the installation.

14. Enter the configuration information required to interact with your authentication system.

    Type the Active Directory or LDAP group names that you want to map to the two built-in user groups in Application Configuration Console, All Users and Administrators. Membership in these groups is administered in the authentication system, not in Application Configuration Console. (You may want to include the members of the Administrators group in the All Users group, or the All Users group will not actually contain all users.)

    Indicate whether names are to be case-sensitive and whether to restrict access to members of the named groups.

    Click **Next** to proceed with the installation.

15. The follow-on panel depends on the directory service you selected.

    For **AD-JNDI** authentication:

    – Type the fully-qualified Bind URL that will allow the Core Server to lookup users.

    – Type the distinguished name to use for group lookups.

    – Use the fully-qualified domain name of the domain controller.

    – Type the username and password of the user that will be used to bind to Active Directory to do the group lookup.

    For **MV-JNDI** authentication:

    – Specify the URLs under which users and groups are stored in the LDAP server.

    – Type the distinguished name and password of the administrative user with bind and read access to the directory trees under the user and group URLs. Whether the password is encrypted or not is specified by the `bind.user.password.encrypted` parameter in `server_modules_registry.xml`.

    – Type the User Account Name Attribute that represents the user's account name.

    – Type the Group Member Attribute that represents the members of the group.

    – Type the Group Name Attribute that represents the name of the group.

    – Type the User Search Spec, which is the object class of the user object. This is used by the LDAP query for users.

    – Type the Group Search Spec, which is the object class of the group object. This is used by the LDAP query for groups.

Click **Next** to proceed with the installation.

**Sample values for Sun ONE Directory Server**

User Provider URL: ldap://localhost:389/ou=people,dc=oacc,dc=local
Group Provider URL: ldap://localhost:389/ou=groups,dc=oacc,dc=local
Bind User: uid=amAdmin,ou=people,dc=oacc,dc=local
Bind User Password: password123
User Account Name Attribute: uid
Group Member Attribute: uniquemember
Group Name Attribute: cn
User Search Spec: (objectClass=person)
Group Search Spec: (objectClass=groupOfuniquenames)

**Sample values for IBM Directory Server** [1]

User Provider URL: ldap://global.NA.acmecorp.com:389/ou=region,o=ac.com
Group Provider URL: ldap://global.NA.acmecorp.com:389/ou=region,o=ac.com
Bind User: uid=oaccadmin,ou=users,ou=region,o=ac.com
Bind User Password: oaccadmin
User Account Name Attribute: uid
Group Member Attribute: member
Group Name Attribute: cn
User Search Spec: (objectClass=person)
Group Search Spec: (objectClass=groupOfNames)

**Sample values for eTrust Directory Server**

User Provider URL:
ldap://10.1.11.31:19389/ou=oaccUsers,ou=oacc,o=democorp,c=AU
Group Provider URL:
ldap://10.1.11.31:19389/ou=oaccGroups,ou=oacc,o=democorp,c=AU
Bind User: cn=leroy,ou=oaccUsers,ou=oacc,o=democorp,c=AU
Bind User Password: password123
User Account Name Attribute: cn
Group Member Attribute: member
Group Name Attribute: cn
User Search Spec: (objectClass=person)
Group Search Spec: (objectClass=groupOfuniquenames)

16. Application Configuration Console can send e-mail messages to alert users of configuration changes. On this panel enter the information required to reach the mail server.
    - You can use a host name or an IP address for the Mail Host.
    - The "Mail from address" will be used as the sending address for the messages. Use a standard-format e-mail address, such as OACC-Tracking@yourcompany.com.
    - If your mail system requires user authentication, set the last option to Yes and then enter a user name and password on the follow-on panel.

[1] These URLs assume an LDAP where the root suffix is o=ac.com, below which is a branch (ou=region), below which there are two subbranches (ou=users and ou=groups). Given this hierarchy, the respective provider URL is set to the parent branch of users and groups, not to the users and groups branches themselves. Additionally, you would add the Application Configuration Console All Users and Administrators groups to the branch ou=groups, ou=region, o=ac.com, and individual users (for example, oaccuser1, oaccuser2, oaccuser3, oaccadmin) to the branch ou=users, ou=region, o=ac.com.

Click **Next** to proceed with the installation.

17. Choose whether to create entries on the Windows All Programs menu to start and stop the Core Server.

    Click **Next** to proceed with the installation.

18. Confirm all of your settings and click **Next** to install the software, or click **Back** to alter settings.

19. When the installation completes, click **Finish** to exit the wizard.

## 4.4 Set Credentials for Windows Service

If you installed the Core Server on a Windows system and you chose to run Tomcat as a service, the installation sets it to run as the Local System user. You should change the logon to an admin account.

## 4.5 SVN Server Startup

Application Configuration Console uses a standalone Subversion (SVN) server for versioning control. The SVN server instance is set up automatically as part of Core Server installation. The SVN server must be running when the Core Server starts up. In fact, the startup process will fail if it detects that the SVN server is not running.

On Windows, the SVN server is set up as a service configured to start automatically. So if the host crashes, the SVN server always comes back on restart. For the same reason, it is desirable to configure an init script on your Linux or UNIX host, which will automatically initialize the SVN server upon restart.

1. To start the SVN server automatically when the host starts, include the following commands in the `init` script:

   ```
   $OACC_INSTALL/svn/bin/svnserve -d -root
   $OACC_INSTALL/svn
   ```

2. To stop the SVN server automatically when the host shuts down, include either of the following commands in the `init` script:

   ```
   pidof svnserve | xargs kill
   ```

   Or

   ```
   killall svnserve
   ```

If you are not authorized to create an init script for the SVN server process, provide this information to your system administrator.

## 4.6 UNC Support

UNC, or Uniform (Universal) Naming Convention, is a common syntax for describing the location of a network resource, such as a shared file, directory, or printer. If you want to use UNC in your environment, you must takes the steps below to acquire and install third-party software, and configure Application Configuration Console to use it.

To get the required JCIFS software:

1. Point your browser at the following URL:

   ```
   http://jcifs.samba.org/src/
   ```

2. Download `jcifs-1.3.12.jar` to a temporary directory. If this version is not available, download a later version. If the later version proves problematic, contact Support.

   Copy the `.jar` file to the following three locations in the Core Server installation directory:

   ```
   $OACC_INSTALL/appserver/tomcat/webapps/mvserver/WEB-INF/lib
   $OACC_INSTALL/appserver/tomcat/webapps/mvtrack/WEB-INF/lib
   $OACC_INSTALL/appserver/tomcat/webapps/mvwebreports/WEB-INF/lib
   ```

To configure Application Configuration Console for UNC support:

1. Navigate to the following location in the installation directory:

   ```
   $OACC_INSTALL/appserver/tomcat/shared/classes
   ```

2. Open `authpack_specification.xml` in a text or XML editor and add `UNC` to the first `Type authpack` element so that it appears as follows:

   ```
   <Type authpack="Username Password" worksWithEnpoint="FTP, SSH, JDBC, UNC" />
   ```

3. Save the file.
4. In the same directory, open `endpoint_specification.xml` in a text or XML editor and uncomment the `<EndpointSpec type="UNC">...</EndpointSpec>` element.
5. Save the file.

A UNC option will now be available in the Application Configuration Console Client user interface.

## 4.7 Post-Installation Tuning

Refer to the subsections that follow for information and suggestions on tuning Application Configuration Console Server.

### 4.7.1 DBCP Connection Pooling

The Application Configuration Console Tomcat server is configured to use DBCP connection pooling provided by Tomcat. The default settings define a maximum connection pool of 10 with a wait time of 3 seconds. Based on the load and throughput requirements, these default settings can be updated as needed. Please refer to Tomcat documentation on DBCP for further details at:

```
http://jakarta.apache.org/tomcat/tomcat-5.0-doc/jndi-datasource-examples-howto.html#Database%20Connection%20Pool%20(DBCP)%20Configurations
```

### 4.7.2 Server Log File Location

Server logs are created at the following locations, for Tomcat and Application Configuration Console, respectively:

```
$OACC_INSTALL/appserver/tomcat/logs
$OACC_INSTALL/appserver/tomcat/logs/mv
```

### 4.7.3 Application Configuration Console Web Reports URL

Web Reports can be accessed with a Web browser at:

```
https://mVserverHost:9943/mvwebreports/index.jsp
```

Anyone who is a member of the All Users or Administrators groups can view the reports.

### 4.7.4 Establish Application Configuration Console Server Dependency on Database

For Windows installations in which the Core Server and the Oracle database are installed on the same server, you can create a dependency so that the Core Server will not start unless the Oracle database listener is running. To do this, open a command window and type the following command:

```
sc config mValentTomcat depend= OracleServiceOACCSERV/OracleOACCTNSListener
```

Where `OracleServiceOACCSERV` is the oracle service and `OracleOACCTNSListener` is the name of the TNS Listener. Note the space after depend= and the forward slash used to separate the services specified as dependencies.

### 4.7.5 Optimize Database Performance

After you have loaded configuration data into Application Configuration Console, you should run the command below to gather usage statistics in the database, which can be used by the database to optimize access plans.

1. Connect to the database as the oaccuser and run this command:

   ```
   exec dbms_stats.gather_schema_stats(ownname => 'OACCUSER', options => 'GATHER
   AUTO');
   ```

2. Stop and restart the Core Server.

You can rerun this command periodically to keep the database tuned.

## 4.8 Starting the Application Configuration Console Server

Refer to the section that applies to your platform for information on starting and stopping the Core server.

> **Note:** As the person who installs the Core Server, you must grant (operating system) permissions to anyone else who will need to start the Server on this machine.

### 4.8.1 Windows

If you chose to configure Tomcat as a service, restart the server to start Tomcat and the Core Server. If you did not configure Tomcat as a service, you can start and stop the Core Server with the batch files located under the server installation directory. If you used the default installation directory, they would be:

```
$OACC_INSTALL\appserver\tomcat\bin\startup.bat
$OACC_INSTALL\appserver\tomcat\bin\shutdown.bat
```

### 4.8.2 Linux

You can start and stop the Core Server with the script files located under the server installation directory.

```
$OACC_INSTALL/appserver/tomcat/bin/startup.sh
$OACC_INSTALL/appserver/tomcat/bin/shutdown.sh
```

The installation also includes an init script that you can use to start and stop the Tomcat server.

1. Change directory to the following location:

   ```
   cd $OACC_INSTALL/appserver/tomcat/conf/mv
   ```

2. Copy the init file to the /etc/init.d directory:

   ```
   cp tomcat_init /etc/init.d/tomcat &&
   ```

3. Set permissions on the new script:

   ```
   chmod 755 /etc/init.d/tomcat
   ```

4. Add the new script to the runlevel information:

   ```
   chkconfig --add tomcat
   ```

The first time you start the Core Server after installation on a Linux server, it may produce some errors. If you see errors, stop the Server and then restart it.

## 4.9 Web Reports and Materialized Views

Application Configuration Console Web Reports includes a System Summary report that can potentially impact performance, owing to the extent of information gathering it undertakes. To cut down on the impact, the Core Server installer creates an Oracle materialized view named `MV_SYSTEM_SUMMARY` to populate the System Summary report more efficiently. The installer specifies a refresh cycle of midnight of every day to update the information used to populate the report.

Given the way a materialized view works, the query results can become stale between refreshes. The report you run today, for example, does not include any activity that occurred since midnight last night. This seems an acceptable trade-off when you consider the nature of a summary report. If, however, you want a report that includes the latest information, you can simply perform a manual refresh prior to running the report.

To perform a manual refresh, execute the following Oracle command while logged into a SQL*Plus session as the Application Configuration Console database user:

```
exec dbms_refresh.refresh(name => 'MV_SYSTEM_SUMMARY');
```

# 5

# Tarball Install on a Linux Server

If you do not have direct access to the server that will host the Core Server and database, or if the server does not have a CD drive, use the tarball installation CDs and copy the files over the network. These instructions cover installing both the database and the Core Server.

## 5.1 Prerequisites

A tarball installation has the following prerequisites:

- The server must have 1 GB of free disk space for the installation files.
- The installation program will compile some needed files. The Linux system must have the C/C++ Compiler and Tools installed (specifically, `libAio_devel` and `glibc_devel`), and the Oracle user (created below) must have execute permission on `/usr/bin/gcc`.

## 5.2 Create a User and Group

To install the Application Configuration Console database on a Linux Server, you must create a new user and group, and then open two sessions to run some scripts, one as the new user and one as root.

1. Log in to the Linux server as a user who has permissions to create groups and users.
2. Create a dba group. If you are using the command line:

   ```
   groupadd dba
   ```

3. Create an Oracle user as a member of the dba group and specify the directory that you want to use for `ORACLE_HOME` as the user's home directory, for example:

   ```
   useradd -g dba -d /opt/mvOracle oracle
   ```

## 5.3 Copy the Installation Files to the Server

1. Create an installation directory, such as `/tmp/mvinstall`.
2. Copy the four tarballs from the two CDs to the install directory.
3. Extract the tar files with `tar -xzvf`. You should end up with two directories: mvOracleInstall and mvalent3.

