



Installation Guide

Sybase Mobiliser Platform 5.1

SP03

DOCUMENT ID: DC01871-01-0513-01

LAST REVISED: September 2013

Copyright © 2013 by Sybase, Inc. All rights reserved.

This publication pertains to Sybase software and to any subsequent release until otherwise indicated in new editions or technical notes. Information in this document is subject to change without notice. The software described herein is furnished under a license agreement, and it may be used or copied only in accordance with the terms of that agreement.

Upgrades are provided only at regularly scheduled software release dates. No part of this publication may be reproduced, transmitted, or translated in any form or by any means, electronic, mechanical, manual, optical, or otherwise, without the prior written permission of Sybase, Inc.

Sybase trademarks can be viewed at the Sybase trademarks page at <http://www.sybase.com/detail?id=1011207>. Sybase and the marks listed are trademarks of Sybase, Inc. ® indicates registration in the United States of America.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world.

Java and all Java-based marks are trademarks or registered trademarks of Oracle and/or its affiliates in the U.S. and other countries.

Unicode and the Unicode Logo are registered trademarks of Unicode, Inc.

IBM and Tivoli are registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

All other company and product names mentioned may be trademarks of the respective companies with which they are associated.

Use, duplication, or disclosure by the government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of DFARS 52.227-7013 for the DOD and as set forth in FAR 52.227-19(a)-(d) for civilian agencies.

Sybase, Inc., One Sybase Drive, Dublin, CA 94568.

Contents

Getting Started	1
Preinstallation Checklist.....	1
Required Skills and Permissions	1
Preinstalled Software	1
Standard Deployment Model	3
Standard Installation.....	7
Installation Checklist.....	7
Create Master Application User (mob-aps-1 and mob-web-1)	8
Network Ports.....	8
Install Java Virtual Machine (mob-aps-1 and mob-web-1)	9
Installing Java Card Packages Required by On-Device Charging.....	9
Unpacking the Software (mob-aps-1)	9
Creating the Directory Structure and Copying Files	10
Install the Database (mob-db-1)	11
ASE Configuration.....	11
Create the Database Schema (mobr5).....	11
Run DBMaintain Against the Database Schema	13
Update the Default Configuration (mob-aps-1).....	14
Creating/Updating Hashed Password for the Universal User	15
Creating/Updating the Encrypted Password for Preferences.....	16
Configure Database Properties (mob-aps-1)	17
Security Keystores.....	17
Creating the Jetty Key.....	17
Creating the Internal Portal Tomcat Key.....	19
Creating the Apache HTTPD Key.....	21
Creating the Keystore for Data Encryption	22
Update Configuration Properties (mob-aps-1 and mob-web-1) ..	23
JMX Configuration.....	24
Install Preferences Software (mob-aps-1).....	24
Unpacking and Copying the Preferences JAR file	24
Install Third-Party Software (mob-aps-1)	25
Creating the JDBC Driver Bundle.....	25
Installing Spring Source	26

Getting Started

Install the Sybase Mobiliser Reporting Module	26
Configure the Apache HTTPD Server	27
Smartphone Mobiliser Mobile Internet Version	28
Mobiliser Platform Proxy	28
Brand Mobiliser	29
On-Device Charging Installation and Configuration	29
Provision Secure Element Keys for DIRECT Mode ...	30
Generate Private Keys Used by On-Device Charging	31
Start Mobiliser Platform	32
Starting the Server and User Interface	32
Starting Proxy	34
Starting Internal Tomcat	35
Starting Apache HTTPD	35
Configuring Preferences	35
Message Properties Configuration	39
Virus Protection	43
Configuring the Virus Scan Adapter for SAP	
NetWeaver	43
Clam AV	44
Validate the Installation	46
Default Web UI Accounts	46
Signing up as a Test Consumer	46
Stop Installation	48
Stopping Apache HTTPD	48
Stopping Internal/External Tomcat	48
Stopping Proxy	48
Stopping Server	48
Index	50

Getting Started

You can install Sybase® Mobiliser Platform 5.1 SP03 in both development and production environments. The Mobiliser Platform consists of three components:

- Sybase Money Mobiliser (Core)
- Sybase Smartphone Mobiliser
- Sybase Brand Mobiliser.

Preinstallation Checklist

Before you install Mobiliser Platform, read *Mobiliser Platform Supported Hardware and Software* on the Sybase Product Documentation Web site.

Required Skills and Permissions

Mobiliser Platform requires some IT technical skills for installation:

- Working knowledge of UNIX
- Working knowledge of Shell (bash)
- Ability to create new OS user accounts
- Good working knowledge with database of choice (Adaptive Server® Enterprise, DB2, or Oracle) and access to privileged database user to create new user/schema
- Basic understanding of HTTP and HTTPS protocol
- Basic understanding of SSL like certificate creation and validation chains

Preinstalled Software

A successful Mobiliser Platform installation requires certain software to be installed and configured on your server.

- Internet access
- Access to Sybase Product Download Center (SPDC) or SAP® Service Marketplace (SMP)
- Operating system (documented in *Mobiliser Platform Supported Hardware and Software*)
- Java (refer to *Mobiliser Platform Supported Hardware and Software* for the latest version) – SAP JVM is recommended. Also, JAVA_HOME and PATH environment variables must be set correctly
- Apache HTTPD (refer to *Mobiliser Platform Supported Hardware and Software* for the latest version) with proxy and SSL modules installed and enabled, or alternatively another Web and proxy server
- Firewall of choice
- Database of choice: ASE, DB2, or Oracle. For ASE, you must also specify page size.
- Create directory structure and copy files
- Install Mobiliser Platform database (mob-db-1)
- Open Source toolkit for SSL/TLS (openssl)

Getting Started

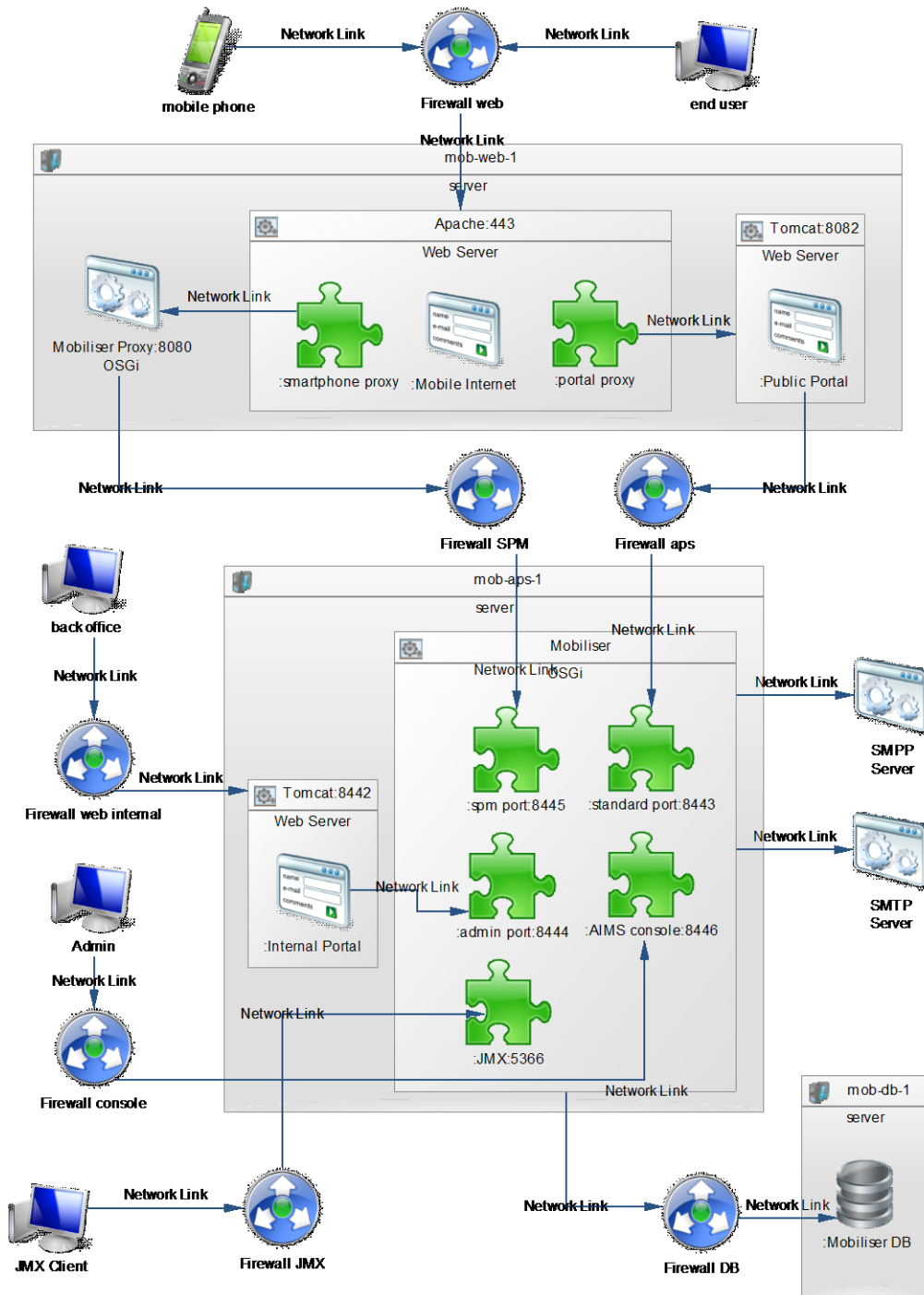
Standard Deployment Model

Each Mobiliser Platform host must meet the requirements for operating system and available disk space. In a development or test environment, you can install the system on a single physical host or virtual machine. In a production environment, deploy the system in a tiered manner to aid in administration, maintenance, and security.

The standard Mobiliser Platform tiered architecture contains:

- Web layer - customer self-service portal
- Application layer - Web service, back office
- Database layer

Standard Deployment Model



Standard Deployment Model

Note: The task steps in this document and the example configuration assume that you are using the exact host names as shown. If you need to make modifications, see the *Sybase Mobiliser Platform System Administration Guide* to learn where and how to change host names and port numbers.

Consider using the host names used in the figure. To do so, you may need to change the host names in your `/etc/hosts` file (requires root privileges).

Standard Deployment Model

Standard Installation

Set up the application directories and accounts that are used to operate Mobiliser Platform. Unless specified, directory structure, system accounts, and other such information are recommendations. Follow your organization's IT best practices, as well as local system and security policies.

Installation Checklist

Use this checklist to track the progress of Mobiliser Platform installation and configuration. Before you begin, verify that your system meets the minimum system requirements.

The installation is done on three different servers: a database server (mob-db-1), an application server (mob-aps-1), and a Web server (mob-web-1).

Task	Server	Completed
1. Create groups and users, see page 8	mob-aps-1 mob-web-1	<input type="checkbox"/>
2. Install SAP JVM (Optional), see page 9	mob-aps-1 mob-web-1	<input type="checkbox"/>
3. Unpack the Mobiliser Platform software, see page 9	mob-aps-1	<input type="checkbox"/>
4. Install Mobiliser Platform database, see page 11	mob-db-1	<input type="checkbox"/>
5. Create required Mobiliser Platform security hashes, see page 15	mob-aps-1	<input type="checkbox"/>
6. Configure Mobiliser Platform database properties, see page 17	mob-aps-1	<input type="checkbox"/>
7. Create security keystores, see page 17	mob-aps-1 mob-web-1	<input type="checkbox"/>
8. Configure Jetty Server, see page 17	mob-aps-1	<input type="checkbox"/>
9. Configure logging, see page 19 of the <i>Sybase Mobiliser Platform System Administration</i> guide	mob-aps-1	<input type="checkbox"/>
10. Update configuration properties, see page 23	mob-aps-1 mob-web-1	<input type="checkbox"/>
11. Verify the Web portal application, see page 32	mob-aps-1 mob-web-1	<input type="checkbox"/>
12. Install third-party software, see page 25	mob-aps-1	<input type="checkbox"/>
13. Configure the setevn.sh script, see EUSER2MOB on page 7 and INTTOMCAT2MOB on page 12 of the <i>Sybase Mobiliser Platform System Administration</i> guide	mob-web-1 mob-aps-1	<input type="checkbox"/>
14. Configure the proxy, see page 28	mob-web-1	<input type="checkbox"/>
15. Start Mobiliser Platform, see page 32	mob-aps-1 mob-web-1	<input type="checkbox"/>

Standard Installation

Task	Server	Completed
16. Validate the installation, see page 46	—	<input type="checkbox"/>

Create Master Application User (mob-aps-1 and mob-web-1)

To create the user account, use the appropriate host operating system command. Except for making changes to the master Apache HTTPD configuration file, this is the only installation task that must be executed by a privileged (root) user.

User Name	Description	Shell	SSH	Home	Host
sybase	Master application user	Bash	Y	/opt/sybase	mob-web-1 mob-aps-1

SAP recommends the following:

- For security reasons, use the sudo feature to restrict control and access of Mobiliser Platform application users.
- Application users do not have a valid shell; therefore, remove the shell from the sybase user after installation and use sudo to execute commands on behalf of the sybase user.

Network Ports

Configure your firewalls to allow communication between the different Mobiliser Platform nodes. Refer to the Standard Deployment Model diagram for an illustration on page 3.

This table describes the default port configuration of Mobiliser Platform. For more details on how to change the ports used by Mobiliser Platform, see the *Port and Host Name Configuration* section in the *Sybase Mobiliser Platform System Administration* guide.

