



Sybase Mobiliser Platform Release Bulletin

Version 5.0 SP02

Document ID: DC01862-01-0500-03

Last Revised: May 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

Product Summary	1
Upgrading to Mobiliser Platform 5.0 SP02 PL01.....	1
Prerequisites	1
Automated Upgrade of an Existing Container.....	1
Syntax.....	2
Examples	2
Options	2
Exit Codes	2
Project Upgrade in Maven Build Files	2
If Upgrading From Mobiliser Platform 5.0 SP01 or Earlier to 5.0 SP02	2
Web UI Upgrade.....	3
Upgrading the Web UI from 5.0 SP01 or Earlier.....	3
Upgrading a Service Package	3
Upgrading the Web UI	3
Additional Upgrade Notes for Upgrading From 5.0 SP01 or Earlier	4
Added Security on Smartphone Mobiliser Services	4
Cleanup Expired Authorization Task Configuration	8
Broken Oracle Trigger	8
Setting up a Local Development Environment.....	8
Customization Project Template	8
Prerequisites.....	8
Initializing a Local Nexus Repository Server	9
Known Issues in 5.0 SP02.....	9

Product Summary

Sybase® Mobiliser Platform is a state-of-the-art service-oriented architecture (SOA) for dynamic, real-time integration and rapid application development. The solution empowers mobile operators and financial institutions to quickly launch mobile commerce services to their customers in both developed and emerging markets.

This release bulletin discusses known issues, workarounds, as well as late-breaking documentation updates for Mobiliser Platform 5.0 SP02.

Upgrading to Mobiliser Platform 5.0 SP02 PL01

You can upgrade to Sybase Mobiliser Platform 5.0 SP02 PL01 from Mobiliser Platform 5.0 SP02.

Important: If you are using an earlier version of Mobiliser Platform, you must upgrade to Mobiliser Platform 5.0 SP02 before proceeding. All update scripts for previous versions are included with the installation image in their associated directories.

For example, if you are upgrading from the Mobiliser Platform 5.0 base version, you need to run each upgrade script in successive order until you are upgraded to 5.0 SP02.

Prerequisites

Before upgrading:

- Stop running the Money Mobiliser container and Money Mobiliser Web UI application.
 - For the Money Mobiliser container: `money/bin/shutdown.sh`
 - For the Web UI: `web/bin/shutdown.sh`
- Make a backup of your container in case of a failed upgrade.
 - Back up your MOBILISER_HOME directory, for example, `/opt/sybase/com.sybase365.mobiliser.dist.ase-5.0.0.SP02`
 - Back up your TOMCAT_HOME directory, for example, `/opt/sybase/apache-tomcat-6.0.33`

Tip: To conserve storage space and reduce backup time, you do not need to back up the contents of the logs directories.

Automated Upgrade of an Existing Container

Each service package gets delivered with an upgrade script included in the container ZIP file. Find the `installservicepack.sh` script in the `./bin` folder. This script applies a service package to an already installed Mobiliser Platform container. All existing product bundles that exist in this installed container are replaced with the latest corresponding bundle from the service package. Any files found in the service package but not in your container are not re-added. It is assumed they were purposely removed. Any new configuration files are added to your container. Any configuration files / startup scripts you have edited are not overwritten. Instead the new files are placed in the configuration directory with a `.new` suffix to allow the system administrator to inspect the changes manually. The database does not have to be changed during a service package upgrade.

Note: You can perform this upgrade procedure from any Mobiliser Platform 5.0 version directly to 5.0 SP02 PL01.

Syntax

```
installservicepack.sh [options] <installation_directory>  
<container_filename>
```

Examples

```
installservicepack.sh /home/mobiliser/money  
/tmp/staging/com.sybase365.mobiliser.vanilla.oracle-5.0.0.SP02PL01.zip
```

Options

- -dry-run (show the changes, but do not upgrade)
- -verbose (show diffs for configuration file changes)
- -md5sum (check the md5sum of the new container before installation)

Exit Codes

0 – success
4 – md5sum mismatch when check requested
5 – md5sum file for package missing
6 – no unzip utility found to unzip service pack
7 – rsync not installed
10 – caught signal (killed)

Project Upgrade in Maven Build Files

To upgrade to the latest service package, replace any product dependencies versions with the latest service package version number and rebuild and release your project. No database upgrade or changes to your database scripts are required. No changes to your source code are required if not stated differently in the list of fixed issues in the cover letter/readme.

If you followed the template customization project, you have a section in the parent POM that lists the version of all product dependencies. Update the settings in the parent POM so that they point to the latest support pack release.

```
<properties>  
  <manifest.name>com/sybase365/mobiliser/custom/project</manifest.name>  
  <manifest.title>Sybase365 Mobiliser Customization Project  
  </manifest.title>  
  <manifest.spec-version>1.0.0</manifest.spec-version>  
  <version.framework>5.0.0.SP02PL01</version.framework>  
  <version.vanilla>5.0.0.SP02PL01</version.vanilla>  
  <version.money>5.0.0.SP02PL01</version.money>  
  <version.mobiliser-tools>5.0.0.SP02PL01</version.mobiliser-tools>  
  ...  
</properties>
```

If Upgrading From Mobiliser Platform 5.0 SP01 or Earlier to 5.0 SP02

This information applies only when upgrading from Mobiliser Platform 5.0 SP01 or earlier version to Mobiliser Platform 5.0 SP02.