## 5.4 Install the Database

1. Open a new shell and become the Oracle user.
2. cd to `/tmp/mvinstall/mvOracleInstall`.
3. Run `./install.sh`, running the root scripts from the root shell. You can safely ignore any warning messages about setting permissions on `/home/oracle/network/agent/html`.
4. When the installation completes, restart the server and then as root test `/etc/init.d/oracle stop/start`.

## 5.5 Set Up the Database

1. `chmod` the `/tmp/mvinstall/mvalent3` directory to `777`.
2. Login again as Oracle (to load the correct profile and variables).
3. cd to `/tmp/mvinstall/mvalent3`.
4. Unzip the `db_setup.zip` file.
5. cd to the unzipped `db/scripts` directory.
6. Set up the database with this command:

   ```
   $ sqlplus /nolog @setup_db.sql
   ```

7. Exit out of the Oracle shell when the database setup is complete.

## 5.6 Install the Server

1. Open a shell as the Application Configuration Console user.
2. cd to `/tmp/mvinstall/mvalent3`.
3. Run the Core Server installation program using the following command (see Section 4.3, "Install the Application Configuration Console Server," for details on the installation):

   ```
   $ server/ApplicationConfigurationConsoleServer.bin
   ```

4. After the Server installation is complete, `login/su` as root and copy the `tomcat_init` script from `/tmp/oaccinstall/mvalent3/extras/tomcat_init` to `/etc/init.d/tomcat`, then `chmod 755` the script.
5. Edit `/etc/init.d/tomcat`, setting `tomcat_home` and `tomcat_user` correctly.
6. As root, do a `chkconfig --add tomcat`.

# 6

# Install the Application Configuration Console Client

Follow the instructions in this chapter to install Application Configuration Console Clients on end-user machines.

## 6.1 System Requirements

Ensure that the machines on which you are installing Application Configuration Console Clients comply with the hardware, software, and browser requirements stated in Table 2–4 in Chapter 2.

### 6.1.1 Client Prerequisites

Note these prerequisites:

- The Core Server must be installed before installing Client software. You will need the server host name and port numbers during the Client installation.
- Each Client system must have a working version of Java (JDK or JRE) to run the installation program, and its location must be included in the PATH.
- If using SSH certificates, you must use Open SSH to generate certificates.

## 6.2 Installing Application Configuration Console Clients

The Client software comes complete in a ready-to-install executable file.

1. Insert the Application Configuration Console Installation CD and double-click client/ApplicationConfigurationConsoleClient.exe to start the automated installation program.

   Click **Next** on the Welcome panel to proceed with the installation.

2. Use the default installation location, or click `Browse` to navigate to a different location. You should not install a new version of the Client in the same directory as a previous version.

   Click **Next** to proceed with the installation.

3. The Client requires a JDK. If you want to continue to use your existing version, select the second option on this panel and then specify the JDK installation directory on the follow-on panel. If you want to use a dedicated version with Application Configuration Console, select the first option to install it with the Client.

   Click **Next** to proceed with the installation.

4. The Client needs to connect to the Application Configuration Console's Core Server. Type the host name and server ports where the Core Server was installed.

   Set the memory (Java heap size) allocated for the Application Configuration Console Client. Do not decrease the Initial Java heap size; the Client requires at least 256 MB of memory. You can leave this default value, or set a larger number if you expect to work with large data sets.

   Click **Next** to proceed with the installation.

5. Select whether the installation is available only to the user who is logged in, or to all users. Also select whether to include an icon for the Uninstaller.

   Click **Next** to proceed with the installation.

6. Confirm all of your settings and click **Next** to install the software, or click **Back** to alter settings.

7. When the installation completes, click **Finish** to exit the wizard.

## 6.3 Starting the Application Configuration Console Client

To start the Client after the installation, simply double-click the desktop icon (if you installed one), or go to the Start menu and select **All Programs > Oracle Application Configuration Console > Client>Start**.

When you first log in to the Client, it creates an application log file (`mvApp.log`) and properties file (`mvApp.properties`), where Client preferences such as the Core Server URL are stored. The default location for these files is as follows:

```
%USERPROFILE%\Application Data\ApplicationConfigurationConsole
```

Where `%USERPROFILE%` defaults to `C:\Documents and Settings\`*`username`*.

The Client creates other properties files as users set specific preferences related to tracking (`mvAppTrack.properties`) and comparison (`mvAppComparison.properties`) operations. Together with the log and application properties files, these combine to make a Client profile on the host machine.

---

**Note:** As the person who installs the Client, you must grant (operating system) permissions to anyone who will need to start the Client on this machine.

---

### 6.3.1 Connecting to Multiple Server Instances with the Same Client Version

Oracle discourages use of separate Client instances of the same version on a Client host. Rather, you should use the same Client login to connect to different server instances, for example staging and development. You can facilitate this by setting a preference in the Application Configuration Console application to prompt for a URL pointer each time you start up the Client.

1. In the Client, select **Window > Preferences**.
2. In the **General** tab, select the **Prompt for server URLs at each login** checkbox.
3. Click **OK**.

Now, each time you start up the Client, you will be prompted to type or select the URL of the Core Server you want to connect to.

### 6.3.2 Using Different Client Versions on the Same Host

User sites often want to run different Application Configuration Console versions in parallel. When you want to run different Client versions on the same host, create discrete Client environments, as follows:

- Install to separate directories
- Create distinct properties profiles and logs

So, for example, if you want to run Application Configuration Console 5.3.1 and 5.3.x on the same Client host, you might install to the separate directories:

```
C:\Program Files\Oracle\oacc\Client-5.3.1
C:\Program Files\Oracle\oacc\Client-5.3.x
```

Then, for each installation, create a unique profile.

1. Navigate to the runtime folder under the 5.3.1 installation directory. For example:

   ```
   C:\Program Files\Oracle\oacc\Client-5.3.1\runtime
   ```

2. Open the `startup.bat` file in a text editor and edit the following line to point to a unique location:

   ```
   -Dmv.userfile.dir="%USERPROFILE%/Application Data/
   ApplicationConfigurationConsole"
   ```

   Where `%USERPROFILE%` defaults to `C:\Documents and Settings\`*`username`*. For example, change the last folder to ApplicationConfigurationConsole-5.3.1. A folder of that name does not have to exist; Application Configuration Console will create it upon startup.

3. Save the file.
4. Repeat the first three steps for the 5.3.x installation, changing the folder name appropriately.

Now, when you start up the 5..3.1 Client, the preferences reflect the appropriate settings to connect to the 5.3.1 Core Server. Likewise, when you start up the 5.3.x Client, the preferences are appropriate to the 5.3.x Core Server.

## 6.4 Repointing Web Reports

Application Configuration Console Client installation places a shortcut on the Start menu to open Web Reports in a browser (**All Programs > Oracle Application Configuration Console > Client>Start Web Reports**). If, for some reason, you want to change the URL that this shortcut points to, take the following steps:

1. Navigate to the following location in the Client default installation directory:

   ```
   C:\Program Files\Oracle\oacc\Client\runtime
   ```

2. Right-click `webreports` and select **Properties** in the popup menu.
3. Click the **Web Document** tab and edit the URL value as appropriate.
4. Optionally type a shortcut key combination.
5. Click **OK**.

You may want to do this, for example, if you decide to redeploy the Web Reports server, as described in Section D.2.1, "Redeploying Web Reports Server."

## 6.5 Integrating Custom Reports

Web Reports can accommodate custom reports developed by Professional Services to meet the reporting needs of individual customers. Custom reports must be coded as JSPs and deployed to the Tomcat server, after which they can be integrated into Web Reports and made accessible on the menu bar. You must use BIRT design formatting if you want to schedule report generation. The default location for Web Reports JSPs is as follows:

```
$OACC_INSTALL/appserver/tomcat/webapps/mvwebreports/jsp
```

To integrate a custom report, you have to edit the Web Reports Registry (`webreports_registry.xml`) to recognize the JSP. This in turn allows users to schedule the report and select it on a menu. You can either add the report as a menu item to an existing menu or create a custom menu. The same report can appear on multiple menus, if desired.

Specific edits to the Web Reports Registry include the following:

- Add a report definition to the report-definitions element that consists of a report name, a target JSP, and a unique identifier (URI). For example:

  ```
  <report name="Your Report"
  target="/mvwebreports/jsp/YourReport.jsp"
  uri="http://mvalent.com/2003/reportDefinitionType#YourReport"/>
  ```

- If you are adding a custom report to an existing menu, add a menu item to the appropriate menu element, Planning, say. For example:

  ```
  <menu name="Planning">
  <menu-item uri="http://mvalent.com/2003/reportDefinitionType#YourReport"/>
  </menu>
  ```

  Use the same URI value as specified in the report definition.

- If you are adding a custom report to a new menu, add a new menu element that includes the menu item. For example:

  ```
  <menu name="Your Menu">
  <menu-item uri="http://mvalent.com/2003/reportDefinitionType# YourReport"/>
  </menu>
  ```

  In fact, the Web Reports Registry includes an element placeholder for a new menu that you can uncomment, replacing "Custom Report" with your menu name and "tbd" with an appropriate URI.

Add report definitions and menu items for each custom report you want to integrate.

> **Note:** Report scheduling depends on the uniqueness of a report's URI. It cannot be altered or reused for some other report. If it is altered in any way (or deleted), scheduling for the report will no longer work. For this reason, be careful as well not to inadvertently change existing elements when editing the Web Reports Registry.

### 6.5.1 Edit the Web Reports Registry

To edit the Web Reports Registry:

1. Open the Application Configuration Console Client.
2. In the Navigator view, locate the configuration file, as follows:

   **System > System Configuration > Common Configuration > MV_CONFIG > Resource View**
3. Right-click `webreports_registry.xml` and select **Open**.
4. In the Editor area, click the **Edit** button and add the necessary elements.
5. Save the file.
6. In the Navigator view, right-click the **System Configuration** folder and select **Make Configuration Changes Live** to render the new report selection in Web Reports.

# 7

# Install Automation Modules

To install automation modules, you must start the Application Configuration Console Server, then start the Client and log in as a member of the Administrators group.

## 7.1 Prerequisites

The WebSphere and WebLogic automation modules require additional software to be installed on the Core Server host system, usually before you install the automation module:

- For the *WebSphere Automation Module*, the WebSphere Deployment Manager must be installed on the same machine as the Core Server. It does not need to be running, but the software must be installed so that the automation module commands can run WSAdmin actions to make changes to WebSphere configurations. Make note of the Deployment Manager installation directory, because you will need to enter it for some commands.
- For the *WebLogic Automation Module*, the WebLogic Server software must be installed on the same machine as the Core Server. It does not need to be running, but the software must be installed so that the automation module commands can run WLST actions to make changes to WebLogic configurations. Make note of the WebLogic Server installation directory, because you will need to enter it for some commands.

## 7.2 Installation

To install an automation module, proceed as follows:

1. Copy the `.jar` file for the automation module to the Core Server host system.
2. In the Client, select **Admin > Install Extension** in the menu bar.

   The Install Extension dialog opens.
3. Select **automation** as the extension type.
4. Click **Browse** to locate the .jar file in the file system.
5. Click **OK** to install the automation module.

Some automation modules prompt for additional information during installation.

The automation module features are available immediately after installation. You do not need to restart the Application Configuration Console Server or Clients.

**Note:** If you install an automation module after redeploying a secondary server, you have to port the AM installation to the secondary server. See Section D.5, "Redeployment and Automation Modules," for details.

## 7.3 Configuring WebSphere for SSL Authentication

This section describes a process for securing communication between `wsadmin` as run by the Application Configuration Console Client and the WebSphere Deployment Manager. The mechanism you will put in place enforces authentication between these components using SSL certificates. WebSphere ships with a repertoire of SSL key files that are preconfigured to support this authentication. These dummy key files, located in the `\etc` directory of your WAS installation, are as follows:

```
DummyServerKeyFile.jks
DummyServerTrustFile.jks
DummyClientKeyFile.jks
DummyClientTrustFile.jks
```

If you choose to create your own keystores, remember that the client and server trust files must each contain both the client and server keys. Go to the following URL for more information:

`http://www.redbooks.ibm.com/redbooks/SG246573/wwhelp/wwhimpl/java/html/wwhelp.htm`

### 7.3.1 Enable Global Security

All security checks, including SSL authentication, are disabled until you enable global security. So this is the first step to implementing an SSL solution.

1. Open the administrative console.
2. Select **Security > User Registries > Local OS**.

**Note:** In a production environment, you would typically select LDAP or Custom in Step 2 to implement deeper role-based security.