Name	Source	Destination	Protocol
PROXY2MOB	mob-web-1:*	mob-aps-1:8445	HTTPS
EUSER2MOB	mob-web-1:*	mob-aps-1:8443	HTTPS
JMX2MOB	Internal_JMX:*	mob-aps-1:5366	HTTPS
ADMIN2MOB	Admin_WS:*	mob-aps-1:8446	HTTPS
BO2INTTOMCAT	Backoffice_WS:*	mob-aps-1:8442	HTTPS
MOB2DB	mob-aps-1:*	mob-db-1:<db_listener>	TCP
ALL2WEB	*	mob-web-1:443	HTTPS
WEB2PROXY	mob-web-1:*	mob-web-1:8080	HTTP
WEB2TOMCAT	mob-web-1:*	mob-web-1:8082	HTTP

Note: During installation, files are copied between servers using `scp` command. Therefore, access to port 22 from `mob-aps-1` to `mob-web-1` and `mob-db-1` is required. If port 22 in your installation is unavailable, use an alternative method for copying the files onto the target machines.

Install Java Virtual Machine (mob-aps-1 and mob-web-1)

(Optional) Mobiliser Platform runs on all JVMs that are compliant with J2SE 1.6 or J2SE 1.7 specifications. We recommend that you run Mobiliser Platform on SAP JVM, which can be downloaded from the SAP Marketplace at <https://service.sap.com/swdc>.

Installing Java Card Packages Required by On-Device Charging

1. Download the Java Card Development Kit 2.2.1.
http://www.oracle.com/technetwork/java/javasebusiness/downloads/java-archive-downloads-javame-419430.html#java_card_kit-2.2.1-oth-JPR
2. Unzip the archived file.
3. Set the `JC_HOME` environment variable to the home subfolder of the unzipped file.

Unpacking the Software (mob-aps-1)

The Mobiliser Platform software ZIP file contains everything you need to complete a successful installation. If you want to follow the Mobiliser Platform standard deployment model, you will need to copy some of the ZIP file subdirectories to various servers.

Execute this operation with the `sybase` user.

1. Log into the `mob-aps-1` server.
2. Unpack the software into:

```
/opt/sybase/mobiliser
```

These installation objects are created:

Object	File Path
Public Tomcat container	/applications/web_public
Internal Tomcat container	/applications/web_internal
Mobiliser Platform container	/applications/money
Sybase ASE script archives	/applications/ase/sql
IBM DB2 script archives	/applications/ibm/sql
IBM DB2 driver patch	/applications/ibm/patch/create_jdbc_bundle.sh /applications/ibm/patch/db2manifest
Oracle script archives	/applications/oracle/sql
Oracle driver patch	/applications/oracle/patch/create_jdbc_bundle.sh /applications/oracle/patch/oraclemanifest

Standard Installation

Object	File Path
Mobiliser Platform proxy	/applications/proxy
Apache HTTPD configuration	/applications/httpd
Smartphone Mobiliser Mobile Internet	/applications/mobileweb

Creating the Directory Structure and Copying Files

Execute all operations with the sybase user. All relevant files are automatically copied to the correct directories on the target machines.

Note: Replace <db> with the appropriate name of the database you are using (ASE, IBM, or Oracle).

1. Log into mob-aps-1.

2. Copy the following files to mob-web-1:

```
scp -r /opt/sybase/mobiliser/applications/web_public/ mob-web-1:/opt/sybase/portal
```

```
scp -r /opt/sybase/mobiliser/applications/proxy/ mob-web-1:/opt/sybase/proxy
```

```
scp -r /opt/sybase/mobiliser/applications/httpd/ mob-web-1:/opt/sybase/httpd
```

```
scp -r /opt/sybase/mobiliser/applications/mobileweb/ mob-web-1:/opt/sybase/mobileweb
```

3. Copy the following files to mob-aps-1:

```
cp -r /opt/sybase/mobiliser/applications/web_internal/ /opt/sybase/portal
```

```
cp -r /opt/sybase/mobiliser/applications/money/ /opt/sybase/
```

4. Rename system property file to configure Mobiliser Platform for correct database:

```
cd /opt/sybase/money/conf
```

```
cp system-<db>.properties system.properties
```

Note: Replace <db> with the appropriate database name.

5. Log into mob-db-1.

6. Create directories:

```
mkdir -p /opt/sybase/db/
```

7. Log into mob-aps-1.

8. Copy the following files to mob-db-1:

```
scp -r /opt/sybase/mobiliser/applications/<db>/sql/ mob-db-1:/opt/sybase/db/sql
```

Install the Database (mob-db-1)

In the standard deployment model, Mobiliser Platform database components are installed on the mob-db-1 server. The required files, which were copied to the appropriate location in the previous task are located in:

```
/opt/sybase/db/sql
```

ASE Configuration

When you create the database in an ASE 15.7 installation, make sure that the page size is 8K instead of the default value of 4K.

Note: If you specify a page size other than 8K, you cannot create composite indexes greater than 1250 bytes, which results in an incomplete installation of Mobiliser Platform 5.1 the database scripts and renders the entire Mobiliser Platform 5.1 installation invalid.

The functionality group changes enables the:

- Permissive Unicode for the database character set
- Quoted identifier enhancements
- Select for update syntax when performing database queries and updates
- Streamlined dynamic SQL (useful for internal QP optimizations)
- Inline default sharing for handling large numbers of defaults

Create the Database Schema (mobr5)

The Mobiliser Platform database schema contains all the data and metadata. Manually create this schema by executing:

```
001_MONEY_drop_and_create_user.DDL
```

This script also creates the roles and privileges that are required for the Mobiliser Platform database. By default, the schema and user are both named “mobr5,” you can change if necessary. Refer to

Standard Installation

Configure Database Properties (mob-aps-1) on page 17 to learn what else to change in the Mobiliser Platform configuration if you are not using the default database user.

For assistance in executing a DDL script against an installed database, see your database platform documentation.

Run DBMaintain Against the Database Schema

After you create the Mobiliser Platform schema, install the data and metadata into the schema by executing a series of scripts using the packaged Java tool DBMaintain.

DBMaintain is an executable JAR file that contains the script archive (DDL scripts) as well as the Java classes that are required to execute the scripts. You must provide the JDBC driver location in the DBMaintain configuration file. JDBC drivers for ASE databases are included with the Mobiliser Platform software; however, you must download the JDBC drivers for Oracle and DB2 databases from the respective Web site, or determine whether they already exist in the database installation directory.

To execute the packaged DBmaintain scriptarchive file, located in the `./sql` directory on `mob-db-1`, you need the script archive JAR files that are in the `./sql` directory.

The script archives are packaged as JAR files with the following names:

Database	JAR File Name
ASE	<code>com.sybase365.mobiliser.vanilla.standalone-5.1.3.RELEASE-scriptarchive-ase.jar</code> <code>com.sybase365.mobiliser.vanilla.standalone-5.1.3.RELEASE-scriptarchive-ase-vanilla.jar</code>
DB2	<code>com.sybase365.mobiliser.vanilla.standalone-5.1.3.RELEASE-scriptarchive-db2-driverless.jar</code> <code>com.sybase365.mobiliser.vanilla.standalone-5.1.3.RELEASE-scriptarchive-db2 -vanilla-driverless.jar</code>
Oracle	<code>com.sybase365.mobiliser.vanilla.standalone-5.1.3.RELEASE-scriptarchive-oracle-driverless.jar</code> <code>com.sybase365.mobiliser.vanilla.standalone-5.1.3.RELEASE-scriptarchive-oracle-vanilla-driverless.jar</code>

Note: The version names and numbers in the file names might be slightly different in your installation.

Security settings that are managed via the database and preference settings do not require a restart of the container to take effect.

- `dbmaintain.properties.<db_choice>` configuration file included in the `./sql` directory
- Database JDBC driver for database (if installing on Oracle or DB2)

Standard Installation

Running DBMaintain

This task includes only the steps that are required to run DBMaintain for the installation process. See the *DBMaintain Guide* for a complete description of the DBmaintain properties file and the settings, as well as the supported command line parameters.

3. Open:
`dbmaintain.properties.<db_choice>`
4. Change the `database.url` value to reflect the TCP port on which your database listener is running.
5. In the `database.driverLocation` field, enter the path to the downloaded JDBC driver.
6. Save the changes to the `dbmaintain.properties.<db_choice>` file.

Note: If you altered the original schema creation script by changing the password information for the `mobr5` schema (or any other name), you must also update the `database.password` in the `dbmaintain.properties.<db_choice>` file.

7. Execute the scriptarchives:
 - a. `java -jar com.sybase365.mobiliser.vanilla.standalone-<version>-scriptarchive-<db_choice>-driverless.jar -c dbmaintain.properties.<db_choice>`
 - b. `java -jar com.sybase365.mobiliser.vanilla.standalone-<version>-scriptarchive-<db_choice>-vanilla-driverless.jar -c dbmaintain.properties.<db_choice>`

The Mobiliser Platform database has been successfully installed on server `mob-db-1`.

Update the Default Configuration (mob-aps-1)

The software is installed with a blank password for the system user. You must set a new password for this user and configure the password in the configuration accordingly. For security reasons, there are several pre-configured values that must be changed for a fresh installation.

Additionally, for the Mobiliser Platform container to function properly, you must configure the universal Mobiliser Platform user (customer ID 100) within the database with the required hashed credential. The Web portal uses the Mobiliser Platform user to authenticate the Mobiliser Platform container. Therefore, you must set the password in hashed format in the database and in encrypted format in the portal configuration.

The hash for the Mobiliser Platform user is placed within the `MOB_CUSTOMERS_CREDENTIALS` table in the database. The encrypted password hash for preferences is used by the portal, and that encrypted value is placed in the `MOB_PREFERENCES` table in the database. The first hash is made from any plain text password, while the second (encrypted) hash is built off of the value chosen for the first hash. Both hashes have specific places in the database.

By default, the Mobiliser Platform and most Java Development Kits (JDK), support AES encryption up to 128 bit key length. If you require to use stronger encryption you must update the security policy jars within the JVM running on mob-web-1 and mob-aps-1:

1. Download the Java Cryptography Extension (JCE) unlimited strength jurisdiction policy file from your JDK vendor.

For Oracle and IBM JDKs, two files are provided:

- local_policy.jar
- US_export_policy.jar

2. Replace these files in your JDK installation directory at:

```
/jre/lib/security
```

3. Refer to the accompanying installation instructions for Java virtual machine (JVM) specific hints.

4. Create the hashed formats using a tool that is packaged with Mobiliser Platform and located in the /opt/sybase/money/tools directory:

```
com.sybase365.mobiliser.vanilla.cli-tools-<version>-  
CLIPasswordEncoderClient.jar
```

The configuration entries in the database require encrypted entries.

5. Create the encrypted values using another tool that is packaged with Mobiliser Platform in the same directory as CLIPasswordEncoderClient:

```
com.sybase365.mobiliser.vanilla.cli-tools-<version>-  
CLIEncrypterClient.jar
```

See the *Configuration* section of the *Sybase Mobiliser Platform System Administration* guide for details about how to enable strong encryption.

Creating/Updating Hashed Password for the Universal User

Note: The following SQL script is for ASE databases. Use the correct SQL syntax for DB2 and Oracle databases.

1. Execute:

```
java -jar com.sybase365.mobiliser.vanilla.cli-tools-  
<version>-CLIPasswordEncoderClient.jar
```

2. Select the hashing method.

3. Enter the plain text password and then the salt, which is the same as the customer ID. Depending on the hashing method, a salt may not be required.

4. Update the hash value in the database by running the following statement on the mob-db-1 database :

```
UPDATE "MOB_CUSTOMERS_CREDENTIALS" SET STR_CREDENTIAL =  
'<Hash Value>' WHERE ID_CUSTOMER = 100
```

Standard Installation

- Update the creation date for both the universal Mobiliser Platform user (customer ID 100) and sysmgr user (customer ID 106) by the appropriate statements on the mob-db-1 database:

Database	States
ASE	UPDATE MOB_CUSTOMERS_CREDENTIALS SET DAT_CREATION = GETDATE() WHERE ID_CUSTOMER IN (100,106)
DB2	UPDATE MOB_CUSTOMERS_CREDENTIALS SET DAT_CREATION = CURRENT_TIMESTAMP WHERE ID_CUSTOMER IN (100,106)
Oracle	UPDATE MOB_CUSTOMERS_CREDENTIALS SET DAT_CREATION = SYSDATE WHERE ID_CUSTOMER IN (100,106)

Creating/Updating the Encrypted Password for Preferences

- Execute:

```
java -jar com.sybase365.mobiliser.vanilla.cli-tools-  
<version>-CLIEncrypterClient.jar <key> <value>
```
- Configure <key> in /opt/sybase/portal/conf/context.xml as the value for the environment element with name “prefs/secret”. The default value is “paybox”.

where <key> represents the decryption key that is used by the Web portals to decrypt data coming from the Preferences service.

Warning: Do not choose a key at random! The key you enter must be identical to the one used by the Web portals to decrypt the data from Preferences; otherwise, the portals cannot connect to Mobiliser Platform.

<value> represents the clear text password used when the creating the hashed password for the universal Mobiliser Platform user in the previous section.

- Once you have successfully created the encrypted value, update the mob-db-1 database with the new preferences:

```
UPDATE MOB_PREFERENCES  
SET STR_VALUE = '{AES-128-PBKDF2}<Hash Value>' WHERE STR_NAME  
= 'mobiliser.password' AND  
( STR_PATH =  
'/presentationlayer/system/com/sybase365/mobiliser/web/util/D  
ynamicServiceConfiguration/' OR STR_PATH =  
'/presentationlayer/system/com/sybase365/mobiliser/util/tools  
/wicketutils/services/Configuration/')
```

The Mobiliser Platform database, located on the mob-db-1 server, is now completely set up with required data and hashes; you need not access it for the remainder of the Mobiliser Platform installation.

Configure Database Properties (mob-aps-1)

If you are using the default configuration and database schema and password, the Mobiliser Platform container is correctly configured. If you need to make any changes to the database configuration, you can modify the values in the configuration files.

You can change the host name, database instance name, and port (JDBC URL) in:

```
/opt/sybase/money/conf/system.properties (url, port)
```

You can change the user name and password in:

- /opt/sybase/money/conf/cfgbackup/com.sybase365.mobiliser.framework.persistence.jdbc.bonecp.pool.properties
- /opt/sybase/money/conf/cfgbackup/com.sybase365.mobiliser.util.report.crystalreports.properties

See the *Configuration* section in *Sybase Mobiliser Platform System Administration* to learn how to encrypt passwords (and other configuration data) and how to use system properties in configuration files.

Other parameters might influence system performance; see *Performance Considerations* section in *Sybase Mobiliser Platform System Administration*.

Security Keystores

Mobiliser Platform uses keys to secure the communication between hosts (HTTPS) and to encrypt sensitive information (for example, credit card data). By default, for security, Mobiliser Platform applications do not contain any keys; you must create them as part of the overall installation process.

Creating the Jetty Key

The Mobiliser Platform container Jetty key secures HTTP communication to the Mobiliser Platform container. It also secures the JMX port of Mobiliser Platform, which can be used for remote monitoring and administration. Therefore, you must configure the password to access the keystore at two different locations.

Execute all operations as the sybase user and be sure to note all passwords for later reference.

1. Log into mob-aps-1.
2. Create the following directory, if necessary, then change to the directory you just created:

```
mkdir -p /opt/sybase/money/conf/keys/server
cd /opt/sybase/money/conf/keys/server
```

3. Generate a new keystore and key (use identical passwords for the keystore and the key itself):

```
keytool -genkey -v -keystore keystore -alias money -keyalg
RSA -keysize 2048 -validity 8000
```

The keytool generates output similar to that shown below. Enter the appropriate values:

- c. Enter keystore password: **CHANGEME**
- d. Re-enter new password: **CHANGEME**

Standard Installation

- e. What is your first and last name?
[Unknown]: mob-aps-1
- f. What is the name of your organizational unit?
[Unknown]: Sybase 365
What is the name of your organization?
[Unknown]: SAP
- g. What is the name of your City or Locality?
[Unknown]: Reston
- h. What is the name of your State or Province?
[Unknown]: Virginia
- i. What is the two-letter country code for this unit?
[Unknown]: US
- j. Is CN=mob-aps-1, OU=Sybase 365, O=SAP, L=Reston,
ST=Virginia, C=US correct?
[no]: yes
- Generating 2,048 bit RSA key pair and self-signed certificate
(SHA1withRSA) with a validity of 9,999 days for: CN=mob-aps-
1, OU=Sybase 365, O=SAP, L=Reston, ST=Virginia, C=US
Enter key password for <money>
(RETURN if same as keystore password):
4. Change the keystore file permissions:
chmod 0600 /opt/sybase/money/conf/keys/server/keystore
5. Export this certificate to /tmp/money.cert, providing the password entered in step 3:
keytool -exportcert -v -keystore keystore -alias money -file
/tmp/money.cert
6. Import this certificate into the Java truststore (cacerts)
keytool -importcert -v -keystore
\$JAVA_HOME/jre/lib/security/cacerts -alias money -file
/tmp/money.cert
- The keytool generates output similar to that shown below. Enter the appropriate values:
Enter keystore password: **changeit** (this is the default Java
keystore password)
Owner: CN=mob-aps-1, OU=Sybase 365, O=SAP, L=Reston,
ST=Virginia, C=US
Issuer: CN=mob-aps-1, OU=Sybase 365, O=SAP, L=Reston,
ST=Virginia, C=US
Serial number: 5058f3d1
Valid from: Wed Sep 19 00:21:05 CEST 2012 until: Fri Feb 03
23:21:05 CET 2040
Certificate fingerprints:
MD5:
B9:1F:A7:1C:85:32:41:60:66:93:BB:F2:01:5D:1E:E7
SHA1:
AA:96:0F:F5:3C:81:AB:4D:DA:10:B3:96:4E:31:2E:C7:55:D0:D1:89
Signature algorithm name: SHA1withRSA
Version: 3
Trust this certificate? [no]: yes

- Certificate was added to keystore
[Storing /opt/sybase/java/current/jre/lib/security/cacerts]
7. Configure the keystore password in:
/opt/sybase/money/conf/system.properties
 8. Use a text editor, such as vi, to open the file:
/opt/sybase/money/conf/system.properties
 9. Locate the line shown below, and replace the highlighted part with the keystore password:
javax.net.ssl.keyStorePassword=**CHANGEME**
 10. Obfuscate the password that was used to create the keystore (<password> is password you entered in step 3):
java -cp /opt/sybase/money/bundles/06-http/pax-web-jetty-bundle-*.jar org.eclipse.jetty.util.security.Password
CHANGEME
 11. Use a text editor, such as vi, to open jetty.xml:
/opt/sybase/money/conf/jetty.xml
 12. Locate the following lines and replace the highlighted section with the output of step 10:
<Set name="KeyStorePassword">**OBF:1z071u9d1wgm1wfc1ua51z0n**</Set>
<Set name="KeyManagerPassword">**OBF:1z071u9d1wgm1wfc1ua51z0n**</Set>
<Set name="TrustStorePassword">**OBF:1z071u9d1wgm1wfc1ua51z0n**</Set>
 13. Log into mob-web-1 and create the following directory
mkdir -p /opt/sybase/keys/
 14. Log into mob-aps-1 and copy the certificate to mob-web-1:
scp /tmp/money.cert mob-web-1:/opt/sybase/keys/
 15. Log into mob-web-1.
 16. Repeat step 6 on mob-web-1, except use this certificate file:
keytool -importcert -v -keystore
\$JAVA_HOME/jre/lib/security/cacerts -alias money -file
/opt/sybase/keys/money.cert

Creating the Internal Portal Tomcat Key

The internal portal Tomcat key secures HTTP communication to the internal portal (Tomcat on mob-aps-1).

Execute all operations as the sybase user and be sure to note all passwords for later reference.

1. Log into mob-aps-1.
2. Create the following directory, if necessary, then change to the directory you just created:
mkdir -p /opt/sybase/portal/conf/keys/server
cd /opt/sybase/portal/conf/keys/server
3. Generate a new keystore and key (**use identical passwords for the keystore and the key itself**):
keytool -genkey -v -keystore keystore -alias tomcat -keyalg
RSA -keysize 2048 -validity 8000

Standard Installation

The keytool generates output similar to that shown below. Enter the appropriate values:

- a. Enter keystore password: **CHANGEME**
- b. Re-enter new password: **CHANGEME**
- c. What is your first and last name?
[Unknown]: mob-aps-1
- d. What is the name of your organizational unit?
[Unknown]: Sybase 365
- e. What is the name of your organization?
[Unknown]: SAP
- f. What is the name of your City or Locality?
[Unknown]: Reston
- g. What is the name of your State or Province?
[Unknown]: Virginia
- h. What is the two-letter country code for this unit?
[Unknown]: US
- i. Is CN=mob-aps-1, OU=Sybase 365, O=SAP, L=Reston, ST=Virginia, C=US correct?
[no]: yes

```
Generating 2,048 bit RSA key pair and self-signed certificate
(SHA1withRSA) with a validity of 9,999 days for: CN=mob-aps-
1, OU=Sybase 365, O=SAP, L=Reston, ST=Virginia, C=US
Enter key password for <tomcat> (RETURN if same as keystore
password):
```

```
Re-enter new password:
```

```
[Storing /opt/sybase/money/conf/keys/server/keystore]
```

4. Change the keystore file permissions:

```
chmod 0600 /opt/sybase/portal/conf/keys/server/keystore
```

5. Export this certificate to /tmp/portal.cert, providing the password entered in step 3:

```
keytool -exportcert -v -keystore keystore -alias tomcat -file
/tmp/portal.cert
```

6. Configure the keystore password in the portal (tomcat).

Use a text editor, such as vi, to open:

```
/opt/sybase/portal/conf/server.xml
```

7. For the keystorePass value, enter the password you entered in step 3:

```
<Connector
    port="8442" scheme="https"
    secure="true" SSLEnabled="true"
    keystorePass="CHANGEME"
    clientAuth="false" ...
```


Creating the Apache HTTPD Key

Execute all operations as the sybase user and be sure to note all passwords for later reference.

1. Log into mob-web-1.
2. Create the following directory, if necessary, then change to the directory you just created:

```
mkdir -p /opt/sybase/httpd/certs
mkdir -p /opt/sybase/httpd/keys
```

3. Create a server key:

```
openssl genrsa -out /opt/sybase/httpd/keys/server.key 2048
```

4. Create a certification request, making sure that you enter the correct host name when asked:

```
openssl req -new -key /opt/sybase/httpd/keys/server.key -out
/opt/sybase/httpd/certs/server.csr
```

You see output similar to:

```
Country Name (2 letter code) [XX]:US
State or Province Name (full name) []:Virginia
Locality Name (eg, city) [Default City]:Reston
Organization Name (eg, company) [Default Company Ltd]:Sybase
365
Organizational Unit Name (eg, section) []:SAP
Common Name (eg, your name or your server's hostname)
[]:www.example.com
Email Address []: Please enter the following 'extra'
attributes to be sent with your certificate request:
  A challenge password []:
  An optional company name []:
```

5. Send the CSR to the CA for signing and copy the certificate to:

```
/opt/sybase/httpd/certs/server.crt
```

Alternatively, you can sign the created CSR yourself, which generates a warning in the browser:

```
openssl x509 -req -days 8000 -in
/opt/sybase/httpd/certs/server.csr -signkey
/opt/sybase/httpd/keys/server.key -out
/opt/sybase/httpd/certs/server.crt
```

6. Change the file owner of server.crt to the user that is used by the Apache process.

This protects the key from unauthorized access.

7. Change the file permissions to be restricted:

```
chmod 0600 /opt/sybase/httpd/keys/server.key
```

8. Execute this command as a user with **root privileges** because <apache user> corresponds to the OS user that launches the Apache HTTPD server, for example, “apache”:

```
chown <apache user> /opt/sybase/httpd/keys/server.key
```

Creating the Keystore for Data Encryption

For credit card payments, the default Mobiliser Platform installation uses asymmetric encryption to secure credit card and bank account information in the front-end and a dummy payment handler implementation in the back-end to decrypt credit card payments.

Execute all operations as the sybase user and be sure to note all passwords for later reference.

1. Log into mob-aps-1.

2. Change the following directory, if necessary, then change to the directory you just created:

```
mkdir -p /opt/sybase/money/conf/keys
cd /opt/sybase/money/conf/keys
```

3. Generate a new keystore and key (use identical passwords for the keystore and the key itself), and modify the `dname` parameters as required:

```
keytool -genkey -validity 7305 -keystore mobiliser.jks -alias
mobiliser_card -keysize 2048 -storepass changeit -keypass
changeit -keyalg RSA -dname "CN=Mobiliser Platform,
OU=System, O=Sybase, L=Raunheim, S=Hessen, C=DE"
```

4. Export the public key. You must enter the store password from step 3:

```
keytool -export -alias mobiliser_card -file
mobiliser_card.crt -keystore mobiliser.jks
```

5. Import the certificate into a new keystore. Change the keystore password (use a different one than above):

```
keytool -import -alias mobiliser_card -file
mobiliser_card.crt -keystore mobiliser_pub.jks -storepass
changeit
```

6. Generate a new key in the same keystore entered in step 3. You can use the same keystore password, but you can select a different key password.

```
keytool -genkey -validity 7305 -keystore mobiliser.jks -alias
mobiliser_bank -keysize 2048 -storepass changeit -keypass
changeit -keyalg RSA -dname "CN=Mobiliser Platform,
OU=System, O=Sybase, L=Raunheim, S=Hessen, C=DE"
```

7. Export the public key. You must enter the keystore password from step 3:

```
keytool -export -alias mobiliser_bank -file
mobiliser_bank.crt -keystore mobiliser.jks
```

8. Import the certificate into the keystore entered in step 5 using the same key store password you entered in step 5:

```
keytool -import -alias mobiliser_bank -file
mobiliser_bank.crt -keystore mobiliser_pub.jks -storepass
changeit
```

9. Generate a new key into the same keystore entered in step 3. You can use the same keystore password, but you can select a different key password.

```
keytool -genkey -validity 7305 -keystore mobiliser.jks -alias
mobiliser_odc_se_ks -keysize 2048 -storepass changeit -
keypass changeit -keyalg RSA -dname "CN=Mobiliser Platform,
OU=System, O=Sybase, L=Raunheim, S=Hessen, C=DE"
```

10. Export the public key, providing the password entered in step 3:


```
keytool -export -alias mobiliser_odc_se_ks -file
mobiliser_odc.crt -keystore mobiliser.jks
```
11. Import the certificate into the keystore generated in step 5, providing the same key store password you entered in step 5:


```
keytool -import -alias mobiliser_odc_se_ks -file
mobiliser_odc.crt -keystore mobiliser_pub.jks -storepass
changeit
```
12. Generate another key into the same keystore created in step 3. You can use the same keystore password, but you can select a different key.


```
keytool -genkey -validity 7305 -keystore mobiliser.jks -alias
mobiliser_odc_signing -keysize 1024 -storepass changeit -
keypass changeit -keyalg RSA -dname "CN=Mobiliser Platform,
OU=System, O=Sybase, L=Raunheim, S=Hessen, C=DE"
```
13. Export the public key, providing the password entered in step 3:


```
keytool -export -alias mobiliser_odc_signing -file
mobiliser_odc.crt -keystore mobiliser.jks
```
14. Import the certificate into the keystore created in step 5, providing the same keystore password entered in step 5:


```
keytool -import -alias mobiliser_odc_signing -file
mobiliser_odc.crt -keystore mobiliser_pub.jks -storepass
changeit
```
15. Change the access privileges for the keystore that contains the keys:


```
chmod 0600 mobiliser.jks
```

The public keys are loaded from the Web portals via a Web service call.