3. In the Configuration tab, enter the **Server User ID** and **Server User Password** in the text boxes provided. These are valid credentials in the local OS where the Deployment Manager executes. On Windows, use Administrator or a local user with administrative privileges. On Linux, use the same user as the Deployment Manager (for example, `root/ wasuser`).
4. Click **Apply** to save these settings
5. Select **Security > Authentication Mechanism > LPTA**.
6. In the **Configuration** tab, create a new password and confirm it. This password is used to generate the LPTA keys. This is a requirement to enable global security. LPTA keys are used in trust association for reverse proxies and SSO configurations.
7. Click **Apply** to save these settings.
8. Select **Security > Global Security**.
9. In the **Configuration** tab, select the **Enabled** check box. Verify that the other settings are appropriate.

10. Click **OK** to save the configuration.

If you federated any nodes, you may want to synchronize these changes on the federated nodes accordingly. You will also need to restart the Deployment Manager.

### 7.3.2 Configure the Deployment Manager

After restarting the Deployment Manager, log in using the user name and password specified in Step 3 under Section 7.3.1, "Enable Global Security." Open the administrative console. Notice that you must now use the https protocol. If you use the old (http) URL, global security redirects you by forcing the server to use the `DefaultSSLSettings` for the Deployment Manager's http transport, as specified in the SSL Configuration Repository.

Now connect to the server from the wsadmin client using the user name and password specified in Step 3, as follows:

```
wsadmin -username serveruser -password serveruserpassword
```

Ensure that you can connect to the Deployment Manager using this syntax, before proceeding.

1. In the administrative console, select **Security > Authentication Protocol > CSIv2 Inbound Authentication**.
2. Set **Basic Authentication** to **Never**.
3. Set **Client Certificate Authentication** to **Required**.
4. Click **OK** to save these settings.

This makes the server force clients to authenticate using the SSL certificates specified in the SSL repertoire (`DefaultSSLSettings`) and to disallow basic authentication (user name and password).

### 7.3.3 Configure the Client

To complete configuration of the client, modify the `soap.client.props` file so that you will not have to pass the user name and password to wsadmin on the command line. The file is located in the `properties` folder of your WebSphere installation, for example:

```
C:\WebSphere\DeploymentManager\properties\soap.client.props
```

Add the user name and password specified in Step 3, underSection 7.3.1, "Enable Global Security," to the following lines:

```
com.ibm.SOAP.securityEnabled=true

#JMX SOAP connector identity
com.ibm.SOAP.loginUserid=serveruser
com.ibm.SOAP.loginPassword=serveruserpassword
```

If you want to encrypt the password, use the `PropFilePasswordEncoder` utility. See the instructions at the top of the `soap.client.props` file.

Note that if you have federated any nodes in the Deployment Manager, you may need to restart the node manager. If you synchronized the changes on your federated nodes, you will also need to make the same changes as above, to the the `soap.client.props` file in your application server installation.

## 7.4 Preserving Configuration Changes

If you reinstall an automation module, the installer checks for changes to certain configuration files that you are allowed to edit. If the installer detects differences, it displays a dialog warning that differences exist between the version that was there and the version just installed. Users often customize the save specification registry, for example, to aid in formulating meaningful comparisons. If you made changes to the `wl9_save_spec_registry.xml` file, the installer notifies you with a message to that effect.

You can compare the installed version to the version that you had edited to decide if you want to retain your changes.

1. In the Navigator view, locate the saveSpecRegistry file (`wl9_save_spec_registry.xml`):

   **System Configuration > Automation Modules > automation#weblogic9AM > Resource View**

2. Open the file in the Editor area and click the **Versions** tab.
3. Select the last two versions (post- and pre-installation).
4. Right-click and select **Compare Properties** to see the differences.
5. Preserve any changes you want to retain by merging your edits into the new file.

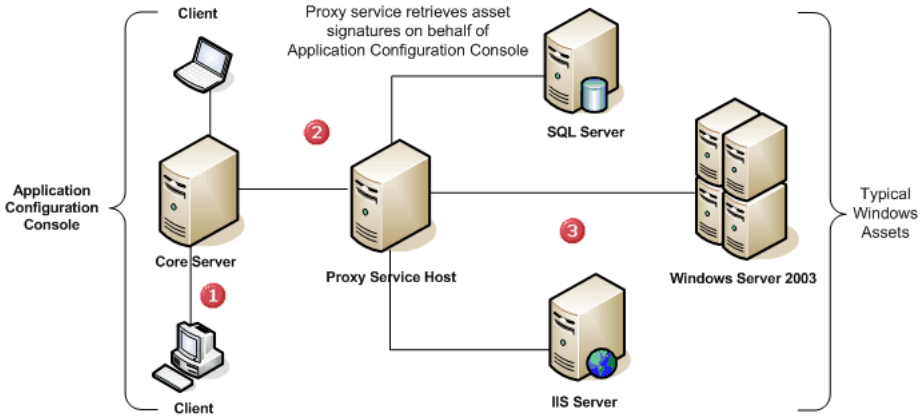
During a reinstallation, differences if any are typically detected in the `saveSpecRegistry` file, and less frequently, in the `viewSpecRegistry` file.

# 8

# Install the Windows Resource Extensions

The Windows Resource Extensions (WRE) is an Application Configuration Console product add-in that enables you to extract Windows configuration data from Windows servers to create assets in Application Configuration Console. To do this, the Application Configuration Console Server connects to a proxy service, which in turn, runs Visual Basic scripts on target machines to extract the data and pass it back through the Core Server to Clients. Figure 8–1 depicts the flow of information in a typical configuration.

***Figure 8–1 Flow of Information in Windows Resource Extensions***




1. From a Client, a user requests an asset load of a Windows asset.
2. The Core Server connects to the proxy service host, passing the relevant request information.
3. The proxy service executes the appropriate script on the target machine to extract the data and return it to the Client through the Core Server.

If you have purchased this product, your Application Configuration Console software distribution disc includes a Windows Resource Extensions folder with the following contents:

- `windowsre_wmi_scripts.zip`–this zip file to extract the WRE scripts to the machine hosting the Application Configuration Console Proxy Service.
- `WindowsRE.jar`–Use this JAR file to install the WRE automation module on Application Configuration Console.

# 8.1 Windows Resource Extensions Prerequisites

Before you proceed with the installation, ensure that the respective environments satisfy the stated requirements.

## 8.1.1 Proxy Service Requirements

General requirements include the following:

- OpenSSH must be installed on the proxy host. See Section 8.3.1, "Download and Install OpenSSH."
- The host machine operating system must be Windows 2000, Windows XP, or Windows Server 2003.
- The credentials used to access the proxy service host and the target machines must be part of the Local Administrators Group on the machine. These credentials must have rights to execute the scripts on the proxy service host and to perform administrative tasks on the target machines.

Additional requirements depend on the Windows asset type.

### 8.1.1.1 IIS 5.0

The IIS 5.0 scripts use the ADSI provider to access configuration data. This requires that the IIS common files be installed on the proxy service host. If not already present, you can install them by opening **Add or Remove Programs** in the Control Panel and selecting **Internet Information Services (IIS) > Common Files**. Note that some of the scripts require that the proxy service be able to communicate with the target machines using UNC path specifications over TCP port 135 (RPC).

### 8.1.1.2 SQL Server 2000

The SQL Server 2000 scripts use SQL-DMO to access configuration data. This requires that SQL Client Tools be installed on the proxy service host, specifically, the following selections:

- "Client Connectivity
- "Development Tools
- "Header and Libraries

As SQL-DMO uses the Microsoft SQL Server ODBC driver to connect to and communicate with SQL Server instances, the proxy service must be able to communicate with the target machines using ODBC.

### 8.1.1.3 IIS 6.0 and Windows OS

The scripts for these assets use WMI providers to access configuration data, so the WMI service must be running (it starts automatically by default). The proxy service must be able to communicate with the target machines over TCP port 135 (RPC).

## 8.1.2 Automation Module Requirements

To install the automation module, you must have already completed the Application Configuration Console Server and Client installations.

## 8.2 What If There's a Firewall?

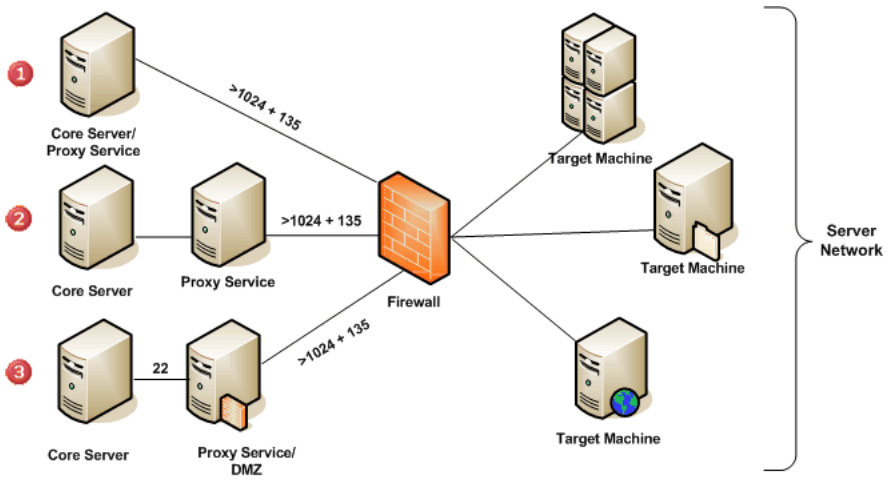
If there's a firewall in play, where you install the Application Configuration Console Proxy Service is a matter of resources and network conventions established at your site. Consider the following possibilities:

- You can install the proxy service outside the firewall, as follows:
  - On the same host as the Core Server (option 1)
  - On a separate host from the Core Server (option 2)
  - "On a separate host from the Core Server and deployed in its own DMZ, the so-called demilitarized or demarcation zone (option 3)
- You can install the proxy service inside the firewall, as follows:
  - On a host where a network server also resides (option 1)
  - On a host separate from the servers in your network (option 2)

### 8.2.1 When the Proxy Service Is Outside the Firewall

Figure 8–2 illustrates the three possibilities identified when the proxy service is outside the firewall.

***Figure 8–2 Installing the Proxy Service Outside the Firewall***



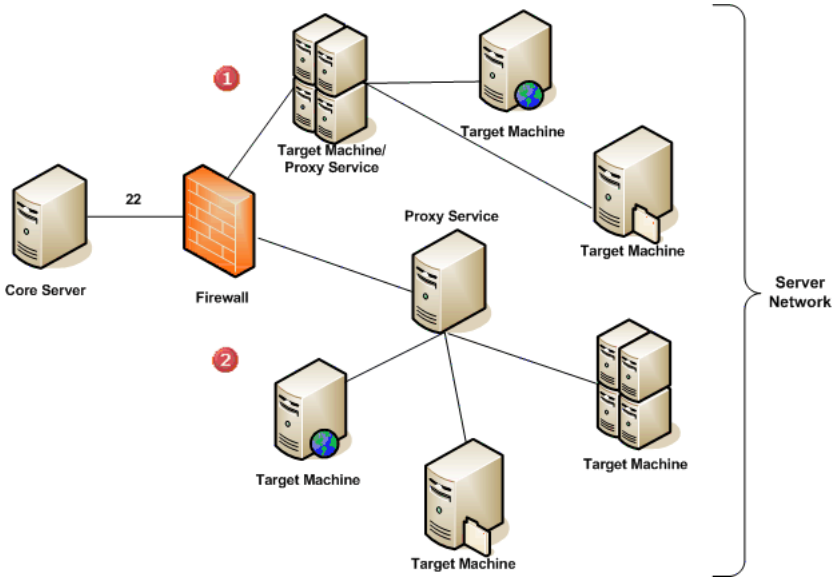

The proxy service uses WMI, SMB, and third-party COM APIs to extract information from target machines. These Windows technologies must therefore be accessible on the target machines. These technologies in turn use DCOM, which allocates random ports to communicate across the network.

Since the proxy service is outside the firewall, the firewall rules must allow communication between target machines and the machine hosting the proxy service on any TCP port over 1024 and on TCP port 135. The rules will need to allow the Windows file sharing ports as well. When the proxy service is in its own DMZ, the rules must be extended to allow communication between the Core Server and proxy service hosts using SSH over TCP port 22. Implicit in the first two options is that the Core Server uses SSH to communicate with the proxy service over TCP port 22.

### 8.2.2 When the Proxy Service Is Inside the Firewall

Figure 8–3 illustrates the two possibilities identified when the proxy service is inside the firewall.

***Figure 8–3 Installing the Proxy Service Inside the Firewall***




The proxy service uses WMI, SMB, and third-party COM APIs to extract information from target machines. These Windows technologies must therefore be accessible on the target machines. Since the proxy service is inside the firewall, the firewall rules must allow communication between the Core Server and proxy service hosts using SSH over TCP port 22.

## 8.3 Application Configuration Console Proxy Service Setup

The Application Configuration Console Proxy Service consists of the following components:

- A version of OpenSSH to enable secure communication with the target machines
- A set of Visual Basic scripts designed to extract configuration and other data from Windows servers

---

**Note:** OpenSSH must already be installed on the machine hosting the proxy service. Additionally, there can be no Cygwin component on this machine.

---

### 8.3.1 Download and Install OpenSSH

Use the following instructions to ensure that your OpenSSH setup conforms to the proxy service requirements:

1. Paste the following URL into your browser:

   ```
   http://sshwindows.sourceforge.net/download/
   ```

2. On the page that opens, click **Binary Installer Releases**.
3. On the next page, download `setupssh381-20040709.zip` to the proxy service host.
4. Log in to the proxy service host as someone with administrative privileges.
5. Extract the zip file to a temporary directory.
6. Navigate to the directory where you extracted the zip file and run setup.exe. Proceed with the installation.
7. At Choose Components, select **Server** and **Start Menu Shortcuts**. Continue with the installation.
8. At Choose Install Location, enter the following as the Destination Folder:

   ```
   C:\Program Files\OpenSSH
   ```

   Continue with the installation.
9. A warning about editing the `passwd` file appears; this is addressed later in the chapter. Click **OK**.
10. When the installation completes, click **Finish**.

### 8.3.2 Load Windows Resource Extensions Scripts

The scripts that ship with WRE must be installed in a fixed location so the resource specifications installed with the WRE Automation Module will work out-of-the-box.

To load the WRE scripts on the proxy service host:

1. Log in as someone with administrative privileges.
2. Create the directory mValentScripts under the OpenSSH installation directory, as follows:

   ```
   C:\Program Files\OpenSSH\mValentScripts
   ```

3. Load the Application Configuration Console software distribution disc and navigate to the Windows Resource Extensions folder.
4. Extract the `windowsre_wmi_scripts.zip` file to the directory you created in Step 2.

> **Note:** If for some reason your organization does not permit use of OpenSSH and you have to use some other means of secure network connectivity, the `mValentScripts` directory containing the scripts must be at root for WRE to work out of the box. So, for example, if you log in remotely, type `cd /mValentScripts`, and successfully change to that directory, you are properly configured. If you have questions or concerns, contact Support.

### 8.3.3 Create the passwd File

You need a `passwd` file on the proxy host machine to be able to log in to and use SSH. The `passwd` file is the equivalent of the `/etc/passwd` file on UNIX-based systems. Application Configuration Console supplies a script to automatically create `passwd` and `group` files of all users and groups. The script gives you the option to create these files for either local or domain (or both) user accounts.

- Local means only users with accounts on the proxy service host will be able to log in to SSH.
- Domain means users in the domain will be able to log in to SSH, but local users will not be able to.
- Both means that domain and local users will be able to log in to SSH. In case of duplicates, the domain user takes precedence; that is, the password associated with the domain user is required to log in.

To run the script:

1. Log in as someone with administrative privileges.
2. In Windows Explorer, navigate to the following directory:

   ```
   C:\Program Files\OpenSSH\mValentScripts
   ```

3. Double-click the following file:

   ```
   mkpasswd.vbs
   ```

   The script presents the option to create domain only (-D), local only (-L), or both domain and local (-D, -L) user accounts.
4. Enter the option appropriate to your network environment and click **OK**.
5. The script announces successful creation of the files. Click **OK** to terminate the script and restart the OpenSSH service.

## 8.4 Installing the Automation Module

To install the automation module:

1. Start the Application Configuration Console Server, then start the Client and log in as a member of the Administrators group.
2. Copy the `WindowsRE.jar` file from the software distribution disc to a location in the file system.
3. In the Client, click the **Admin** menu in the menu bar.
4. Click **Install Extension** to open the dialog of the same name.
5. Select **automation** as the extension type.
6. Click **Browse** to locate the `WindowsRE.jar` file in the file system.
7. Click **OK** to install the automation module.

This creates the Windows Resource Extensions resource specifications under the **System > Resource Specifications** folder in the Navigator View. These resource specifications have counterpart scripts on the proxy service host that they call to extract configuration and other data from target machines. The scripts are located at the following location:

```
C:\Program Files\OpenSSH\mValentScripts
```

---

**Note:** If you install an automation module after redeploying a secondary server, you have to port the AM installation to the secondary server. See Section D.5, "Redeployment and Automation Modules," for details.

---

# A

# Upgrade to Version 5.3.2

Follow these steps to upgrade your Core Server installation from version 5.x.x to version 5.3.2. If you are on a version earlier than 5.x.x, you must first upgrade to a 5.x.x version before you can upgrade to 5.3.2. Note that you do not need to upgrade Application Configuration Console Clients because they do not store any data. After upgrading the Core Server, uninstall the existing Client software and install the version 5.3.2 Client.

> **Note:** All instructions, paths, and so forth, say 5.x.x. Substitute the specific 5.x.x version you are upgrading from.

Instructions in this appendix observe the following convention: `$OACC_INSTALL` refers to the Core Server installed location, for example, `opt/oracle/oacc/server`.

## A.1 Shut Down Application Configuration Console

Make a list of installed automation modules. Log in to the Client as an administrator and select **Admin > Uninstall Extension** to see a list of installed automation modules.

The Application Configuration Console Server and the Clients must be shut down to upgrade. Close any running Clients, and then stop the Server. Check the server log file to verify that it has shut down. The log is located at the following path:

```
$OACC_INSTALL/appserver/tomcat/logs/mvServer.log
```

## A.2 Backup and Preparation

Address the following prerequisites before running the upgrade program:

- Verify that there are no open windows pointing at or below the Core Server installation directory. Also ensure that there aren't any files open in editors at or below the installation directory, such as a log file open in an editor.
- Have your DBA make a backup copy of the Oracle database. Make sure that the export owner is the same user ID that was specified for Application Configuration Console installation.
- Make a backup copy of the Core Server installation directory.

- Make a backup copy of the SVN repository. You can find the location of the repository in `server_modules_registry.xml`. See the "url" property value for the following module:

  ```
  "com.mvalent.service.system.repository.version.impl.subversion.SvnSessionContex
  t"
  ```

### Password Unencrypted?

If you use an unencrypted password for the bind user (JNDI authentication), you must encrypt it before running the upgrade procedure. Use the encryption program appropriate to your platform (`MVEncryption.bat` or `MVEncryption.sh`) to encrypt the password. You can find them in the `$OACC_INSTALL/appserver/tomcat/shared/scripts` directory.

When you execute the script, it prompts you for the string to encrypt. After you run the script, edit the `server_modules_registry.xml` file to set the `bind.user.password.encrypted` value to `true`.

## A.3 Perform Upgrade Input Validation

You must run a script to perform upgrade input validation. This script must be run on the Oracle server; it cannot be run remotely.

Before proceeding, ensure that your environment complies with the following requirements:

- The `$ORACLE_HOME/bin` directory must be in the user's PATH in order for the validation script to perform all testing successfully.
- (**Linux/UNIX**) Grant the Oracle user read write access to the `scripts` directory of the `unzipped oracle_db_setup.zip` file, for example, `chmod 777 /tmp/db/oracle/scripts`. Grant executable permissions to the `*.sh` scripts, for example `chmod +x /tmp/db/oracle/scripts/*.sh`. Sudo to the Oracle user: `sudo - oracle`.

1. From the command line on the Oracle server, change directory to the unzipped `db/oracle/scripts` directory, for example:

   ```
   cd /tmp/OaccInstall/db/oracle/scripts
   ```

2. Run the script appropriate to your platform:

   ```
   validate_oracle_w32.bat (Windows)
   validate_oracle_lin.sh (Linux)
   validate_oracle_sol.sh (UNIX)
   ```

   To accept a default value for those prompts that display a default, press Enter.

3. Type u to perform preupgrade validation (i is for preinstallation validation, which is described in Chapter 3).

   ```
   Validation test scope? (i/u): u
   ```

4. Type the name of the Oracle Service that you want to use with Application Configuration Console. If you are using the Application Configuration Console-supplied database, take the default.

   ```
   Oracle Service (default=OACCUSER): servicename
   ```

5. Type the Oracle System username and password at the prompts.

```
System Username (default=SYSTEM)            : username
System Password (no default)                : password
```

6. Type an Oracle username and password for the Application Configuration Console user. This username must be entered in all uppercase letters.

```
Application Configuration Console Username (default=OACCUSER): oaccusername
Application Configuration Console Password (no default)      : oaccpassword
```

7. Type the full path to the lib directory under the unzipped directory. The path must not contain lib, spaces, or a trailing (back)slash.

```
Path to the "lib" directory (no default): unzip_dir/v5xx_v532
```

8. In the directory from which you ran the script (the current directory), open the `validate_oracle_settings.out` log file and look for the following entry at the end of the file:

```
SUCCESS.  You are ready to run the upgrade_db.sql script.
```

   If the log reports errors, do not continue with database upgrade. Notify your system administrator and contact Oracle Support.

## A.4 Upgrade the Database

You must run a script to upgrade the schema and stored procedures in the Oracle database used by Application Configuration Console. This script must be run on the Oracle server; it cannot be run remotely. As you run the script, specify values for the prompts based on what you entered during validation. You can find these values, excluding passwords, in the `validate_oracle_settings.out` log file. If the file contains multiple entries, use the values at the end, as they are the most recent. Here is a sample output from the log file:

```
...
Oracle Service                                          : OACCSERV
System username                                         : SYSTEM
System password                                         : ******
Application Configuration Console username              : OACCUSER
Application Configuration Console password              : ******
...
Validating stored procedures directory: /tmp/OaccdbUpgrade/db/oracle
...
```

Perform database upgrade as follows:

1. From the 5.3.2 Application Configuration Console installation CD, extract `oacc_upg_v5xx_v532_db_setup.zip` to a temporary directory on the Oracle server, such as `/tmp/OaccdbUpgrade`. *The path to the unzip directory cannot contain spaces*.

2. From the command line on the unzipped `/tmp/OaccdbUpgrade/v5xx_v532/scripts` directory, run the upgrade script appropriate to your platform:

```
upgrade_db.bat (Windows)
upgrade_db.sh (Linux or UNIX))
```

3. Type the full path to the lib directory under the unzipped directory. The path must not contain lib, spaces, or a trailing (back)slash.

```
Path to the "lib" directory (no default): unzip_dir/v5xx_v532
```

4. Type the name of the Oracle Service that you want to use with Application Configuration Console. If you are using the Application Configuration Console-supplied database, take the default.

   ```
   Oracle Service (default=OACCSERV): servicename
   ```

5. Type the Oracle System username and password at the prompts.

   ```
   System Username (default=SYSTEM): username
   System Password (no default)    : password
   ```

6. Type an Oracle username and password for the Application Configuration Console user. This username must be entered in all uppercase letters.

   ```
   Application Configuration Console Username (default=OACCUSER): oaccusername
   Application Configuration Console Password (no default)      : oaccpassword
   ```

   You should see Connected followed by a long sequence of SQL configuration commands and their results. You can safely ignore any error messages about tables or views that don't exist. At completion you should see the following message:

   ```
   *** Update of schema and stored procedures completed. ***
   ```

Note the existence of the setup_db.out file in the unzipped scripts directory (`/tmp/OaccdbUpgrade/v5xx_v532/scripts`). This file contains information that can be useful to Oracle technical support in troubleshooting situations.

## A.5 Upgrade the Application Configuration Console Server

Use the Upgrade option in the Core Server installation program to upgrade your 5.x.x Server to version 5.3.2.

> **Note:** Do not run the installation program as root on Linux or UNIX systems. Log in as a non-root user to run the installation program.

1. Double-click the installation file on the CD to launch the automated installation program. Optionally, you can run the program from the command line to receive console output.

   - For Windows servers:
     `server\ApplicationConfigurationConsoleServer.exe`
   - For Linux and UNIX servers:
     `./server/ApplicationConfigurationConsole.bin`

   Click **Next** on the Welcome panel to proceed with the upgrade.

2. On the Installation Type panel, select **Upgrade** and then click **Next** to proceed with the upgrade.
3. On the Directory panel, verify the location of your existing Core Server installation and then click **Next** to proceed with the upgrade.
4. On the Upgrade Server panel, verify that the **Data retrieval status** is **Successful** for the three items listed.

   If data retrieval was not successful, stop at this point and contact Oracle technical support; otherwise, click **Next** to proceed with the upgrade.

5. Verify that Tomcat is not running and then click **Next** to proceed with the upgrade.

6. The Core Server requires a JDK. If it is already installed on this system, you can select the second option on this panel and then specify the JDK installation directory on the follow-on panel. If the JDK is not already installed, or if it is not the required version, select the first option and it will be installed with the Core Server.