You must use the Mobiliser Platform Operations Dashboard portal to configure passwords.

Update Configuration Properties (mob-aps-1 and mob-web-1)

Certain configuration files, such as keys that are used for encryption, contain sensitive information. You cannot monitor or control access to those files from the Mobiliser Platform application; they are controlled only by the system administrator. The relevant files and directories are:

- /opt/sybase/money/conf/ (Mobiliser Platform Container on mob-aps1)
- /opt/sybase/portal/conf/ (Web_UI Tomcat Container on mob-web-1)

Note: Access is limited to those users who run the respective server and all read/write access should be logged.

The user who starts these servers (user sybase) does not require any elevated privileges (for example, super user or sudoers).

Standard Installation

JMX Configuration

You can remotely manage Mobiliser Platform server using JMX, which is configured in this file:

```
/opt/sybase/money/conf/cfgbackup/com.sybase365.mobiliser.framework.gateway.security.authentication.jmx.properties
```

Check the parameters for:

- `jmxPort` – this is the port used to connect. The default is 5366.
- `serviceUrl` – if you change the port number, also update the service URL to match that port number.

For Example:

```
service:jmx:rmi://host1:5366/jndi/rmi://host1:5366/jmxrmi
```

If, when connecting remotely, you use a different host name, or only an IP address, you must also set a system property:

```
opt/sybase/money/conf/system.properties
```

Then set a value for the property:

```
java.rmi.server.hostname=externalhost1
```

Authentication is via the standard Mobiliser authentication mechanism; the user requires the privilege `JMX_ACCESS`. The user `sysmgr` is preconfigured with this privilege.

The JMX configuration utilizes the Jetty keystore that you created to enable SSL for JMX in *Creating the Jetty Key* on page 17. You must set another system property:

```
opt/sybase/money/conf/system.properties
```

Then set a value for this property:

```
javax.net.ssl.keyStorePassword=<jetty keystore password>
```

To disable remote JMX access, move the bundle `/opt/sybase/money/bundles/11-mobiliser-framework/com.sybase365.mobiliser.framework.gateway.security.authentication.jmx*-SNAPSHOT.jar` to `/opt/sybase/money/bundles/99-disabled`.

You can still access MBeans directly from the machine, if necessary, by using the process ID.

Install Preferences Software (mob-aps-1)

The Mobiliser container **requires** a preferences jar file that is available in the original Mobiliser 5.1.0 RELEASE package.

Note: The original Mobiliser 5.1.0 RELEASE needs to be downloaded from Sybase Product Download Center (SPDC) or SAP Marketplace (SMP). Please speak to the necessary representative to see if you have access to the repositories and/or software.

Unpacking and Copying the Preferences JAR file

1. After completely unpacking the Mobiliser 5.1.0 RELEASE zip file, navigate to:

```
/applications/<db_backend_of_choice>
```

- Unpack the corresponding Mobiliser container zip file, for example:

```
com.sybase365.mobiliser.vanilla.ase-5.1.0.RELEASE-dist.zip
```

- Navigate to the bundles section of the unpacked Mobiliser container:

```
cd com.sybase365.mobiliser.vanilla.ase-5.1.0.RELEASE/bundles/  
12-mobiliser-prefs
```

- Locate the following file:

```
com.sybase365.mobiliser.util.prefs.api-5.1.0.RELEASE.jar
```

Note: If the original installed Mobiliser release is after Mobiliser 5.1.0.RELEASE, the specific name of the jar file may be slightly different. The version number (5.1.0.RELEASE) will be different in the jar file name

- Copy `com.sybase365.mobiliser.util.prefs.api-5.1.0.RELEASE.jar` to `/opt/sybase/money/bundles/12-mobiliser-prefs`:

```
cp com.sybase365.mobiliser.util.prefs.api-5.1.0.RELEASE.jar  
/opt/sybase/money/bundles/12-mobiliser-prefs
```

Install Third-Party Software (mob-aps-1)

There are several third-party JAR files that are required for normal operation. Obtain this software from the respective vendors and deploy it directly into the Mobiliser OSGi container.

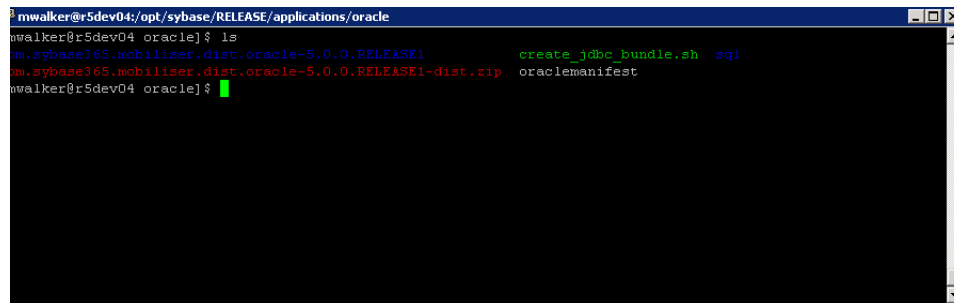
Creating the JDBC Driver Bundle

Note: This step needs to only be performed if the database choice for your installation is Oracle or DB2. The ASE JDBC driver is included in the original downloaded Mobiliser Platform package.

- Log into `mob-aps-1` as the `sybase` user, then navigate to:

```
/opt/sybase/mobiliser/applications/<db>/patch
```

where `<db>` is either Oracle or IBM.



```
mwalker@r5dev04:/opt/sybase/RELEASE/applications/oracle$ ls  
m.sybase365.mobiliser.dist.oracle-5.0.0.RELEASE1      create_jdbc_bundle.sh  sql  
m.sybase365.mobiliser.dist.oracle-5.0.0.RELEASE1-dist.zip  oraclemanifest  
mwalker@r5dev04 oracle]$
```

- Download either the Oracle or DB2 JDBC driver that is compatible with the JRE that was installed onto your system:

- Oracle: <http://www.oracle.com> (for example, `ojdbc6.jar`)
- DB2: <http://www.ibm.com> (for example, `db2jcc4.jar`)

Standard Installation

The JDBC JAR from the database provider must be packaged in an OSGi bundle. Without the JDBC JAR file, you will not be able to create the required JAR used by Mobiliser Platform to connect to and utilize the Oracle or DB2 database.

3. Run the `./create_jdbc_bundle.sh` using the appropriate manifest file (for example, `Oraclemanifest`) and JDBC jar as input variables, for example:
 - Oracle: `./create_jdbc_bundle.sh oraclemanifest ojdbc6.jar`
 - DB2: `./create_jdbc_bundle.sh db2manifest db2jcc4.jar`
4. Rename the newly created JAR file:
 - Oracle: `bundle_<name of jdbc>` to `oracle-jdbc-osgi_11.2.0.2.0-1.0.1.jar`
 - DB2: `com.sybase365.com.ibm.db2jcc4-9.7.4.jar` for DB2
5. Copy `oracle-jdbc-osgi_11.2.0.2.0-1.0.1.jar` or `com.sybase365.com.ibm.db2jcc4-9.7.4.jar` to `/opt/sybase/money/bundles/07-frameworks`:

```
cp oracle-jdbc-osgi_11.2.0.2.0-1.0.1.jar
/opt/sybase/money/bundles/07-frameworks
```

Installing Spring Source

1. Open a browser and navigate to the Spring Source software repository, located at: <http://ebr.springsource.com/repository/app/bundle/version/detail?name=com.springsource.org.jgroups&version=2.2.8>
2. Download and copy `com.springsource.org.jgroups-2.2.8.jar` into:
`/opt/sybase/money/bundles/07-frameworks`
3. Download and copy these JAR files into `/opt/sybase/money/bundles/16-framework-reports`:
 - `com.springsource.javax.media.jai.codec-1.1.3.jar`
(<http://ebr.springsource.com/repository/app/bundle/version/detail?name=com.springsource.javax.media.jai.codec&version=1.1.3>)
 - `com.springsource.javax.media.jai.core-1.1.3.jar`
(<http://ebr.springsource.com/repository/app/bundle/version/detail?name=com.springsource.javax.media.jai.core&version=1.1.3>)

Install the Sybase Mobiliser Reporting Module

The Sybase Mobiliser Reporting Module is a separate module that is licensed separately. It provides reporting capabilities to Mobiliser Platform, and leverages SAP Crystal Reports.

You can download Reporting Mobiliser from SAP Service Marketplace. The download package consists of various software bundles that must be placed in the appropriate `/opt/sybase/money/bundles` directory.

If you unpack the downloaded ZIP file into `/opt/sybase/money/`, all files are extracted into the correct location. If you have unpacked to a different location, manually copy the files into the correct location.

The unloaded or unpacked files and locations should be (the version numbers at the end of the file names might be higher or later than indicated here):

`/opt/sybase/money/bundles/17-crystalreports` directory:

- `com.businessobjects.cvom_12.2.212.1346-1.0.1.jar`
- `com.businessobjects.foundation.logging_12.2.212.1346-1.0.1.jar`
- `com.businessobjects.reports.jdbinterface_12.2.212.1346-1.0.1.jar`
- `com.businessobjects.visualization.pfjgraphics_12.2.212.1346-1.0.1.jar`
- `com.crystaldecisions.common.keycode_12.2.212.1346-1.0.1.jar`
- `com.crystaldecisions.reports.runtime_12.2.212.1346-1.0.1.jar`

`/opt/sybase/money/bundles/18-report-fragments` directory:

`com.azalea.ufl.barcode_1.0-1.0.1.jar`

`/opt/sybase/money/bundles/20-mobiliser-reports-services` directory:

- `com.sybase365.mobiliser.util.report.crystalreports.impl-5.1.3.RELEASE.jar`
- `com.sybase365.mobiliser.util.report.crystalreports.util-5.1.3.RELEASE.jar`
- `com.sybase365.mobiliser.util.report.crystalreports.web-5.1.3.RELEASE.war`
- `com.sybase365.mobiliser.util.report.watcher-5.1.3.RELEASE.jar`

Configure the Apache HTTPD Server

The required configuration files for the Apache HTTPD server should be located in `/opt/sybase/httpd/conf` on the `mob-web-1` server. The required SSL key should be in the preconfigured folder.

The configuration files:

- Should enable SSL on port 443 with the formerly created certificate,
- Include a reverse proxy configuration that sends all requests to the `/portal` URL to the Tomcat server,
- Include a reverse proxy configuration that sends all requests to `/mobiliser/smartphone`, `/mobiliser/rest/smartphone`, `/mobiliser/binary`, and `/mobiliser/rest/binary` to the Mobiliser Platform proxy,
- Should create an alias for `/mobileweb` and use `/opt/sybase/mobileweb` as the document root for this alias.

Verify that the proxy and SSL modules are available and enabled in the Apache installation.

Standard Installation

The only configuration required is to include the default Apache configuration file, which is included with Mobiliser Platform, into your Apache master configuration.

This master configuration file location varies by Apache packaging and OS. The examples that follow use files named:

```
/etc/httpd/conf/httpd.conf
/etc/apache2/apache2.conf
```

Before continuing, verify that there is no existing default configuration (VirtualHost) for port 443.

To configure the Apache HTTPD server:

1. Log into mob-web-1 using root user or owner the Apache configuration file.
2. Locate the Apache master configuration file and open it using a file editor.
3. Verify that this line is not part of an existing <VirtualHost> element, then add it to the configuration file:

```
include /opt/sybase/httpd/conf/mobiliser_httpd_ssl.conf
```

The standard configuration files that are supplied are preconfigured for Apache 2.2. If you are using Apache 2.4, open the files /opt/sybase/httpd/conf/mobiliser_httpd_ssl.conf and /opt/sybase/httpd/conf/mobiliser_httpd_locations.conf and modify the sections that are clearly marked for version 2.4.

Smartphone Mobiliser Mobile Internet Version

Smartphone Mobiliser comes in two different formats:

- Distribution for Android, Apple iPhone, and BlackBerry devices. The distribution requires compilation of source code, cryptographic signing of the deployment package, and distribution via the appropriate App store. This process is not included in this document.
- Mobile Internet version, which is a set of HTML, CSS, JavaScript, and image files. These files are located on mob-web-1 in the folder /opt/sybase/mobileweb and are served by the Apache HTTPD server.

The provided Apache configuration files already contain the entire required configuration and do not need to be modified.

Mobiliser Platform Proxy

The validating proxy is a specially assembled OSGi container that contains a subset of the Mobiliser Platform Core bundles.

Mobiliser Platform Proxy, which is installed on mob-web-1 in the folder /opt/sybase/proxy checks whether incoming requests comply with the contract that has been defined between client and server.

The Mobiliser Platform Proxy contains the same Jetty-specific configuration options as the Mobiliser Platform Core container, which is documented in the *Sybase Mobiliser Platform System Administration Guide*. In addition, there is a configuration file that contains the URL for the Mobiliser Platform Core to which the requests are forwarded (after successful validation).