   Click **Next** to proceed with the upgrade.

7. Specify the time zone in which the Core Server is being upgraded. Select a value appropriate to your geographic region from the drop-down menu. If your time zone is not listed, specify your location as a displacement from GMT/UTC in the format of '[+-}hh:mm'.

   Click **Next** to proceed with the upgrade.

8. Review the summary installation and click **Next** to begin the upgrade.

9. When the upgrade completes, click **Finish** to exit the wizard.

You have now completed the first part of the Core Server upgrade.

## A.6 Run the Upgrade Script on the Application Configuration Console Server

After you complete the installation part of the upgrade, you have to run an upgrade script on the Core Server to temporarily disable the Web Reports and tracking services. Run the upgrade script appropriate to your platform.

1. Navigate to the following directory:

   ```
   $OACC_INSTALL/appserver/tomcat/shared/scripts
   ```

2. (**Linux/UNIX**) Before you can run scripts on Linux or UNIX, you have to change the mode. Execute the following command:

   ```
   chmod +x *
   ```

3. Run the `dos2unix` command on all `.xml` and `.sh` files in the current directory:

   ```
   dos2unix *xml
   dos2unix *sh
   ```

4. Run the upgrade script appropriate to your platform, as follows:

```
Pre_MVUpgrade.sh
Pre_MVUpgrade.bat
```

5. You should see the following (or similar) confirmation messages:

```
Utility to upgrade mValent server from 5.x.x to 5.3.2.
Disabling mvwebreports.
Disabling mvwebreports context file.
Disabling mvtrack.
Disabling mvtrack context file.
```

## A.7 Start the Application Configuration Console Server

Following successful completion of the upgrade script, start the Core Server to verify certain log file entries.

1. Navigate to the following directory:

```
$OACC_INSTALL/appserver/tomcat/bin
```

2. (**Linux/UNIX**) Before you can run scripts on Linux or UNIX, you have to change the mode. Execute the following command:

```
chmod +x *
```

3. Execute the startup script appropriate to your platform as follows:

```
startup.sh
startup.bat
```

   (**Linux/UNIX**) If you receive messages of the form "bad interpreter: No such file or directory" when trying to run the script on Linux or UNIX, it typically means that invalid characters are present as a result of copying files from Windows. Use the `dos2unix` command to resolve the problem, as follows:

```
dos2unix startup.sh
```

   Then, run the corrected script again.

4. Navigate to the following directory:

```
$OACC_INSTALL/appserver/tomcat/logs/mv
```

5. Check the `mvServer.log` file for an entry that contains the following message:

```
Reload of MV_CONFIG asset completed successfully
```

6. If you see the message, return to the following directory:

```
$OACC_INSTALL/appserver/tomcat/bin
```

7. Execute the shutdown script appropriate to your platform as follows:

```
shutdown.sh
shutdown.bat
```

   You are stopping the Server at this point to ensure that the `mvserver` and `mvwebreports` services have the same database context the next time you start the Server.

## A.8 Run the Post-Upgrade Script on the Application Configuration Console Server

Now run the post-upgrade script on the Core Server to remove the upgrade service from `service_registry.xml` and to enable the Web Reports and tracking services. To do so, follow these instructions:

1. Navigate to the following directory:

   ```
   $OACC_INSTALL/appserver/tomcat/shared/scripts
   ```

2. Run the post-upgrade script appropriate to your platform, as follows:

   ```
   Post_MVUpgrade.sh
   Post_MVUpgrade.bat
   ```

3. You should see the following (or similar) confirmation messages:

   ```
   Utility to upgrade mValent server from 5.x.x to 5.3.2.
   Removing the version upgrade service from service_registry.xml.
   Done removing the version upgrade service.
   Enabling mvwebreports.
   Enabling mvwebreports context file.
   Enabling mvtrack.
   Enabling mvtrack context file.
   ```

## A.9 Upgrade Redeployed Secondary Servers

This step is necessary only if either or both of the secondary servers (tracking and Web Reports) were redeployed on the system being upgraded. For information on server redeployment, see Appendix D.

If the upgrade process detects that secondary servers have been redeployed, it creates a directory, `secondaryApps`, under the `$OACC_INSTALL` directory. If both tracking and Web Reports servers were redeployed, `secondaryApps` contains the following files:

```
mvtrack
mvtrack.xml
mvwebreports
mvwebreports.xml
```

To upgrade redeployed secondary servers:

1. Ensure that all Application Configuration Console primary and secondary servers are shut down.
2. On the Web Reports server (if applicable), back up the original files:
   a. Copy `$OACC_INSTALL/appserver/tomcat/conf/Catalina/localhost/mvwebreports.xml` to `mvwebreports.xml.bak`.
   b. Copy `$OACC_INSTALL/appserver/tomcat/webapps/mvwebreports` to `$OACC_INSTALL/appserver/tomcat`.

3. On the tracking server (if applicable), back up the original files:

   a. Copy `$OACC_INSTALL/appserver/tomcat/conf/Catalina/localhost/mvtrack.xml` to `mvtrack.xml.bak`.

   b. Copy `$OACC_INSTALL/appserver/tomcat/webapps/mvtrack` to `$OACC_INSTALL/appserver/tomcat`.

4. Move the Web Reports upgrade files (if applicable), from `$OACC_INSTALL/secondaryApps` on the primary server to the following locations on the Web Reports secondary server:

   ```
   $OACC_INSTALL/appserver/tomcat/webapps/mvwebreports
   $OACC_INSTALL/appserver/tomcat/conf/Catalina/localhost/mvwebreports.xml
   ```

5. Move the tracking upgrade files (if applicable), from `$OACC_INSTALL/secondaryApps` on the primary server to the following locations on the tracking secondary server:

   ```
   $OACC_INSTALL/appserver/tomcat/webapps/mvtrack
   $OACC_INSTALL/appserver/tomcat/conf/Catalina/localhost/mvtrack.xml
   ```

## A.10 If You Use UNC

If you used UNC, or Uniform (Universal) Naming Convention, in an earlier release, or if you want to use it going forward, you must takes the steps below to acquire and install third-party software, and configure Application Configuration Console to use it.

> **Note:** If you previouslydefined UNC endpoints, they will no longer work, unless you follow the instructions below.

To get the required JCIFS software:

1. Point your browser at the following URL:

   ```
   http://jcifs.samba.org/src/
   ```

2. Download `jcifs-1.3.12.jar` to a temporary directory. If this version is not available, download a later version. If the later version proves problematic, contact Support.

   Copy the `.jar` file to the following three locations in the Core Server installation directory:

   ```
   $OACC_INSTALL/appserver/tomcat/webapps/mvserver/WEB-INF/lib
   $OACC_INSTALL/appserver/tomcat/webapps/mvtrack/WEB-INF/lib
   $OACC_INSTALL/appserver/tomcat/webapps/mvwebreports/WEB-INF/lib
   ```

> **Note:** If you are upgrading an environment with server redeployments, you will have to copy the `.jar` file to the appropriate place in the installation directory on the redeployed server's host machine.

To configure Application Configuration Console for UNC support:

1. Navigate to the following location in the installation directory:

   ```
   $OACC_INSTALL/appserver/tomcat/shared/classes
   ```

2. Open `authpack_specification.xml` in a text or XML editor and add `UNC` to the first `Type authpack` element so that it appears as follows:

   ```
   <Type authpack="Username Password" worksWithEnpoint="FTP, SSH, JDBC, UNC" />
   ```

3. Save the file.
4. In the same directory, open `endpoint_specification.xml` in a text or XML editor and uncomment the `<EndpointSpec type="UNC">...</EndpointSpec>` element.
5. Save the file.

A UNC option will now be available in the Application Configuration Console Client user interface.

## A.11 Start the Application Configuration Console Server Again

Start the Core Server. Even if you have configured it to start as a service in Windows, you should start it using the icon or the command line so that you can see the console log messages. Also start the secondary servers, if applicable.

### A.11.1 Preserve Your Certificates

If you use company-specific SSL certificates (server or root certificates) to communicate with LDAP, you will need to export them from the old JDK into the new (upgrade version) JDK. Certificates are typically stored in the JDK's `cacerts` file, which, following the upgrade, can be found in the following directory:

```
$OACC_INSTALL/mv_backup/Java/jre/lib/security
```

Install the exported `cacerts` file in the following directory of the new JDK:

```
$OACC_INSTALL/Java/jre/lib/security
Typically, you install certificates using the JDK's keytool binary ($OACC_
INSTALL/Java/bin).
```

Typically, you install certificates using the JDK's `keytool` binary (`$OACC_INSTALL/Java/bin`).

## A.12 Uninstall and Reinstall the Application Configuration Console Clients

There is no upgrade option for Clients because they do not store any data that needs to be upgraded. Instead, uninstall the 5.x.x Clients and then reinstall the 5.3.2 Clients. You can use the Uninstall command from the Windows Start menu, or execute the following:

```
$OACC_INSTALL\uninstaller\UninstallClient.exe
```

Before uninstalling, ensure that there are no active Client instances and that there are no `.lock` files in the following directory:

```
$OACC_INSTALL\runtime\workspace\metadata
```

If there are no active clients and you do see a `.lock` file, a client terminated abnormally. Delete any such .lock files before uninstalling. Also, don't uninstall if you want to preserve your preference settings (see the next section).

You must install at least one version 5.3.2 Client so that you can install automation modules, but you do not need to install all 5.3.2 Clients at this time.

### A.12.1 Do You Want to Preserve Preferences?

You can set preferences in the Client to tailor system behavior in a variety of ways. For example, you can change display columns in tracking and comparison alerts, and make tags mandatory on provision and update operations. For descriptions of all preference settings, see the *Application Configuration Console Client Online Help*.

To preserve the preference settings from the previous version:

1. Before uninstalling the old Client version, navigate to the following location:

   ```
   %USERPROFILE%\Application Data\ApplicationConfigurationConsole
   ```

   Where `%USERPROFILE%` defaults to `C:\Documents and Settings\`*`username`*.

2. Save copies of the following files:

   ```
   mvApp.properties
   mvAppTrack.properties
   mvAppComparison.properties
   ```

   If neither of the last two files exists, it simply means that you hadn't set those particular preferences.

3. After installing the new Client version, copy the files you saved to the following location:

   ```
   %USERPROFILE%\Application Data\ApplicationConfigurationConsole
   ```

The settings from the old Client version will be in effect when you start up the new Client version.

## A.13 Install 5.3.2 Extensions

If you use WebSphere Automation Module or WebLogic Automation Module, start the Application Configuration Console Client and log in as an administrator, then follow the instructions that appear in Chapter 7.

If you use Windows Resource Extensions, note that OpenSSH is no longer bundled with the product. See Chapter 8 for details.

## A.14 Verify Data

Log in as an administrator. Expand the **My Workspace** and **Public Workspace** folders and verify that your configuration data is there.

# A.15 Mapping Considerations

Before resuming operations, consider the following issues related to file mapping in Application Configuration Console:

- If you are upgrading from 5.2.1 to 5.3.2, you have to make certain edits to the `mapping_registry.xml` file.
- If you are doing any other 5.xx to 5.3.2 upgrade, you may want to update the mapping registry.

## A.15.1 5.2.1 to 5.3.2 Upgrades and Mapping

If you are upgrading from 5.2.1 to 5.3.2, you have to update class references in `mapping_registry.xml` as follows:

1. In the Navigator view, expand **System Configuration > Common Configuration > MV_CORE > Resource View**.
2. Right-click `mapping_registry.xml` and select **Open** in the popup menu.
3. In the Editor area, enable editing and then right-click and select **Find and Replace**.
4. Replace all occurrences of this string:

   ```
   "com.mvalent.ext.columnar.mapping.ColumnarParser"
   ```

   With this string:

   ```
   "com.mvalent.ext.regexcolumnar.mapping.RegexColumnarParser"
   ```

5. Replace all occurrences of this string:

   ```
   "com.mvalent.ext.columnar.mapping.ColumnarTransform"
   ```

   With this string:

   ```
   "com.mvalent.ext.regexcolumnar.mapping.RegexColumnarTransform"
   ```

6. Replace all occurrences of this string:

   ```
   "com.mvalent.ext.properties.mapping.PropertiesParser"
   ```

   With this string:

   ```
   "com.mvalent.ext.regexproperties.mapping.RegexPropertiesParser"
   ```

7. Replace all occurrences of this string:

   ```
   "com.mvalent.ext.properties.mapping.PropertiesTransform"
   ```

   With this string:

   ```
   "com.mvalent.ext.regexproperties.mapping.RegexPropertiesTransform"
   ```

8. Save your edits.
9. In the Navigator view, right-click **System Configuration** and select **Make Configuration Changes Live**.