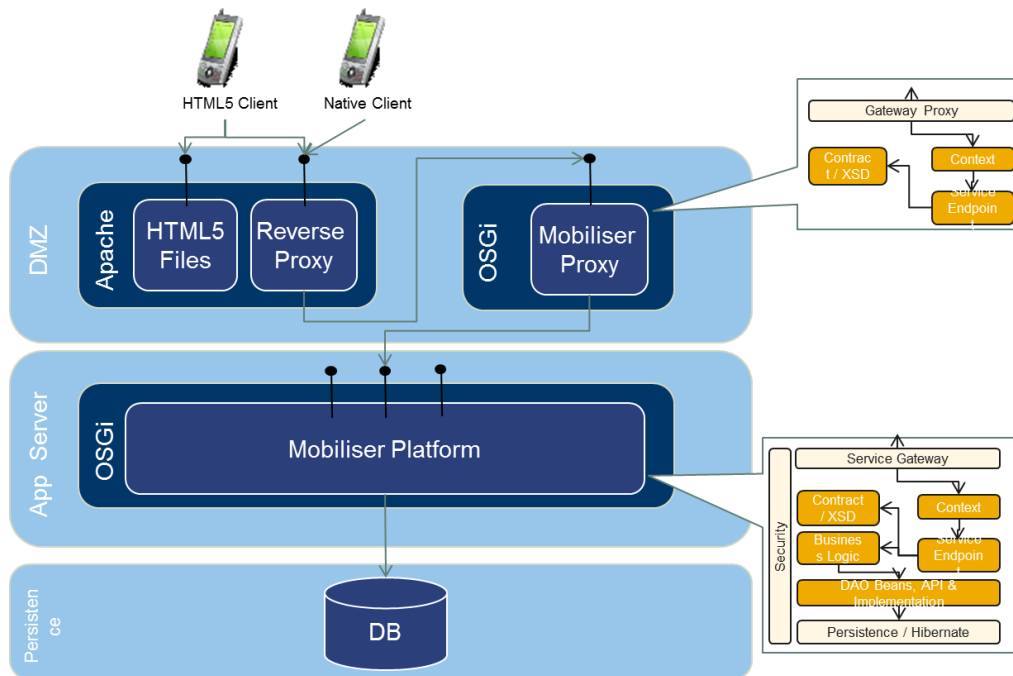
On mob-web-1, this file is located in:

```
/opt/sybase/proxy/conf/cfgbackup/com.sybase365.mobiliser.framework.service.proxy.properties
```

In addition to restricting access to certain services, the validating proxy makes sure that the incoming request corresponds to the contract definition (XSD) for the appropriate service. This check can be applied on all supported protocols (SOAP, plain XML, and JSON).

The validating proxy contains a subset of the bundles from the original Mobiliser Platform Core. It contains only the contract definitions (XSD) and the context and endpoint information.

When the request is validated successfully, it is forwarded to the Mobiliser Platform in its original format.



Brand Mobiliser

For installation and configuration information for Brand Mobiliser, see the *Sybase Brand Mobiliser User Manual* on the Sybase Product Documentation Web site.

On-Device Charging Installation and Configuration

On-device charging (ODC) provides the capability to store sensitive data, such as stored-value account (SVA) balances, on a smartphone, which can interact with external systems through near-field communication (NFC).

Provision Secure Element Keys for DIRECT Mode

Each new Secure Element (SE) that is issued by the Mobiliser Platform operator can be identified by a unique ID, and requires a specific keyset. The SE unique ID is stored in a structure called “Card Production Life Cycle (CPLC) data,” which uniquely identifies each SE and is stored into each SE prior to configuration.

The association between the SE and the unique keyset is usually provided by the SE manufacturer, and is generally required by the SE issuer, to maximize security. The card issuer would be taking a large risk if all SE accesses were based on a single keyset.

To deploy the MER on SE in DIRECT mode, ODC needs the CPLC/keyset pair for each SE. Each pair is stored in the ODC table ODC_DIRECT_SE_INFO.

The SE manufacturer generally includes an additional CSV file that contains all SE information (CPLC/keyset pair) with each SE batch. ODC registers the CSV files using:

```
###
CPLC_data; keyVersionNumber; keyIdentifier; key1_type; key1_
value ; key2_type; key2_value; key3_type; key3_value ###
CPLC_data: a string containing hexadecimal numbers.
keyVersionNumber: integer, a technical number provided by the
party who performs the keys installation at personalization
phase.
keyIdentifier: integer, a technical number provided by the party
who performs the keys installation at personalization phase.
keyX_type: one of the following strings (“senc”, “smac”, “dek”).
keyX_value: a string containing hexadecimal numbers.
```

Example of line:

```
2A4790502116716320431790159B5EE10C664647926242167362571674627200
000000000000000000000000,42,0,senc,25C69649A518622044BF1915BF65F7A
B7737CF11B26EA506F5DE6163B08CB876E1ED5DE0D3BF457F6418D321BAD62AE
DEA6220423A7FC87C08C0F71748CFF2CA20EDC29AECC6879C951D26861305F37
FE218288DA46A11D28B28603A8B0A6263686DDD19E4B894F31E2758E8EA53EBC
0EE7703A8565F87A5C90DE6B8201471AB3287B5A0477A9326A66CA390182BF79
4D6877D49283C168CC9EB2D44D63A8D5328D418F9BAADA7F5AA88E0466599092
7411AAFE13A84290783DC2C21BDED9751BB512592014C42C4E0CFFDD552D41E1
493E013D1F7C7DA03BA70E799D80A8CF9A4AA13DF1A31173330B4F20FC8BCBB5
D311FD5A9B7C9F418B0E81A591DD37229,smac,62ED9BAC8A66493F89B6BB6C5
079A2ED1F69646B98FCC49AECCFF76119C323AAD200C1BFAD210E7E0EBFD4D0B
CC5E86BB26F5092942AD3E2004AA87632776D27DF3AC71E5C944E423982C5B85
41E0B1E62EDC94D391E8FE1D31F226A450F9556F806DDDDC6FEE4E2B9EE6DD9F
96A822348F537417451C9B235D3B8E369D81B01C8DEDDBC96380603B5E976468
878B6EFF1C33D67E5323578B592ABEACBFC30E956E1233ABCA608B6A0EF09861
D2BB6943DA615ECCEFC95CCE2F423706C9F5FBFAFA206377C589D57F2D01C66CB
E6D1065365A14170BC25B56E4C45D930D623D82AFFB57F20B7A7CEDA0E473188
29382C492949FFB1454A12060F2AB467B0F349D,dek,55E2A101A100F498ECB7
C72563FE712CDD00682F174CA0C0BD4214FBAE9AA949A1FDF807AD78CBF6F887
C61320729FC7FF3CBEC696E32BBAE28B43830B9883E108D04DFFE359D815215
1D38E94CECA041C14C1C91D5C77850CD18EC753BF49C326ACAAF024D34D5F997
```

```
9CF8BC37D1EE3BDBD4EAE66C9BF8022A8CD78C5E73B2BF2C04764C86203B989E
90BCB7C1C8CD2F78F8B6580C3C669B4D544D4367C6D465AC0D456F106654942E
75BF216746284ACC3A14C6F60ABB721814AA6DBA9B4F2BCC772379EB02EEB83D
ADC899182EC825CC8F0FA1932E235AAA3979D54C0721DF9A3837B7AB0DFAB6AA
1A4C864F1FF566758F22CEA42F2B6FF94016F21AF2B3
```

To guarantee the security, all the key values will have to be ciphered by the SE manufacturer with the public key stored into the `mobiliser_odc.crt` file. Refer to step 10 on page 23 of the *Creating the Keystore for Data Encryption* section.

The ciphering algorithm that must be used by the SE manufacturer is RSA/ECB/PKCS1Padding.

To install this file into the `ODC_DIRECT_SE_INFO` table the following command has to be used:

```
java -jar com.sap.odc.tool.securtiy.odckeytool.jar
populate_se_info -url < mobiliser_url>
-login <mobiliser_user_login> -passwd <passwd> -csv
<csvFilePath>]
```

where `<mobiliser_user_login>` and `<passwd>` authenticates the user against Money Mobiliser after having successfully passed the Mobiliser HTTP gateway.

Note: This command can be executed each time a new SE batch file is registered. At last, this command manages the doubletons.

Generate Private Keys Used by On-Device Charging

By default, encrypting communications between the MER and the point of sale (POS), ODC requires two root keys—MPcK (Mer Private chargeKey) and MPrK (Mer Private readKey)—that are installed into each MER, and generate a specific and separate keyset for each merchant. The keys, which are 192 bits in size, are used by 3-DES algorithms (DESede/CBC/PKCS5Padding).

In addition, ODC requires an additional key—MPsK (Mer Private signingKey)—for signing the transactions and producing the eToken (a signed transaction). By default, the encryption algorithm used by ODC/MER for signing the generated transactions is RSA/ECB/PKCS1Padding.

However, the user can alternatively switch on a 3-DES algorithm (DESede/CBC/PKCS5Padding) to generate a smaller signature size and, thus, increase the number of eTokens that can be stored into the secure element. This option requires the Bouncycastle package on the server side for verifying the eTokens.

To generate the required keys, execute:

```
java -jar com.sap.odc.tool.securtiy.odckeytool.jar
gen_odc_keys -url < mobiliser_url>
-login <mobiliser_user_login> -passwd <passwd> [-desSigning]
```

where `<mobiliser_user_login>` and `<passwd>` authenticates the user against Money Mobiliser after having successfully passed the Mobiliser HTTP gateway.

Note: If you specify the `-desSigning` option, the eToken signing process uses 3-DES instead of the default RSA signing algorithm.

Standard Installation

This command:

- Automatically generates all the keys used by 3-DES algorithms,
- Reads the private RSA signing key from the Mobiliser Platform key store,
- Encrypts all those private keys by using the mobiliser_odc_se_ks private key, and
- Stores them into the security keyset database.

You can generate keys only once, during installation. Attempting to generate ODC private keys multiple times prevents the deployed MER from communicating with the existing registered merchant POS, and also prevents existing customers from using ODC with new merchants.

Start Mobiliser Platform

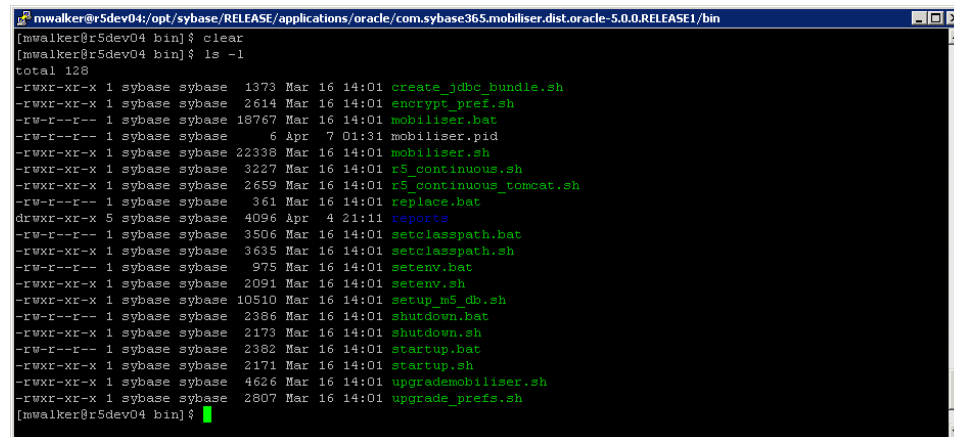
Starting the Server and User Interface

Execute all operations as the sybase user.

1. Log into the mob-aps-1.
2. Execute the start-up script:

```
/opt/sybase/money/bin/startup.sh
```

Note: shutdown.sh and other admin scripts are also located in this directory.



```
mwalker@r5dev04:/opt/sybase/RELEASE/applications/oracle/com.sybase365.mobiliser.dist.oracle-5.0.0.RELEASE1/bin
[mwalker@r5dev04 bin]$ clear
[mwalker@r5dev04 bin]$ ls -l
total 128
-rwxr-xr-x 1 sybase sybase 1373 Mar 16 14:01 create_jdbc_bundle.sh
-rwxr-xr-x 1 sybase sybase 2614 Mar 16 14:01 encrypt_pref.sh
-rw-r--r-- 1 sybase sybase 16767 Mar 16 14:01 mobiliser.bat
-rw-r--r-- 1 sybase sybase 6 Apr 7 01:31 mobiliser.pid
-rwxr-xr-x 1 sybase sybase 22338 Mar 16 14:01 mobiliser.sh
-rwxr-xr-x 1 sybase sybase 3227 Mar 16 14:01 r5_continuous.sh
-rwxr-xr-x 1 sybase sybase 2659 Mar 16 14:01 r5_continuous_tomcat.sh
-rw-r--r-- 1 sybase sybase 361 Mar 16 14:01 replace.bat
drwxr-xr-x 5 sybase sybase 4096 Apr 4 21:11 reports
-rw-r--r-- 1 sybase sybase 3506 Mar 16 14:01 setclasspath.bat
-rwxr-xr-x 1 sybase sybase 3635 Mar 16 14:01 setclasspath.sh
-rw-r--r-- 1 sybase sybase 975 Mar 16 14:01 setenv.bat
-rwxr-xr-x 1 sybase sybase 2091 Mar 16 14:01 setenv.sh
-rwxr-xr-x 1 sybase sybase 10510 Mar 16 14:01 setup_m5_db.sh
-rw-r--r-- 1 sybase sybase 2386 Mar 16 14:01 shutdown.bat
-rwxr-xr-x 1 sybase sybase 2173 Mar 16 14:01 shutdown.sh
-rw-r--r-- 1 sybase sybase 2382 Mar 16 14:01 startup.bat
-rwxr-xr-x 1 sybase sybase 2171 Mar 16 14:01 startup.sh
-rwxr-xr-x 1 sybase sybase 4626 Mar 16 14:01 upgrademobiliser.sh
-rwxr-xr-x 1 sybase sybase 2807 Mar 16 14:01 upgrade_prefs.sh
[mwalker@r5dev04 bin]$
```