## A.15.2 Update the Mapping Registry

The upgrade process does not automatically update the mapping registry, so as not to overwrite any customized mappings that may exist in the previous version. In recent releases, new mappings have been introduced. Thus, at this point in the upgrade process, the database copy of the mapping registry typically is different from the file system copy shipped on the distribution media.

If you want to make use of the new mappings or continue to use your customized mappings, you have to update the mapping registry.

> **Note:** Updating the mapping registry may impact existing resource specifications and assets already loaded, depending on the mappings they use. If you intend on updating the mapping registry, Oracle recommends that you contact technical support for guidance.

To update the mappings registry, do the following:

1. Open the existing `mapping_registry.xml` file and cut and paste into a text file any customized mappings you want to preserve.
2. In the Navigator view, expand **System Configuration > Common Configuration > MV_CORE > Resource View**.
3. Right-click `mapping_registry.xml` and select **Synchronize > Update from Resource** in the popup menu. This makes the Application Configuration Console database and the file system versions in synch.
4. Merge any customizations you saved in Step 1 into `mapping_registry.xml`.
5. In the Navigator view, right-click **System Configuration** and select **Make Configuration Changes Live**.

Verify that the updated list of mappings appears in the Advanced Resource Definition Parameter Configuration dialog in the Client.

# B

# Security Concerns

This appendix covers several aspects related to Application Configuration Console security.

## B.1 Best Practices

In keeping with Oracle's mandate to protect customer data, Application Configuration Console observes the following best practices:

- Uses the AES (Advanced Encryption Standard) encryption algorithm, with an encryption key size of 128 bits and an encryption mode of CBC (cipher-block chaining).
- Imposes 0750 permission on `SERVER_INSTALL/appserver/tomcat/logs`
- Imposes 600 permission on `SERVER_INSTALL/appserver/tomcat/conf/Catalina/localhost/*.xml`
- Imposes 600 permission on `SERVER_INSTALL/appserver/tomcat/conf/server.xml`
- Imposes 600 permission on the following files in `SERVER_INSTALL/appserver/tomcat/shared/classes/`
  - `java.login.config`
  - `mvstore`
  - `server_modules_registry.xml`

Application Configuration Console stores passwords for keystore and truststore in obfuscated format on the Server:

`SERVER_INSTALL/appserver/tomcat/conf/server.xml`

And on the Client:

`CLIENT_INSTALL/client/runtime/plugins/com.mvalent.integrity_5.3.2/config/client_boot.xml`

If you suspect that these files or the keystore have been compromised in any way, contact Oracle support for assistance in changing passwords.

## B.2 Generating New Keystore and Truststore Files

To ensure a secure connection, the Core Server and the Client use the SSL protocol to exchange sensitive information contained in files known as a keystore and a truststore. A keystore contains private keys, and the certificates with their corresponding public keys. A truststore contains certificates from other parties that you expect to communicate with, or from Certificate Authorities that you trust to identify other parties. To learn more about the Java SSL protocol and keystores and truststores, visit the following URL:

`http://java.sun.com/docs/books/tutorial/security/sigcert/index.htm`

The Core Server generates a self-signed certificate during installation. The keystore files (keystore and truststore) are generated the first time the Client connects to the Server. If something happens to the Client's keystore files (modified or deleted, for example), they are regenerated on the next startup, when you are prompted to accept the certificate. If something happens to the Server's keystore files, however, or if tampering is suspected, then it becomes necessary to generate new keystore files to continue secure operations.

### B.2.1 Before You Begin

Back up either or both original files if they exist (`mvserver.ks` and `mvserver.ts`), located in the following directory:

`$OACC_INSTALL/com.mvalent.integrity_5.3.2/webserver/tomcat`

You will use the `keytool` utility that comes with any JDK 1.6 installation to regenerate your keystore files. This utility displays passwords in cleartext so be sure to take the necessary security precautions. Also, you may want to record the values that you supply, such as passwords and paths, as you will need to provide them several times during the process. To learn more about `keytool` visit the following URL:

`http://java.sun.com/javase/6/docs/technotes/tools/windows/keytool.html`

### B.2.2 Generate a New Keystore

To generate a new keystore file, proceed as follows:

1. Open a command shell prompt.
2. Change directory to the `JDK1.6_HOME/bin` directory; to here, for example, if you elected to use the version embedded with the Core Server installation:

   `$OACC_INSTALL/java/bin`

3. Execute the `keystool` command as follows:

   `keytool -genkey -alias` *`alias_value`* `-keyalg RSA -keysize 1024 -storepass` *`password`* `-keypass` *`password`* `-keystore` *`store_path`* `-storetype jks -dname` *`dname_values`*

   Where:

   - *alias_value* is the identifier of the original key created during keystore creation. This value can be any string.
   - *password* is a strong password for accessing the keystore file. Tomcat requires that you specify the same value to access the keystore and its private key.
   - *store_path* is the path of the keystore file that you are generating.

- *dbname_values* are optional values that identify the owner of the credentials passed from the Server to any Client. For example: `"CN=Application Configuration Console Server, OU=Enterprise Manager Grid Control, O=Oracle Corporation,L=Redwood City, S=California, C=US"`

---

**Note:** If you receive the following error: `"keytool error: java.io.IOException: Keystore was tampered with, or password was incorrect"`, it probably means that the value you specified for `store_path` already exists. Move or rename the file and rerun `keytool`. If this is not the case, contact support.

---

4. Verify that the keystore file was created at the specified location. It should be about 2KB in size.

### B.2.3 Generate a New Truststore

To generate a new truststore file, proceed as follows:

1. Open a command shell prompt.
2. Change directory to the `JDK1.6_HOME/bin` directory; to here, for example, if you elected to use the version embedded with the Core Server installation:

   ```
   $OACC_INSTALL/java/bin
   ```

3. Execute the `keystool` command as follows:

   ```
   keytool -genkey -alias alias_value -keyalg RSA -keysize 1024 -storepass password -keypass password -keystore store_path -storetype jks -dname dname_values
   ```

   Where:

   - *alias_value* is the identifier of the original key created during truststore creation. This value can be any string.
   - *password* is a strong password for accessing the truststore file. Tomcat requires that you specify the same value to access the truststore and its private key.
   - *store_path* is the path of the truststore file that you are generating.
   - *dbname_values* are optional values that identify the owner of the credentials passed from the Server to any Client. For example: `"CN=Application Configuration Console Server, OU=Enterprise Manager Grid Control, O=Oracle Corporation,L=Redwood City, S=California, C=US"`

---

**Note:** If you receive the following error: `"keytool error: java.io.IOException: Keystore was tampered with, or password was incorrect"`, it probably means that the value you specified for `store_path` already exists. Move or rename the file and rerun `keytool`. If this is not the case, contact support.

---

4. Verify that the truststore file was created at the specified location. It should be about 2KB in size.

### B.2.4 Update Tomcat server.xml

Update `server.xml` to include the values.

1. Navigate to the following directory on the Core Server host and open `server.xml` in a text or XML editor:

   ```
   $OACC_INSTALL/appserver/tomcat/conf
   ```

2. At the bottom of the file, locate the XML element `<Connector port="9943" ... />` and edit the values of these four properties (`keystoreFile, keystorePass, truststoreFile, truststorePass`) with the values specified during keystore/truststore generation.
3. Save your changes and restart the Core Server. You can restart from command line as follows: `$OACC_INSTALL/appserver/tomcat/bin/startup.bat` (for Windows) or `startup.sh` for (Linux/UNIX).

## B.3 Disable Anonymous Read Write Access on the SVN Server

Some environments may require a layer of security between the Core Server and the SVN server. This section provides instructions for requiring authentication on the SVN server, and disabling anonymous read and write access.

First, you should encrypt an Application Configuration Console password, using the `MVEncryption.bat` file as follows:

1. Navigate to the following directory:

   ```
   $OACC_INSTALL/appserver/tomcat/shared/scripts
   ```

2. Run the following command:

   ```
   MVEncryption.bat mvpassword
   ```

   Command output resembles the following:

   ```
   Encrypting [mvpassword]

   Encrypted characters: [67|115|98|97|83|81|100|53|82|90|67|122|73|109|99|111|70]
   Encrypted string....: [CsbaSQd5RZCzImcoF45kmg=]
   ```

3. Make a copy of the Encrypted string value between the brackets (shown in bold in the example). This is your encrypted password.

Now do the following:

1. Make the following changes to the `svnserve.conf` file in `$OACC_INSTALL/svn/db/conf`:

   a. Uncomment the following line:

      ```
      password-db=mvuserfile
      ```

   b. Change the access in the general section to the following values:

      ```
      [general]
      anon-access = none
      auth-access = write
      ```

2. In $OACC_INSTALL/svn/db/conf, create an ASCII file named mvuserfile with the following contents:

```
[USERS]
mvadmin=unencryptedpassword
```

3. Set permissionss on mvuserfile such that the Application Configuration Console-specific operating system user (OACCUSER) has at least read access.
4. Add username and password property values to the following module in server_modules_registry.xml:

```
<module name="com.mvalent.service.system.repository.version.impl.
subversion.SvnSessionContext">

<property name="username" value="mvadmin"/>
<property name="password" value="encryptedpassword"/>\
```

5. Make this change to the versions of server_modules_registry.xml in the following locations:

```
$OACC_INSTALL/appserver/tomcat/shared/classes/
$OACC_INSTALL/appserver/tomcat/webapps/mvtrack/WEB-INF/classes
$OACC_INSTALL/appserver/tomcat/webapps/mvwebreports/WEB-INF/classes
```

   Anonymous read and write access is now disabled on the SVN server.

## B.4 Optionally Use a Customer-Supplied SSL Certificate

The Core Tomcat server uses an SSL certificate to ensure secure communication with the Application Configuration Console Clients. If desired, you can use your own certificate instead of the one supplied by Oracle. The certificate can be JKS or PKCS #12 format, and you must have the associated private key. With PKCS #12, for example, do the following:

1. Stop the Core Server and Clients if they are running.
2. Use OpenSSL to produce a keystore file from your certificate file:

```
> openssl pkcs12 -export -in mycertificate.cer -inkey mycertificate.key
-out mvserver.ks -name tomcat
```

3. Copy the mvserver.ks file for use by the Clients:

```
> cp mvserver.ks mvclient.ts
```

4. On the Core Server system, rename the existing mvserver.ks file:

```
> cd $OACC_INSTALL/com.mvalent.integrity_5.3.2/webserver/tomcat
> rename mvserver.ks mvclient.ks.orig
```

5. Copy the new mvserver.ks file that you created in Step 2 to the following directory:

```
$OACC_INSTALL/com.mvalent.integrity_5.3.2/webserver/tomcat
```

6. Open the $OACC_INSTALL/appserver/tomcat/conf/server.xml file in a text editor.
7. Move to the end of the file and look for the <Connector> element that starts with port="9943".

8. Change the `keystorePass` attribute in the `Connector` element to the password required by your certificate.
9. Save and close the `server.xml` file.
10. On each Client machine, copy the `mvclient.ts` file that you created in Step 3 to the following directory:

    `$OACC_INSTALL\runtime\plugins\com.mvalent.integrity_5.3.2\config.`

11. On each Client, navigate to the following directory and open `client_boot.xml` in a text editor:

    `$OACC_INSTALL\runtime\plugins\com.mvalent.integrity_5.3.2\config`

12. Change the value of the `trustStorePassword` to the password required by your certificate.
13. Save and close the `client_boot.xml` file.
14. Restart the Core Server and Clients.

# C

# LDAP Setup for SSL

This appendix describes how to set up the Core Server to communicate with LDAP over an SSL connection. Your LDAP server is assumed to have a signed certificate and to be enabled for SSL.

Application Configuration Console uses the Sun LDAP service provider, which uses the Java Secure Socket Extension (JSSE) software for its SSL support. The JSSE is available as part of the JDK. The premise of the setup is to ensure that the Core Server, as the LDAP client, trusts the LDAP server it uses. The setup is the same, regardless of which LDAP directory you have.

The first step, which is optional, is to import the root CA certificate that was used to sign the certificate in use by the LDAP server.

> **Note:** You only need to do this if the certificate in use by the LDAP server is not signed by one of the multitude of trusted certificate authorities listed in the cacerts keystore shipped by Sun Microsystems.

If you need to import the root CA certificate into your JRE's database of trusted certificates, run commands similar to the following:

```
cd JAVA_HOME/jre/lib/security

keytool -import -v -trustcacerts -file /cert-path/509_cert -keystore
/cert-path/cacerts
```

Where *cert-path* is the path to the respective certificate.

When you run `keytool`, you will be prompted for a password. Assuming you are using the default keystore, the default password is `changeit`.

The next step, which is required, is to edit the `java.login.config` file. Specifically, you have to change the LDAP URL scheme to ldaps and the port to an SSL-enabled port, in the appropriate section for the login module (AD-JNDI or MV-JNDI) you are using. This will force the Sun LDAP service provider to use SSL for communications.

# D

# Set Up Secondary Server Instances

For performance and other considerations, it may be beneficial to set up secondary server instances on other host machines. This appendix describes two scenarios:

- Redeployment of secondary servers (Web Reports server and tracking server) from the Application Configuration Console Server installation host where the primary (Core) server resides
- Deployment of additional tracking servers

This appendix provides instructions on how to set up secondary servers for Application Configuration Console. The instructions for each scenario are similar; briefly:

- Shut down Application Configuration Console.
- Make a copy of the complete Application Configuration Console installation.
- Place the copy on the target (secondary) host.
- On the primary host, disable the server being deployed.
- On the target host, disable the servers that will continue to run on the primary host.
- Edit configuration files appropriately.
- Restart the primary and secondary servers.

> **Note:** The scenarios described in this appendix involve setup of another instance of a Tomcat JVM on a separate host machine. This is Oracle's recommendation. If you have any questions, contact Oracle Technical Support.

Before proceeding, ensure that the host where you are setting up a secondary server complies with the hardware and software requirements stated in Table 2–2 in Chapter 2. Oracle recommends that your product implementation be complete before you set up any secondary server instances. So, for example, if you intend to use automation modules, install them prior to secondary server setup.

# D.1 Secondary Server Target Environment

The environment where you set up secondary servers should meet or exceed the hardware and software requirements of the Core Server host. The instructions tell you to relocate the entire Core Server installation directory so that everything else—configurations, user IDs and groups and so forth—is in place in the target environment. Additionally, the instructions assume that you are going to the same platform, for example, Linux to Linux.

# D.2 Server Redeployment from the Installation Host

An Application Configuration Console installation creates separate deployments of the primary and secondary servers on the host machine. This facilitates server redeployment to another instance of a Tomcat JVM. The ability to redeploy these servers is an important aspect in any discussion of Application Configuration Console performance and scalability. Redeployment of secondary servers reduces the load on the Core Server so that response times are faster for other tasks, such as compare and provision operations.

This section provides instruction on redeploying the Web Reports server and the tracking server from the installation host to a target host.

## D.2.1 Redeploying Web Reports Server

To redeploy the Web Reports server:

1. Shut down Application Configuration Console on the primary host.
2. Compress the entire Core Server installation directory on the primary host.
3. Copy the compressed file and unpack it in the same location on the target host. For example, if you installed the Core Server at `opt/oracle/oacc/server/` on the primary host, unpack it to `opt/oracle/oacc/server/` on the target host.
4. On the primary host, disable the Web Reports deployment as follows:
   a. Rename `$OACC_INSTALL/appserver/tomcat/conf/Catalina/localhost/mvwebreports.xml` to `mvwebreports.xml.bak`.
   b. Move `$OACC_INSTALL/appserver/tomcat/webapps/mvwebreports` to `$OACC_INSTALL/appserver/tomcat`.
5. On the target host, disable the Core Server and tracking server deployments as follows:

   **Core Server**

   a. Rename `$OACC_INSTALL/appserver/tomcat/conf/Catalina/localhost/mvserver` to `mvserver.xml.bak`.
   b. Move `$OACC_INSTALL/appserver/tomcat/webapps/mvserver` to `$OACC_INSTALL/appserver/tomcat`.

**Tracking Server**

a. Rename `$OACC_INSTALL/appserver/tomcat/conf/Catalina/localhost/mvtrack` to `mvtrack.xml.bak`.

b. Move `$OACC_INSTALL/appserver/tomcat/webapps/mvtrack` to `$OACC_INSTALL/appserver/tomcat`.

6. On the primary host, change the hostname of the Web Reports URL so that alerts sent in e-mail notifications are redirected appropriately:

   a. Open `$OACC_INSTALL/appserver/tomcat/shared/classes/server_modules_registry.xml`.

   b. For tracking alerts, search for the following property and change hostname to the name of the target host:

   ```
   <property name="tracking.alert.url"
   value="https://hostname.mvalent.local:9943/mvwebreports" />
   ```

   c. For comparison alerts, search for the following property and change hostname to the name of the target host:

   ```
   <property name="comparison.alert.url"
   value="https://hostname.mvalent.local:9943/mvwebreports" />
   ```

   d. Save your changes.

   e. If the tracking server remains on the primary host, make the same changes to `$OACC_INSTALL/appserver/tomcat/webapps/mvtrack/WEB-INF/classes/server_modules_registry.xml`

7. On the target host, back up `$OACC_INSTALL/appserver/tomcat/shared/classes/server_modules_registry.xml`, then copy the version of `server_modules_registry.xml` that you edited in Step 6 from the primary host to the same location on the target host.

8. **Linux/UNIX only**. Disable automatic startup and shutdown of the SVN server on the target host, as follows:

   a. Navigate to the following directory:

   ```
   $OACC_INSTALL/appserver/tomcat/bin
   ```

   b. Remove the following line from the `startup.sh` script and save your changes:

   ```
   $OACC_INSTALL/svn/bin/svnserve -d --root "$OACC_INSTALL/svn" --listen-port
   nnnn --pid-file $OACC_INSTALL/appserver/tomcat/bin/svn_pid
   ```

   c. Remove the following line from the the `shutdown.sh` script and save your changes:

   ```
   cat svn_pid | xargs kill
   ```

9. Start Application Configuration Console on the primary and target hosts.

After redeploying the Web Reports server, you may want to edit the Start menu shortcut on the Client host (**All Programs >Oracle Application Configuration Console > Client >Start Web Reports**), as described in Section 6.4, "Repointing Web Reports."

## D.2.2 Redeploying the Tracking Server

To redeploy the tracking server:

1. Shut down Application Configuration Console on the primary host.
2. Compress the entire Core Server installation directory on the primary host.
3. Copy the compressed file and unpack it in the same location on the target host. For example, if you installed the Core Server at `opt/oracle/oacc/server/` on the primary host, unpack it to `opt/oracle/oacc/server/` on the target host.
4. On the primary host, disable the tracking deployment as follows:
   a. Rename `$OACC_INSTALL/appserver/tomcat/conf/Catalina/localhost/mvtrack.xml` to `mvtrack.xml.bak`.
   b. Move `$OACC_INSTALL/appserver/tomcat/webapps/mvtrack` to `$OACC_INSTALL/appserver/tomcat`.
5. On the target host, disable the Core Server and Web Reports deployments as follows:

   **Core Server**

   a. Rename `$OACC_INSTALL/appserver/tomcat/conf/Catalina/localhost/mvserver` to `mvserver.xml.bak`.
   b. Move `$OACC_INSTALL/appserver/tomcat/webapps/mvserver` to `$OACC_INSTALL/appserver/tomcat`.

   **Web Reports Server**

   a. Rename `$OACC_INSTALL/appserver/tomcat/conf/Catalina/localhost/mvwebreports` to `mvwebreports.xml.bak`.
   b. Move `$OACC_INSTALL/appserver/tomcat/webapps/mvwebreports` to `$OACC_INSTALL/appserver/tomcat`.
6. **Linux/UNIX only**. Disable automatic startup and shutdown of the SVN server on the target host, as follows:
   a. Navigate to the following directory:

      ```
      $OACC_INSTALL/appserver/tomcat/bin
      ```

   b. Remove the following line from the `startup.sh` script and save your changes:

      ```
      $OACC_INSTALL/svn/bin/svnserve -d --root "$OACC_INSTALL/svn" --listen-port nnnn --pid-file $OACC_INSTALL/appserver/tomcat/bin/svn_pid
      ```

   c. Remove the following line from the the `shutdown.sh` script and save your changes:

      ```
      cat svn_pid | xargs kill
      ```
7. Start Application Configuration Console on the primary and target hosts.

# D.3 Multiple Tracking Server Deployment

The notion in this case is that you have more than one tracking server handling tracking operations in your environment. Introducing multiple tracking servers can relieve resource bottlenecks on particular machines.

For the purposes of this discussion, the current tracking server is the primary tracking server, and any additional deployments are secondary tracking servers. This section considers two possibilities:

- Deploying the tracking server on the installation host
- Deploying a redeployed tracking server

The system captures as part of the tracking operation record which tracking server instance executed the operation. The information (host name and unique tracking server ID) appears in the appropriate Client scheduled jobs list (tracking, comparisons, provisioning). See the *Application Configuration Console Online Help* for details.

## D.3.1 Deploying the Installed Tracking Server

To deploy a secondary tracking server based on the installed primary tracking server:

1. Shut down Application Configuration Console on the primary host.
2. Compress the entire Core Server installation directory on the primary host.
3. Copy the compressed file and unpack it in the same location on the target host. For example, if you installed the Core Server at `opt/oracle/oacc/server/` on the primary host, unpack it to `opt/oracle/oacc/server/` on the target host.
4. On the target host, remove the Core Server and Web Reports deployments as follows:

   **Core Server**

   a. Remove `$OACC_INSTALL/appserver/tomcat/conf/Catalina/localhost/mvserver`.

   b. Remove `$OACC_INSTALL/appserver/tomcat/webapps/mvserver` .

   **Web Reports Server**

   a. Remove `$OACC_INSTALL/appserver/tomcat/conf/Catalina/localhost/mvwebreports`.

   b. Remove `$OACCm_INSTALL/appserver/tomcat/webapps/mvwebreports`.

5. Assign a unique server ID to the secondary tracking server deployment, as follows:

   a. On the target host, open the following file in a text or xml editor:

   ```
   $OACC_INSTALL/appserver/tomcat/webapps/mvtrack/WEB-INF/web.xml.
   ```

   b. Edit the value of the `mv.server.id <param-name>` element so that it is unique across the global namespace of all server instances. A simple approach is to add a numerical sequence to the `mvtrack` qualifier of the default tracking server ID; for example:

   ```
   mvtrack-2:764c1a8796092b09f27a320bdb8d9f2f
   ```

Sequentially increase the number for each secondary tracking server deployment. The only stipulation besides uniqueness is that the server ID cannot exceed 255 characters.

c. Save the file.

6. Add the secondary tracking server ID to the event distribution list, as follows:

a. On the primary host, open the following file in a text or xml editor:

```
$OACC_INSTALL/appserver/tomcat/shared/classes/server_modules_registry.xml
```

b. Append to the value of the `system.event.destination.server.ids` property the unique tracking server ID assigned to the secondary tracking server deployment; for example:

```
<property name="system.event.destination.server.ids"
value="mvtrack:764c1a8796092b09f27a320bdb8d9f2f,mvweb:a64c1a8796092b09f27a3
20bdb8d9e2e,mvtrack-2:764c1a8796092b09f27a320bdb8d9f2f"/>
```

Add other unique secondary tracking server IDs, as appropriate, separated by commas.

c. Save the file.

d. If the Web Reports server remains on the primary host, make the same changes to `$OACC_INSTALL/appserver/tomcat/webapps/mvwebreports/WEB-INF/classes/server_modules_registry.xml`

7. **Linux/UNIX only**. Disable automatic startup and shutdown of the SVN server on the target host, as follows:

a. Navigate to the following directory:

```
$OACC_INSTALL/appserver/tomcat/bin
```

b. Remove the following line from the `startup.sh` script and save your changes:

```
$OACC_INSTALL/svn/bin/svnserve -d --root "$OACC_INSTALL/svn" --listen-port
nnnn --pid-file $OACC_INSTALL/appserver/tomcat/bin/svn_pid
```

c. Remove the following line from the the `shutdown.sh` script and save your changes:

```
cat svn_pid | xargs kill
```

8. Start Application Configuration Console on the primary and target hosts.

Once you have a deployed secondary tracking server, it becomes a simple task to deploy additional tracking servers, based on this deployment. All you have to do is change the server ID on the next deployed tracking server instance, and add the ID to the event distribution list on the Core Server.

## D.3.2 Deploying a Redeployed Tracking Server

To deploy a secondary tracking server based on a redeployed primary tracking server:

1. Shut down Application Configuration Console on the Core Server and on th e redeployed tracking server host.

2. Compress the entire Core Server installation directory on the primary tracking server host.

3. Copy the compressed file and unpack it in the same location on the secondary tracking server host. For example, if you redeployed to `opt/oracle/oacc/server/` on the primary tracking server host, unpack it to `opt/oracle/oacc/server/` on the secondary tracking server host.
4. On the secondary tracking server host, remove the Core Server and Web Reports deployments as follows:

   **Core Server**

   a. Remove `$OACC_INSTALL/appserver/tomcat/conf/Catalina/localhost/mvserver`.
   b. Remove `$OACC_INSTALL/appserver/tomcat/webapps/mvserver`.