3. Monitor the server log at /opt/sybase/money/logs/felix.out until the log specifies AutoDeploy finished.

```

mwalker@r5dev04:/opt/sybase/RELEASE/applications/oracle/com.sybase365.mobiliser.dist.oracle-5.0.0.RELEASE1/logs
Welcome to Apache Felix Gogo

2012-04-07 01:31:44.434:INFO:oejs.Server:jetty-7.x.y-SNAPSHOT
2012-04-07 01:31:44.503:INFO:oejs.AbstractConnector:Started NIOSocketConnectorWrapper@0.0.0.0:8080 STARTING
2012-04-07 01:31:46.137:INFO:oejsh.ContextHandler:started HttpServiceContext(httpContext=org.apache.felix.webconsole.internal.servlet.OsgiManagerHttpContext@7d6ac92e)
2012-04-07 01:31:46.185:INFO:oejsh.ContextHandler:stopped HttpServiceContext(httpContext=org.apache.felix.webconsole.internal.servlet.OsgiManagerHttpContext@7d6ac92e)
2012-04-07 01:31:46.535:INFO:oejsh.ContextHandler:started HttpServiceContext(httpContext=org.apache.felix.webconsole.internal.servlet.OsgiManagerHttpContext@2c4dd413)
Persistence bundle starting...
Persistence bundle started.
2012-04-07 01:31:55.470:INFO:oejsh.ContextHandler:started HttpServiceContext(httpContext=DefaultHttpContext(bundle=com.sybase365.mobiliser.framework.gateway.httpservice [308]))
2012-04-07 01:31:55.679:INFO:/:Initializing Spring FrameworkServlet 'Mobiliser'
2012-04-07 01:31:56.794:INFO:oejsh.ContextHandler:started HttpServiceContext(httpContext=DefaultHttpContext(bundle=com.sybase365.mobiliser.framework.gateway.security.filters.session [314]))
2012-04-07 01:33:03.181:INFO:oejsh.ContextHandler:started HttpServiceContext(httpContext=org.ops4j.pax.web.extender.war.internal.WebAppWebContainerContext@3ddb7e6d)
2012-04-07 01:33:03.907:INFO:oejsh.ContextHandler:started HttpServiceContext(httpContext=DefaultHttpContext(bundle=com.sybase365.mobiliser.util.management.logic [1]))
Aut@Deploy finished
0
53,1 27%

```

- Verify that the Mobiliser Platform console has initialized successfully by viewing the customer WSDL in your Web browser (<https://mob-aps-1:8443/mobiliser/customer/Custom.wSDL>).

```

Mozilla Firefox
http://r5dev04:8080/_atone_/Customer.wsd
r5dev04:8080/mobiliser/customer/Custom.wsd

This XML file does not appear to have any style information associated with it. The document tree is shown below.

- <wsdl:definitions targetNamespace="http://mobiliser.sybase365.com/money/customer">
- <wsdl:types>
- <xs:schema attributeFormDefault="unqualified" elementFormDefault="unqualified" jxb:extensionBindingPrefixes="jgc" jxb:version="2.0" targetNamespace="http://mobiliser.sybase365.com/framework/contract/v5_0base">
- <xs:annotation>
- <xs:appinfo>
- <jxb:schemaBindings>
- <jxb:package name="com.sybase365.mobiliser.framework.contract.v5_0base"/>
- <jxb:schemaBindings>
- <jxb:globalBindings generateIsSetMethod="false">
- <jcs:serializable uid="17">
- <jxb:globalBindings>
- <xs:appinfo>
- <xs:documentation>
The XML Schema for mobiliser requests. Version: $HeadURL: http://ormoco.sybase.com/svn/mobiliser/m5/framework/tags/com.sybase365.mobiliser.framework-5.0.0.RELEASE1/contract/contract/main/resources/com/sybase365/mobiliser/framework/contract/v5_0base-5.0.xsd $
- <xs:documentation>
- <xs:annotation>
- <xs:simpleType name="strSmall">
- <xs:restriction base="xs:string">
- <xs:maxLength value="6"/>
- <xs:minLength value="0"/>
- </xs:restriction>
- </xs:simpleType>
- <xs:simpleType name="strSmallNonEmpty">
- <xs:restriction base="strSmall">
- <xs:maxLength value="6"/>
- <xs:minLength value="1"/>
- </xs:restriction>

```

- To start the UI, log into mob-web-1 and execute:
/opt/sybase/portal/bin/startup.sh

Note: If the Mobiliser Platform installation is performed on server names that are different from the default names in this guide, you **MUST** change the MOBILISER_HOST variable in the /opt/sybase/portal/bin/setenv.sh script before running the startup.sh script. The MOBILISER_HOST reflects the URL the Web portal is connecting to (default value = <https://mob-aps-1:8443>).

Standard Installation

```
mwalker@r5dev04:/opt/sybase/tomcat/bin
[mwalker@r5dev04 bin]$ ls -l
total 612
-rw-r--r-- 1 sybase sybase 22705 Aug 16 2011 bootstrap.jar
-rw-r--r-- 1 sybase sybase 11830 Aug 16 2011 catalina.bat
-rwxr-xr-x 1 sybase sybase 17708 Aug 16 2011 catalina.sh
-rw-r--r-- 1 sybase sybase 2374 Aug 16 2011 catalina-tasks.xml
-rw-r--r-- 1 sybase sybase 24172 Aug 16 2011 commons-daemon.jar
-rw-r--r-- 1 sybase sybase 199623 Aug 16 2011 commons-daemon-native.tar.gz
-rw-r--r-- 1 sybase sybase 1342 Aug 16 2011 cpappend.bat
-rw-r--r-- 1 sybase sybase 2108 Aug 16 2011 digest.bat
-rwxr-xr-x 1 sybase sybase 1689 Aug 16 2011 digest.sh
-rw-r--r-- 1 sybase sybase 3150 Aug 16 2011 setclasspath.bat
-rwxr-xr-x 1 sybase sybase 4114 Aug 16 2011 setclasspath.sh
-rwxr-xr-x 1 sybase sybase 694 Mar 13 00:01 setenv.sh
-rw-r--r-- 1 sybase sybase 2108 Aug 16 2011 shutdown.bat
-rwxr-xr-x 1 sybase sybase 1628 Aug 16 2011 shutdown.sh
-rw-r--r-- 1 sybase sybase 2109 Aug 16 2011 startup.bat
-rwxr-xr-x 1 sybase sybase 2023 Aug 16 2011 startup.sh
-rw-r--r-- 1 sybase sybase 26828 Aug 16 2011 tomcat-juli.jar
-rw-r--r-- 1 sybase sybase 241274 Aug 16 2011 tomcat-native.tar.gz
-rw-r--r-- 1 sybase sybase 3479 Aug 16 2011 tool-wrapper.bat
-rwxr-xr-x 1 sybase sybase 3472 Aug 16 2011 tool-wrapper.sh
-rw-r--r-- 1 sybase sybase 2113 Aug 16 2011 version.bat
-rwxr-xr-x 1 sybase sybase 1632 Aug 16 2011 version.sh
[mwalker@r5dev04 bin]$
```

6. Verify that the Tomcat Web UI application has initialized successfully by viewing it in your Web browser (<https://mob-web-1:8082/portal>).



Note: If you have configured a Mobile Web installation within the {TOMCAT_HOME} location, the login page is available at: <http://mob-web-1/mobileweb>.

Starting Proxy

Execute this operation as the sybase user.

1. Log into mob-web-1.
2. Execute the startup script `/opt/sybase/proxy/bin/startup.sh` to start the proxy container on mob-web-1.

All incoming mob-web-1 requests, related to smartphone access, are validated through this proxy before reaching the endpoint on mob-aps-1.

Starting Internal Tomcat

Execute this operation as the sybase user.

1. Log into mob-aps-1.
2. To start the internal Tomcat container, execute the start-up script on mob-aps-1:

```
/opt/sybase/portal/bin/startup.sh
```

This container provides an internal administrative portal for changes to the deployed Mobiliser Platform container.

Note: If the Mobiliser Platform installation is performed on server names that are different from the default names in this guide, you **MUST** change the MOBILISER_HOST variable in the `/opt/sybase/portal/bin/setenv.sh` script before running the `startup.sh` script. The MOBILISER_HOST reflects the URL the Web portal is connecting to (default value = `https://mob-aps-1:8444`).

Starting Apache HTTPD

Execute this operation as the user with access to the Apache HTTPD service.

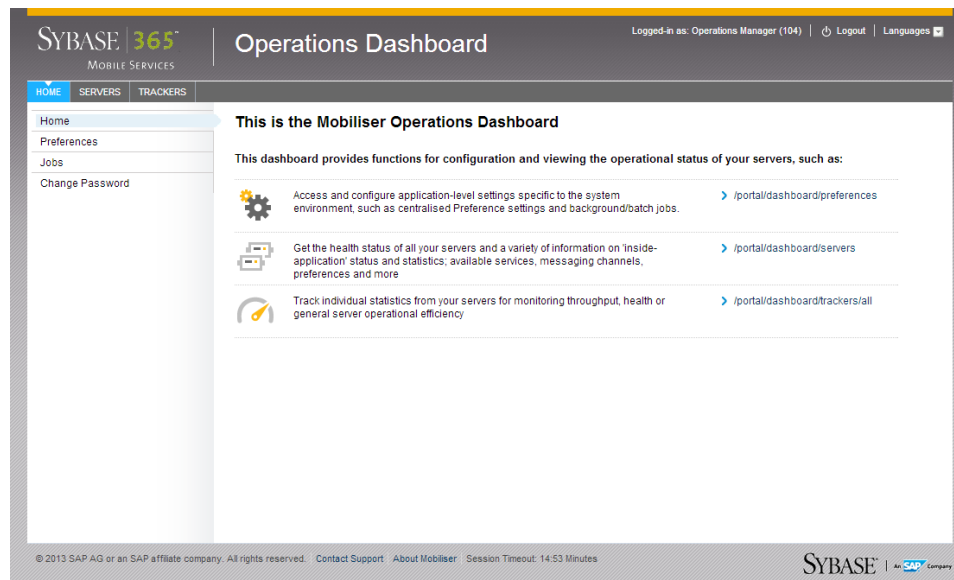
1. Log into mob-web-1.
2. Start or restart the Apache HTTPD server. The exact command differs by installation, for example, the RedHat command is:

```
service httpd restart
```

Configuring Preferences

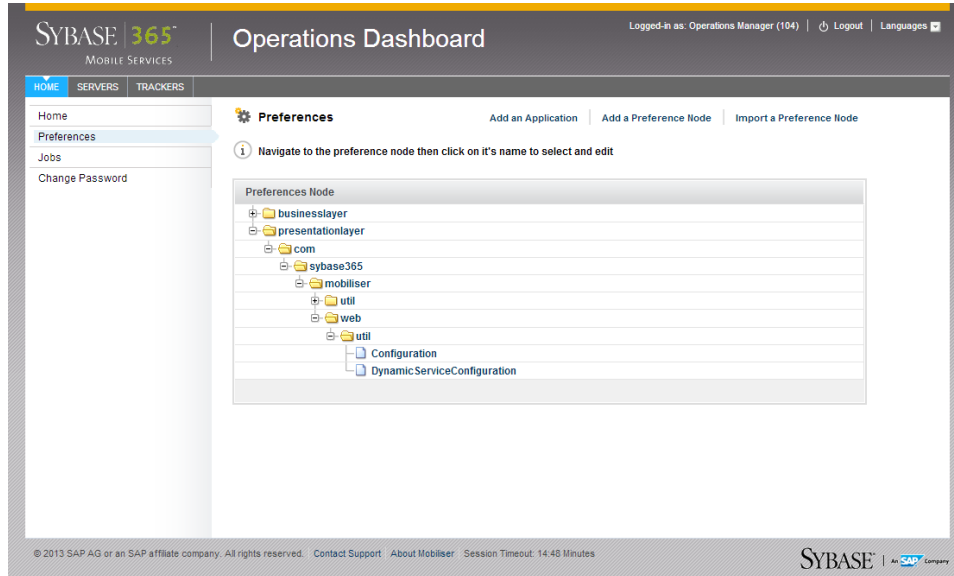
After installation, you might need to change some Mobiliser Platform application configurations. This section describes the general procedure for making such changes.

1. Log into the Operations Dashboard at `https://mob-aps-1:8442/portal` as the opsmgr user. You are immediately prompted to change the password.



Standard Installation

2. In the left pane, select **Preferences**. The main section of the page shows the Preferences tree with two root nodes “businesslayer” and “presentationlayer.”



When you click a node, you see a new page that lists all the Preferences for that node. Each page shows 10 entries; you might need to navigate through the pages to find the entry you are looking for.