   **Web Reports Server**

   a. Remove `$OACC_INSTALL/appserver/tomcat/conf/Catalina/localhost/mvwebreports`.
   b. Remove `$OACCm_INSTALL/appserver/tomcat/webapps/mvwebreports`.
5. Assign a unique server ID to the secondary tracking server deployment, as follows:

   a. On the secondary tracking server host, open the following file in a text or xml editor:

      `$OACC_INSTALL/appserver/tomcat/webapps/mvtrack/WEB-INF/web.xml`.

   b. Edit the value of the `mv.server.id <param-name>` element so that it is unique across the global namespace of all server instances. A simple approach is to add a numerical sequence to the `mvtrack` qualifier of the default tracking server ID:

      ```
      mvtrack-2:764c1a8796092b09f27a320bdb8d9f2f
      ```

      Sequentially increase the number for each secondary tracking server deployment. The only stipulation besides uniqueness is that the server ID cannot exceed 255 characters.

   c. Save the file.
6. Add the secondary tracking server ID to the event distribution list, as follows:

   a. On the Core Server host, open the following file in a text or xml editor:

      `$OACC_INSTALL/appserver/tomcat/shared/classes/server_modules_registry.xml`

   b. Append to the value of the `system.event.destination.server.ids` property the unique tracking server ID assigned to the secondary tracking server deployment; for example:

      ```
      <property name="system.event.destination.server.ids"
      value="mvtrack:764c1a8796092b09f27a320bdb8d9f2f,mvweb:a64c1a8796092b09f27a3
      20bdb8d9e2e,mvtrack-2:764c1a8796092b09f27a320bdb8d9f2f"/>
      ```

      Add other unique secondary tracking server IDs, as appropriate, separated by commas.

   c. Save the file.

d. If the Web Reports server remains on the primary host, make the same changes to `$OACC_INSTALL/appserver/tomcat/webapps/mvwebreports/WEB-INF/classes/server_modules_registry.xml`

7. Start Application Configuration Console on the Core Server, and primary and secondary tracking servers.

Once you have a deployed secondary tracking server, it becomes a simple task to deploy additional tracking servers, based on this deployment. All you have to do is change the server ID on the next deployed tracking server instance, and add the ID to the event distribution list on the Core Server.

## D.4 Verify Server Redeployment

Use this process to verify Web Reports and tracking server redeployment, as well as secondary tracking server deployment.

1. Start the Core Server.
2. Start the tracking server (redeployed and deployed secondary, as appropriate).
3. Start the Client.
4. Load some data and enable tracking; be sure to request e-mail notification on alerts.
5. Verify that the tracking server is handling the requests.
6. Verify that you can access Web Reports by clicking the URL in an e-mail notification.
7. For secondary tracking server deployments, verify in the Client that the scheduled jobs list identifies the appropriate host name and server ID of the secondary tracking server.

## D.5 Redeployment and Automation Modules

If you install an automation module after deploying secondary servers, you have to port the AM installation to the secondary server host.

1. On the primary host, navigate to the following location:

   `$OACC_INSTALL/appserver/tomcat/shared/classes/extensions`

2. Make a copy of the `extensions` folder.
3. On the secondary host (Web Reports, tracking, or both, as appropriate), place the copied folder at the following location:

   `$OACC_INSTALL/appserver/tomcat/shared/classes/`

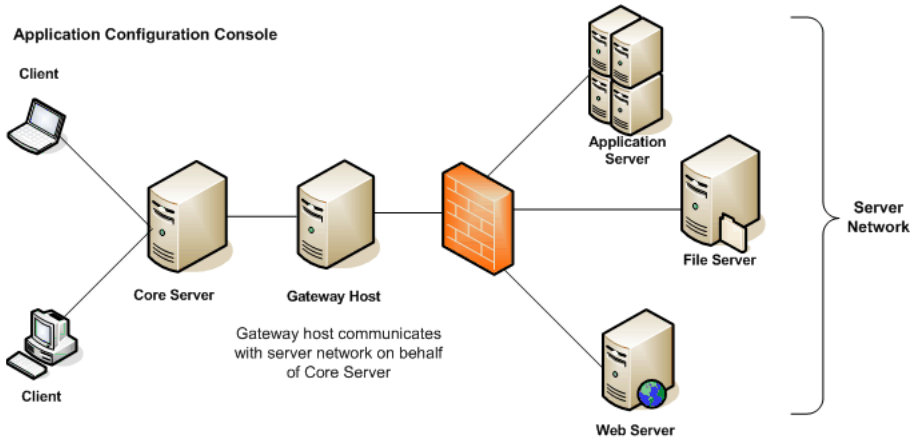
It is not necessary to restart the primary or the secondary server.

# E

# Application Configuration Console and SSH Tunneling

Many companies prefer not to expose their server network to remote access and so want to take security to the next level. The desired effect is to have remote applications such as Application Configuration Console communicate with a single point of contact that is trusted to handle communications with the servers on the network. The solution is SSH tunneling, also known as port forwarding, as depicted in Figure E–1.

***Figure E–1 Application Configuration Console and SSH Tunneling***




The single point of contact, the gateway host, is typically outside the firewall, but it can also be inside the firewall. To support this mode of communication, you have to perform setup on the network side and on the Application Configuration Console side.

## E.1 On the Network Side

The basic steps to set up SSH tunneling on your network are as follows:

1. Designate a host machine to be the gateway to your server network.
2. Install Cygwin (or a similar application that supports port forwarding) on the gateway host.
3. Enable a port on the gateway host for port forwarding of all SSH traffic to TCP port 22 on a server within your network that the Core Server needs to communicate with, for example:

   ```
   ssh -N -L 2424:server1:22 user1@server1
   ```

   This command enables TCP port 2424 for port forwarding to TCP port 22 on server1, where user1 is an authenticated user on server1. You would issue a similar command for each server in your network, changing the port number, host name, and authenticated user, as appropriate.

   **Note:** If the gateway is inside the firewall, you will have to open each port you enable for port forwarding.

## E.2 On the Application Configuration Console Side

The basic steps to set up SSH tunneling on Application Configuration Console are as follows:

1. In the Client, create a host and endpoint for the gateway host. Be sure to change the endpoint SSH default TCP port (22) to the port that you enabled for port forwarding on the gateway. In the example for setup on the gateway, this would be TCP port 2424.
2. In the Client, create an authentication pack with credentials that are valid on a server within your network. Remember that traffic coming in to the gateway is being forwarded to the Core Server's ultimate target, a network server, so the username and password need to be valid on the target, not on the gateway host. In the example for setup on the gateway, this would be credentials for user1.

   Repeat this step for each network server with which the Core Server is to communicate.
3. Test the setup by using the host and respective authentication pack you just created to ensure that you can browse the file system on each network server.

# F

# Internationalization and Localization

This appendix describes how Application Configuration Console supports internationalization and what has to be done to localize the product for a given locale.

## F.1 I18N Implementation in Application Configuration Console

Application Configuration Console provides internationalization support by externalizing message and UI elements in resource bundles so that the text content can be translated into the local language.

The resource bundles are contained in the following properties files:

- `plugin.properties`–contains values that appear in the About dialog, including View names and other Eclipse-related items.
- `MessagesBundle.properties`–contains text strings (informational and error messages; UI controls and labels) for all Application Configuration Console components, including the Core Server and Client, the three automation modules, and Web Reports.

> **Note:** A third properties file, `HelpBundle.properties`, contains strings that link to context-sensitive help topics. These strings are not help content and are therefore not candidates for translation.

The Application Configuration Console Client now includes controls that allow you to select an encoded character set to interpret file and directory names on the target system. You also select the encoded character set to interpret the contents of configuration files that you load into Application Configuration Console as assets.

All Application Configuration Console Server and Client files (.log,.xml,.txt, and so forth) are in UTF-8 format so that they are interpretable in any locale. Likewise, all server-generated e-mails are in UTF-8 format. Thus, the mail server must be able to forward e-mail correctly; that is, any problems with displaying/interpreting e-mail sent from Application Configuration Console stem from the mail server. Note, however, that the Core Server and Client installer panels remain in English, as assistance can be provided for those who need it.

## F.2 Localize the Database

The first step to localization is to create a new database in the appropriate encoded character set. For multibyte support, the only deviation from a standard default installation is to make the following selection on the **Character Sets** tab:

```
Use Unicode (AL32UTF8)
```

If you want to preserve your data, do a full export/import to properly convert all data to the new character set.

## F.3 Text Translation

After you localize the database, have the files translated into the language of the locale. Minimally, you will want to translate the resource bundles. You may also want to translate the Application Configuration Console Online Help.

If the translated properties file is encoded in a non-Latin-1 character set, convert the file using the Java `native2ascii` utility prior to placing it in the installation directory. Be sure to remove the BOM (byte order mark) character before or after the conversion.

With the translated text properly in place in the application, you can configure the system to support the local language.

## F.4 Localize Application Configuration Console

Application Configuration Console localization involvesa number of aspects as covered in this section.

### F.4.1 Preparing the Client Machine for Localization

The machine hosting the Client must be configured appropriately to support the local language, as follows:

- The requisite language packs and the fonts to support them must be installed on the host machine; otherwise, translated text will not display properly.
- **Control Panel > Regional and Language Options** settings must be configured to support the language and customs of the locale.
- For Web Reports, set the browser to display the appropriate language.

---

**Note:** Problems of interpretation or display can be attributable either to font availability or encoding support. Open (transparent) squares denote a font issue; question marks (????) indicate an encoding issue.

---

## F.4.2 Starting a Localized Client

To run a locale-specific version of the Client, you have to edit the `startup.bat` file.

1. On the Client machine, navigate to the following directory:

   ```
   $CLIENT_HOME/runtime
   ```

2. Open the `startup.bat` file in a text editor.
3. Add an argument similar to the following to the `start eclipse.exe` command.

   ```
   -nl locale
   ```

   where locale is the ISO language code of the local language, fr_FR, for example.
4. Save the file.

Upon startup, the Client checks for a matching `MessagesBundle_`*`isocode`*`.properties` file in the appropriate location. The Client retrieves from the original (English) version of `MessagesBundle.properties` any string that it does not find in the translated version.

## F.4.3 Localizing for Configurations

Application Configuration Console manages configurations on target systems. A localized version of the product needs to encode configuration file contents and the directory structure in which the files exist. This section describes how to configure the Client accordingly.

### F.4.3.1 Localizing the Target System Directory Structure

If a target system's file and directory names are in a different encoding from that of Application Configuration Console, the names will be corrupted. To circumvent the problem, set the encoding to match the target system when you create the host/endpoint in Application Configuration Console.

See the *Application Configuration Console Online Help* for instructions on how to set the encoding for the target host.

### F.4.3.2 Localizing Configuration Contents

To load asset configurations in Application Configuration Console in the encoding of the locale, you have to set the encoding for each resource definition as part of the process of creating a file resource specification.

See the *Application Configuration Console Online Help* for instructions on how to set the encoding for resource definitions.

## F.4.4 Localizing Custom Web Reports

You can integrate custom reports developed by Professional Services into Web Reports and make them accessible from the menu bar. The process is described in the *Web Reports Online Help*. For purposes of translation, you can edit the appropriate string values in `Messages.Bundle.properties`, specifically:

```
report.name.customreport=Custom Report
report.menu.custommenu=Custom Menu
```

These names correspond to the report definitions and menu items you add to the Web Reports Registry (`webreports_registry.xml`) to integrate the custom reports.

Remember that if the translated property file is encoded in a non-Latin-1 character set, you must convert the file using the Java `native2ascii` utility; for example:

```
native2ascii -encoding utf-8 MessagesBundle.fr MessagesBundle_fr.properties
```

And then place the property file appropriately on the Client and the Core Server, as follows:

```
$CLIENT_HOME/runtime/plugins/com.mvalent.integrity_5.3.2/config/MessagesBundle_
fr.properties
$SERVER_HOME/appserver/tomcat/shared/classes/MessagesBundle_fr.properties
```

Note that the translated version has a suffix appropriate to the local language.

## F.5 Localization Checklist

Use this checklist as a guideline to ensure proper localization setup.

- Have you localized the Oracle database?
- Does the local Windows environment support the proposed character encoding?
- What is the character encoding of the files and directories on the target host, that is, the host whose resources you want to manage?
- What is the character encoding of the configuration file contents that you want to manage?
- If your translated properties file is encoded in a non-Latin-1 character set, did you convert it using the Java `native2ascii` utility?
- Have you created a host/endpoint in Application Configuration Console that specifies the character encoding of the target host?
- Have you created a resource specification of resource definitions that specify the character encoding of the configuration files you want to manage?
- Were there any parsing problems during the resource specification creation process?
  - Ensure that you selected the correct character encoding for the resource definition.
  - Check that the default transform parameters are right for the configuration file. For example, you may need to change the comment character or end-of-line character.

  If you cannot resolve the problem, select **No Mapping**. This will allow you to load assets using the resource specification, which you can then track, compare, provision, and so forth.