The screenshot shows the 'Preferences » Node' configuration page in the Sybase 365 Operations Dashboard. The page is titled 'Operations Dashboard' and shows the user is logged in as 'Operations Manager (104)'. The left sidebar contains navigation options: Home, Preferences, Jobs, and Change Password. The main content area shows the 'Selected Node' configuration with the following details:

- Application: presentationlayer
- Node Full Path: /com/sybase365/mobiliser/web/Util/Configuration

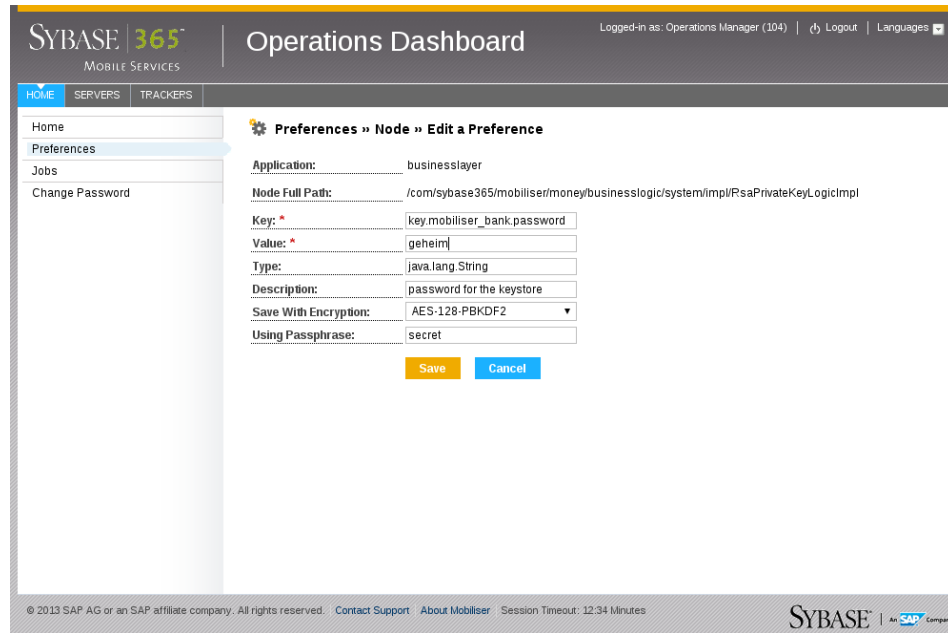
Below this, there are buttons for 'Refresh', 'Remove', 'Export', and 'Cancel'. The 'Preferences' section shows a table of 31 - 40 (50 Total) entries. The table has three columns: Key, Value, and Actions. The actions for each entry are 'Edit' and 'Remove'.

Key	Value	Actions
otpTemplateType	sms	Edit Remove
publicKeyStore	mobiliser_pub.jks	Edit Remove
reportProxyServerPath	/ReportViewer	Edit Remove
reportProxyServerUrl	http://localhost:8080/crystalrpt	Edit Remove
reportServerUrl	ReportViewer	Edit Remove
resetPasswordTemplateType	email	Edit Remove
resetPinTemplateType	sms	Edit Remove
riskcategories	0,1,2,3,4,5,6,7	Edit Remove
smsOtpTemplate	otpsignup	Edit Remove
svaCurrency	EUR	Edit Remove

At the bottom of the page, there is a footer with the text: '© 2013 SAP AG or an SAP affiliate company. All rights reserved. Contact Support About Mobiliser Session Timeout: 14:31 Minutes SYBASE | An SAP Company'.

- Add a new preference entry or change an existing one. To enter an encrypted value (for example, for passwords or other sensitive information), select **AES-128-PBKDF2** from the “Save with encryption” list, and provide a passphrase.
- Configure the secret used for the presentation layer in the portal under:
`/opt/sybase/portal/conf/context.xml`
- Define the secret for the business layer in:
`/opt/sybase/money/conf/cfgload/com.sybase365.mobiliser.util.prefs.encryption.aes.properties`
The default for both is “secret”.

Standard Installation



Keystore Configuration

Use the general procedure described in the previous section to configure passwords for the keystore and keys that encrypt and decrypt data in Mobiliser Platform. The *Creating the Keystore for Data Encryption* section on page 22 contains the instructions for setting up the keystore.

Path	/businesslayer/com/sap/odc/core/security/manager/SecurityConfigProvider/	
Key	Description	
key.se.password	Defines the password for the private key that was created with the mobiliser_odc_se_ks alias.	
key.signing.password	Defines the password for the private key that was created with the mobiliser_odc_signing alias.	
key.store.password	Defines the password for the keystore mobiliser.jks file.	

Path	/businesslayer/com/sybase365/mobiliser/money/businesslogic/system/impl/RsaPublicKeyLogicImpl/	
Key	Description	
key.store.password	Defines the password for the keystore mobiliser.jks file.	

Path	/businesslayer/com/sybase365/mobiliser/money/businesslogic/system/impl/RsaPrivateKeyLogicImpl/	
Key	Description	
key.store.password	Defines the password for the keystore mobiliser.jks file.	
key.mobiliser_card.password	Defines the password for the private key that was created with the mobiliser_card alias.	
key.mobiliser_bank.password	Defines the password for the private key that was created with the mobiliser_bank alias.	

Path	/businesslayer/com/sybase365/mobiliser/money/businesslogic/payment/handlers/card/impl/DummyCardPaymentHandler/	
Key	Description	
key.store.password	Defines the password for the keystore mobiliser.jks	
key.password	Defines the password for the private key that was created with the mobiliser_card alias.	

Message Properties Configuration

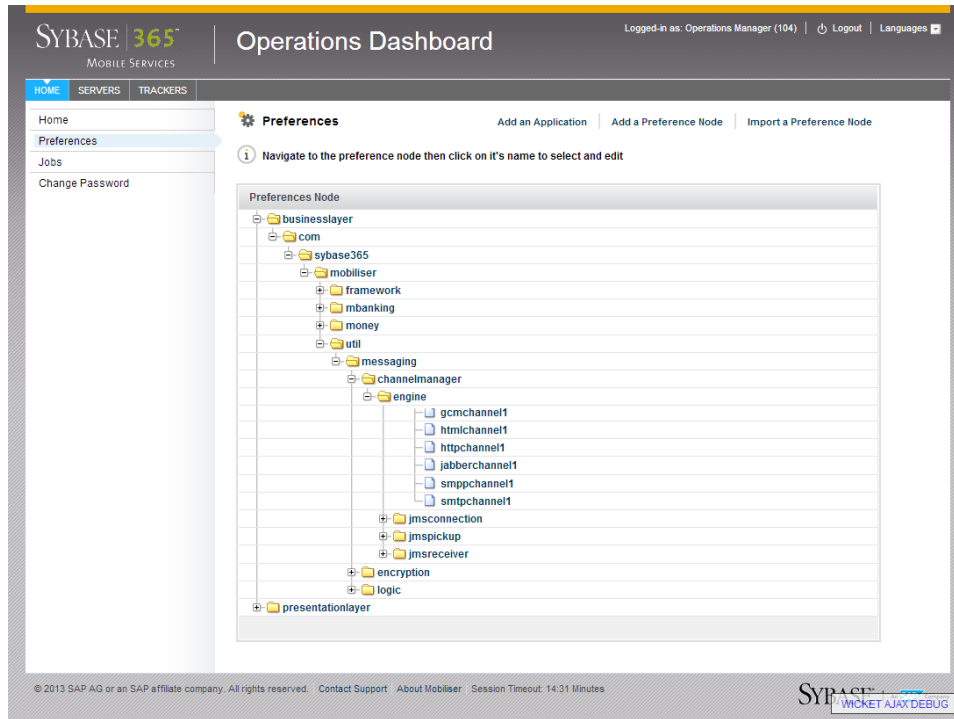
Email Security message control properties allow you to set message size and volume limits, and determine how invalid recipients are handled. For details about the Google Cloud Messaging (GCM) configuration options, see the *GCMChannel* section of the *Sybase Mobiliser Platform System Administration* guide.

Configuring Short Message Peer to Peer

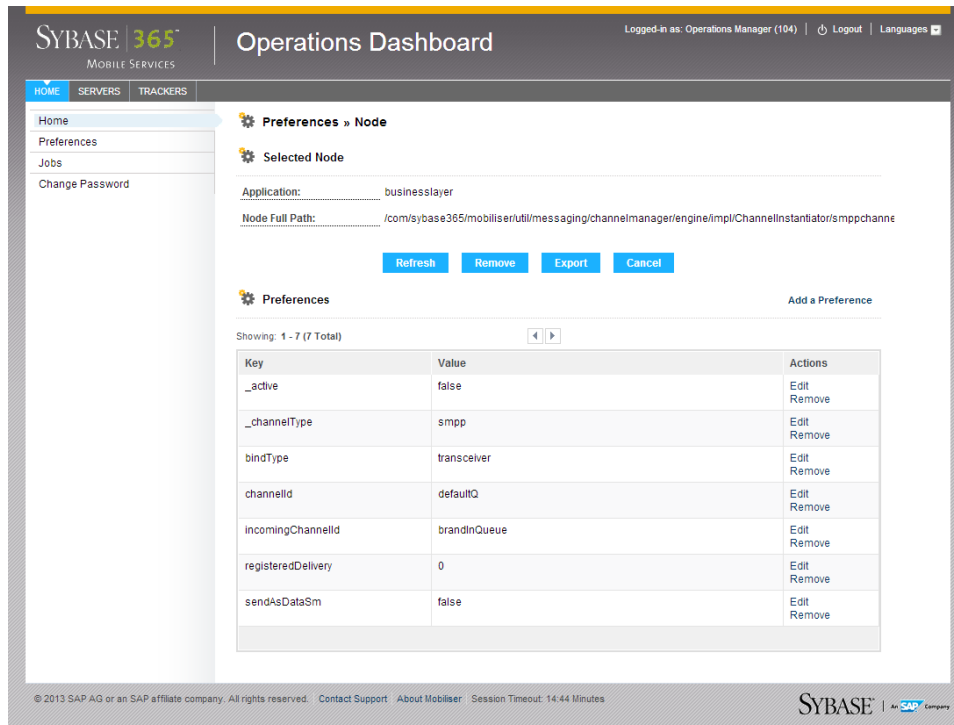
(Optional) Use the Operations Dashboard to configure the short message peer to peer (SMPP) messaging connector settings.

1. Log into the Operations Dashboard as the opsmgr user.
2. In the left pane, select **Preferences**.
3. Expand to the following path:
/businesslayer/com/sybase365/mobiliser/util/messaging/channel manager/engine/
4. Click **smpchannel1**.

Standard Installation



5. Navigate through the node preferences, and enter all relevant SMPP account information.



You can change these preferences:

Path	/businesslayer/com/sybase365/mobiliser/util/messaging/channelmanager/engine/smppchannel1
Key	Description
_active	Toggles the function of the SMPP communication channel.
_channelType	Describes the messaging protocol for the communication channel.
bindType	Describes the connection type the communication channel needs to connect to the short message service center (SMSC). These connection types are transmitter, receiver, and transceiver.
channelID	Defines the internal queue used by Mobiliser Platform to store outbound messages before delivery to SMSC. For active connection, use defaultQ as the default value.
tx.host	Defines the IP address or host name for the SMSC accepting connections from this Mobiliser Platform communication channel.
tx.password	Defines the authentication password set by the SMSC to allow this Mobiliser Platform communication channel to make an SMPP connection.
tx.port	Defines the port opened up on the SMSC to allow this Mobiliser Platform communication channel to make an SMPP connection.
tx.systemID	Defines the authentication user name set by the SMSC to allow this Mobiliser Platform communication channel to make an SMPP connection.
tx.systemType	Defines the type of SMPP account that the SMSC has set up to be available to the Mobiliser Platform communication channel, for example, external short messaging entity.
tx.usingSSL	Specifies whether SSL is used to deliver messages to the SMSC.

6. Click **Refresh** to ensure that preference changes are committed.

Configuring Simple Mail Transfer Protocol

Use the Operations Dashboard to configure the simple mail transfer protocol (SMTP) settings.

1. Log into the Operations Dashboard as the opsmgr user.
2. In the left pane, select **Preferences**.
3. Click **Add a Preference Node**.
4. Select **businesslayer** in the Application field.
5. In the Full Node Path field, enter:
com/sybase365/mobiliser/util/messaging/channelmanager/engine/
6. Click **Save**.
7. Expand to the following path:
/businesslayer/com/sybase365/mobiliser/util/messaging/channelmanager/engine/
8. Click **smtpchannel1**.
9. Click **Save**.

Standard Installation

10. Navigate to the newly created preference node in the preference tree.
11. Double-click the **smtpchannel1** node.

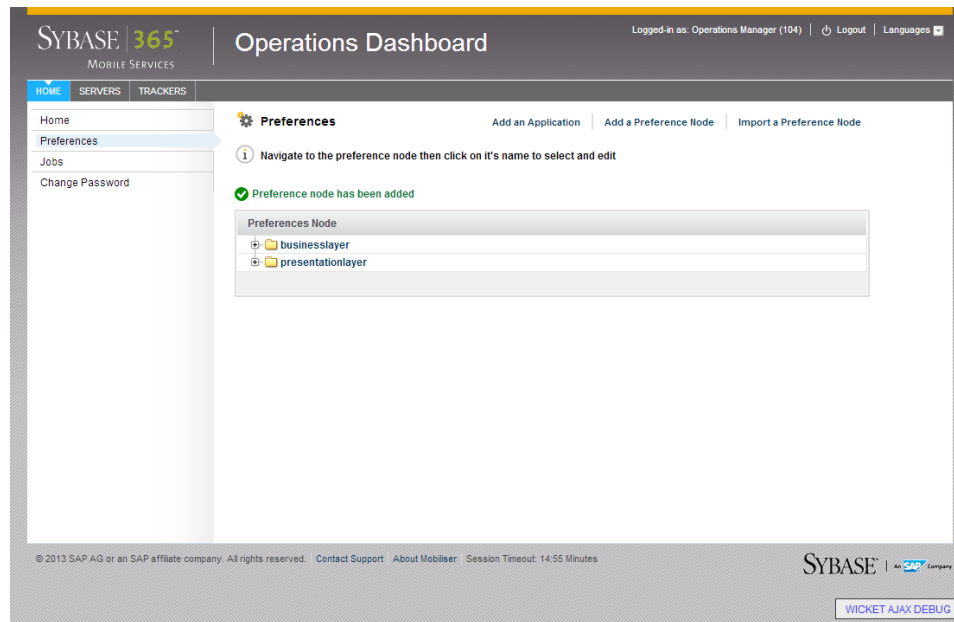
The essential keys associated with the SMTP communication channel include:

Path	/businesslayer/com/sybase365/mobiliser/util/messaging/channelmanager/engine/ smtpchannel1
Key	Description
_active	Toggles the function of the SMTP communication channel.
_channelType	Describes the messaging protocol for the communication channel.
channelID	Defines the internal queue used by Mobiliser Platform to store outbound messages before delivery to outbound mail server. For active connection, use defaultQ as the default value.
mail.host	Defines the IP address or host name for the outbound mail server that the Mobiliser Platform messaging channel uses.
mail.port	Defines the port for the outbound mail server that the Mobiliser Platform messaging channel uses.
mail.protocol	Describes the messaging protocol for the communication channel.

12. Click **Add a Preference** and enter the following values:

Key	Value	Type
channelType	Email	java.lang.String
channeled	Default	java.lang.String
mail.host	Localhost	java.lang.String
mail.port	25	java.lang.String
mail.protocol	Smtplib	java.lang.String
mail.sign	False	java.lang.String
sign.hashAlgorithm	-1	java.lang.String
sign.keyId	-1	java.lang.String

13. Click **Save** after each entry.
14. Click **Refresh** to ensure that preference changes are committed.



Virus Protection

Antivirus software is one of the most important tools for safe-guarding vital information and personal data from the daily onslaught of viruses and worms.

Configuring the Virus Scan Adapter for SAP NetWeaver

Mobiliser Platform 5.1 introduces the Virus Scan Adapter for SAP NetWeaver[®], which scans all files uploaded to the Mobiliser Platform via Web services. The adapter uses a plug-in to connect to various virus scan engines that scan the binary data. For more details, see *Setting Up Virus Scan Providers* located at:

http://help.sap.com/saphelp_nw04/helpdata/EN/ca/7cb340be761b07e10000000a155106/frame/eset.htm

1. Install/copy the virus scan adapter for your virus scanner, which is provided by your virus scan vendor.

The NW-VSI integration bundle comes with a graphical configuration and test GUI, which is part of the VSI bundle:

```
{ $MOBILISER_HOME } / bundles / 07-frameworks / nw.vsi-1.92.0.jar
```

2. Start the GUI:

```
$> java -jar $MOBILISER_HOME / bundles / 07-frameworks /  
com.sap.security.vsi${version}.jar
```

Test the connection with the European Institute for Computer Antivirus Research (EICAR) test pattern and mark the provider as the default. The Mobiliser Platform engine uses only the default provider.

Standard Installation

3. Open the Mobiliser Platform configuration file:
`${mobiliser_home}/conf/cfgbackup/com.sybase365.mobiliser.framework.vsi.properties`
4. Copy all lines from the vsi.properties file and replace the similar ones in the Mobiliser Platform configuration file.
5. Restart the Mobiliser Platform bundle (or the complete container) and examine the mobiliser.log file.
6. Verify that there are no WARN entries, such as:

```
2012-08-28 08:22:10,768 [aims-init-10] WARN
com.sybase365.mobiliser.framework.vscan.impl.VScanImpl -
Cannot initialize Virus Scan Service. The following service
exception occurred: Virus scan provider VSA_DEFAULT does not
exist.
2012-08-28 08:22:10,890 [aims-init-10] INFO
com.sybase365.mobiliser.framework.vscan.impl.VScanImpl - No
virus scan is performed.
```

If there are WARN statements, then the virus scan software is not functioning properly.

Clam AV

One widely used UNIX virus scan engine is ClamAV, which utilizes specialized packages to recognize viruses and other threats. You can use the ClamSAP library to connect the virus scan adapter to the ClamAV engine. This document includes the required steps for installing and configuring the Mobiliser Platform 5.1 virus scan adapter with ClamAV on a Linux server.

The Mobiliser Platform 5.1 virus scan adapter with ClamAV requires:

- ClamAV – is the virus scan engine and development package
- libclamsap – is required to recognize potential threats

The first package is usually available from Linux distributors. The latest version of libclamsap is available from:

<http://sourceforge.net/projects/clamsap/files/>

Use libclamsap when Mobiliser Platform can access a local ClamAV engine.

Configuring ClamAV

1. To configure the ClamAV adapter, enable the default adapter, then edit the adapter path to point to the libclamsap shared library:

```
com.sybase365.mobiliser.framework.vsi.properties
(...)
vsi.provider.Virus_Scan_Adapter.Adapters.VSA_DEFAULT=VSA_DEFAULT
vsi.provider.Virus_Scan_Adapter.Adapters.VSA_DEFAULT.Active=true
vsi.provider.Virus_Scan_Adapter.Adapters.VSA_DEFAULT.AdapterPath=/home/
sybase/libclamsap.so
vsi.provider.Virus_Scan_Adapter.Adapters.VSA_DEFAULT.Description=DEFAULT
PROVIDER
vsi.provider.Virus_Scan_Adapter.Adapters.VSA_DEFAULT.Group=DEFAULT
```


Standard Installation

```
vsi.provider.Virus_Scan_Adapter.Adapters.VSA_DEFAULT.PoolInstanceTimeOut=3600
```

```
vsi.provider.Virus_Scan_Adapter.Adapters.VSA_DEFAULT.PoolMaxInstances=50
```

```
vsi.provider.Virus_Scan_Adapter.Adapters.VSA_DEFAULT.ReInitTime=0
```

(...)

2. Restart Mobiliser Platform and examine the log.

The adapter is loaded successfully when you see these log lines in the mobiliser.log file:

(...)

```
2012-09-06 08:43:48,747 [aims-init-15] DEBUG  
com.sybase365.mobiliser.framework.vscan.scanner.impl.VScanImpl - VSI  
Virus Scan Service initialization was successful
```

(...)

If the adapter is not loaded successfully, the virus scan installation failed. You will need to analyze the log file to troubleshoot the issue.

Validate the Installation

Validating the Mobiliser Platform installation is a simple process. You must register as a test consumer to run an end-to-end test that verifies that the Mobiliser Web Portal application can access the SOAP services. Optionally, you can log in with one of the predefined accounts.

Default Web UI Accounts

The installation process adds a set of predefined accounts that have the special privileges that are required to administer the portal system configuration. You must use these same accounts to manage user accounts, notifications and alerts, and merchants.

Note: When you log in using the predefined password, you are prompted to change it immediately.

Account	Description
cstfull:secret	Full administration portal privileges
usermgr:secret	Manage user accounts
notifmgr:secret	Manage notifications and alerts
headquarter:secret	Create and manage merchants
opsmgr:secret	View and manage system configuration
sysmgr:secret	Monitor all functions of the Mobiliser Platform container

Signing up as a Test Consumer

Registering as a test consumer verifies that a new Money Mobiliser consumer can sign up.

1. From the Mobiliser Login page, click **Consumer Signup**.

Note: You can also use the Mobile Web to sign up a test consumer.

2. Select a consumer type.
3. Enter all required information.
The default time zone is Europe/Berlin.
4. Enter the CAPTCHA characters.
5. Accept the terms and conditions, then click **Continue**.
6. Accept the terms of the license agreement, then click **Next**.
7. Review your information and click **Continue**.
A one-time passcode (OTP) is created.
8. Log into the Channel Manager (<http://<localhost>:8080/mobiliser/channelmgr/html>), using the following credentials:
 - a. User name: mobiliser
 - b. Password: secret
9. Enter the OTP.

10. Click **Continue** to finalize your registration.
11. Click **Continue** again to return to the Mobiliser Login page.
12. Log into the portal using the newly created user credentials.

Note: If you are unable to register a test consumer, then there is a problem with the system. If you are unable to log into the portal after registering a test consumer, ensure you are using the correct password for the test account.

Stop Installation

You can stop or shut down the various components of a working Mobiliser Platform instance.

Stopping Apache HTTPD

Execute this operation with the user who has access to the Apache HTTPD service.

1. Log into mob-web-1.
2. Stop the Apache HTTPD server.

The exact command differs by installation, for example, the RedHat command is:

```
service httpd stop
```

Stopping Internal/External Tomcat

Execute all operations with the sybase user.

1. Log into mob-web-1.
2. Shut down the external Tomcat container:

```
/opt/sybase/portal/bin/shutdown.sh
```

3. Log into mob-aps-1.

4. Shut down internal Tomcat container:

```
/opt /sybase/portal/bin/shutdown.sh
```

Stopping Proxy

Execute this operation with the sybase user.

1. Log into mob-web-1.
2. Execute:

```
/opt/sybase/proxy/bin/shutdown.sh
```

Stopping Server

Execute this operation with the sybase user.

1. Log into mob-aps-1.
2. Execute:

```
/opt/sybase/money/bin/shutdown.sh
```

Stop Installation

Index

A

accounts, 45
ASE configuration, 11

B

Brand Mobiliser, 28

C

card production life cycle (CPLC), 29
checklist
 preinstallation, 1
ClamAV, 43
 configuring, 43
configure
 ASE, 11
 ClamAV, 43
 database properties, 16
 HTTPD server, 26
 java management extensions (JMX), 23
 keystore, 37
 message properties, 38
 ODC (on-device charging), 28
 on-device charging (ODC), 28
 preferences, 34
 properties, 22
 short message peer to peer, 38
 simple mail transfer protocol, 40
 SMPP, 38
 SMTP, 40
 virus scan adapter, 42
copy files, 10
CPLC (card production life cycle), 29
create
 database schema, 11
 directory structure, 10
 encrypted password, 15
 hashed password, 14
 HTTPD key, 20
 internal portal tomcat key, 18
 JDBC driver bundle, 23, 24
 jetty key, 16
 keystore for database encryption, 21
 master application user, 8
 tomcat key, 18

D

database
 ASE configuration, 11
 configuring properties, 16
 create schema, 11
 install, 11
 run dbmaintain, 12
database properties
 configuring, 16
dbmaintain, 12
 running, 13
default user accounts, 45
deployment model, 3

E

encrypted password preferences
 create, 15
 update, 15

G

generate private keys, 30
getting started, 1

H

hashed password
 creating, 14
 updating, 14
HTTPD server
 Brand Mobiliser, 28
 proxy, 27
 smartphone mobile internet version, 27
 starting, 34
 stopping, 47

I

install
 database, 11
 java card packages, 9
 java virtual machine (JVM), 9
 JVM (java virtual machine), 9
 ODC (on-device charging), 28
 on-device charging (ODC), 28

- reporting module, 25
- spring source, 25
- third-party software, 23, 24

installation checklist, 7

J

- JDBC drivers, 12
- JMX (java management extensions updating), 23

K

- keystore
 - configure, 37

M

- master application user
 - creating, 8
- message properties
 - configuration, 38
 - configuring short message peer to peer, 38
 - configuring simple mail transfer protocol, 40
 - configuring SMPP, 38
 - configuring SMTP, 40

N

- network ports, 8

O

- ODC (on-device charging), 28
 - configuration, 28
 - generating private keys, 30
 - installation, 28
 - installing java card packages, 9
 - provision secure element keys, 29
- on-device charging (ODC), 28
 - configuration, 28
 - generating private keys, 30
 - installation, 28
 - installing java card packages, 9
 - provision secure element keys, 29
- operations dashboard
 - configuring preferences, 34

P

- preferences
 - configuring, 34
 - creating encrypted password, 15
 - updating encrypted password, 15
- preinstallation checklist, 1
 - preinstalled software, 1
 - required skills and permissions, 1
- preinstalled software, 1
 - properties
 - message, 38
 - updating, 22
- provision secure element keys, 29
- proxy, 27
 - starting, 33
 - stopping, 47

R

- required permissions, 1
- required skills, 1
- run dbmaintain, 12, 13

S

- secure element keys, 29
 - provision, 29
- security
 - keystores, 16
- security keystores, 16
 - creating HTTPD key, 20
 - creating jetty key, 16
 - creating keystore for database encryption, 21
 - creating tomcat key, 18
- server
 - start, 31
 - stopping, 47
- sign up as test consumer, 45
- smartphone mobile internet version, 27
- software
 - installing third-party, 23, 24
- standard deployment model, 3
- standard installation, 7
 - checklist, 7
 - configuring database properties, 16
 - configuring HTTPD server, 26
 - copying files, 10
 - creating directory structure, 10

Index

- install java card packages, 9
- installing database, 11
- java virtual machine (JVM), install, 9
- JVM (java virtual machine), install, 9
- master application user, create, 8
- network ports, 8
- ODC (on-device charging), 28
- on-device changing (ODC), 28
- security keystores, 16
- third-party software, 23, 24
- unpacking software, 9
- updating configuration properties, 22
- updating default configuration, 13
- start
 - HTTPD server, 34
 - Mobiliser Platform, 31
 - proxy, 33
 - server, 31
 - tomcat, 34
 - user interface, 31
- stop
 - HTTPD, 47
 - installation, 47
 - proxy, 47
 - server, 47
 - tomcat, 47

T

- third-party software, 23, 24

- creating JDBC driver bundle, 23, 24
- installing reporting module, 25
- installing spring source, 25
- tomcat
 - creating key, 18
 - starting, 34
 - stopping, 47

U

- unpack software, 9
- update
 - configuration properties, 22
 - default configuration, 13
 - encrypted password, 15
 - hashed password, 14
- user accounts, 45
- user interface accounts, 45

V

- validate installation, 45
 - signing up as test consumer, 45
- virus protection, 42
 - ClamAV, 43
 - configuring virus scan adapter, 42
- virus scan adapter, 42