Due to changes to the container packaging mechanism, you need to rename any references to the Vanilla artifacts from `com.sybase365.mobiliser.vanilla.${container.type}` to `com.sybase365.mobiliser.vanilla.standalone`. Perform a search-and-replace in your distribution POM as well as the distribution assembly XML located at `./dist/pom.xml` and

`./dist/src/main/assembly/dist.xml`. Additionally, add an `<excludes>` section to the references to the Vanilla standalone artifacts to prevent non-required database drivers from being pulled in, meaning, exclude all database driver artifacts that you do not need in your specific installation. For example, if you are using Oracle:

```
<exclusions>
  <exclusion>
    <groupId>org.postgresql</groupId>
    <artifactId>com.sybase365.org.postgresql.jdbc4</artifactId>
  </exclusion>
  <exclusion>
    <groupId>com.ibm</groupId>
    <artifactId>com.sybase365.com.ibm.db2jcc4</artifactId>
  </exclusion>
  <exclusion>
    <groupId>com.sybase365.arf.thirdparty</groupId>
    <artifactId>jconnect-osgi</artifactId>
  </exclusion>
</exclusions>
```

Next, check the `./conf/system-<DBMS>.properties` file from the Vanilla standalone distribution and merge the content with any custom changes you might have done to the `system.properties` file that is part of the customization distribution module (located in `src/main/resources/money/conf`).

Web UI Upgrade

No web UI upgrade is required from 5.0 SP02 to 5.0 SP02 PL01.

Upgrading the Web UI from 5.0 SP01 or Earlier

The following procedures are required if upgrading from Mobiliser Platform 5.0 SP01 or earlier versions.

The upgrade procedure for the Web-UI is more involved than updating a container. Each customization requires branching the code and then proceeding with its customization work; that is, there is no clear separation between product and customization code anymore and the changes are not as easy applied as exchanging updated artifacts. The key to upgrading is determining what has changed in your customization.

Upgrading a Service Package

Each service package delivers a patch file in the `./upgrade` folder of the distribution ZIP file. This patch file contains the changes made to the Web-UI in this service package. To apply the fixes to your customized Web-UI, run both scripts or just the second script, depending on from which version you are upgrading:

```
patch -p0 --merge < webui_5.0.0.RELEASE1_to_5.0.0.SP01.diff
patch -p0 --merge < webui_5.0.0.SP01_to_5.0.0.SP02.diff
```

Upgrading the Web UI

1. First locate a copy of the vanilla source of the version of the Web-UI that was the base of your branch.

If your customization is based on 5.0.0.RELEASE1, get a copy of the original sources and extract them into a directory.

2. Next make a copy of your sources and place them in a second directory. With these two directories, perform a diff, which contains only the changes made to your customization.

```
diff -E -N --strip-trailing-cr -ru src.old src.cur > customization.diff
```

3. Now if you are using version control software, replace the Web-UI sources in it with the vanilla sources of the next version and commit this as the base for the next step. This includes deleting any added files - the idea being to bring the state up to the pure vanilla version of the Web-UI.

This step is not strictly necessary, but makes it easy to see exactly where you started with the newest version and which changes were yours and which came only from the upgrade. The next step involves replaying the changes you made on top of the new version, fixing any conflicts along the way. So after you have your working copy updated to the newest release, it is time to rebase your changes on top of it.

```
patch -p0 --merge < customisation.diff
```

This leaves merge markers in the files for any conflicts that arise. Now go through and resolve the conflicts and check in the resulting state. Be mindful when resolving merge-conflicts since the leading and trailing lines contain those lines from the old version so they can be misleading when resolving changes.

Additional Upgrade Notes for Upgrading From 5.0 SP01 or Earlier

The following procedures are required if upgrading from Mobiliser Platform 5.0 SP01 or earlier. If upgrading from 5.0 SP02, these steps are already included in your installation.

Added Security on Smartphone Mobiliser Services

We have identified a few potential gaps in the current services that are exposed for Smartphone Mobiliser. Some of the existing services are now protected with a new privilege by default. The standard Smartphone Mobiliser application does not use these services and should not be affected.

If the project is using any of the services, you can make them available again by adding the privileges to the database (script provided). This should only be done on a case-by-case basis as needed!

```
GetCustomerResponse getCustomer(final GetCustomerRequest request);
INSERT INTO MOB_UMGR_PRIVILEGES (ID_PRIVILEGE, STR_PRIVILEGE, ID_CUSTOMER_CREATION)
VALUES ('WS_SPM_GET_CUSTOMER', 'Allows to call the SPM web service getCustomer',
0);
INSERT INTO MOB_UMGR_ROLE_PRIVILEGES (ID_ROLE, ID_PRIVILEGE, ID_CUSTOMER_CREATION)
VALUES ('STANDARD_CONSUMER', 'WS_SPM_GET_CUSTOMER', 0);
--

DeleteCustomerResponse deleteCustomer(DeleteCustomerRequest request);
```

```
INSERT INTO MOB_UMGR_PRIVILEGES (ID_PRIVILEGE, STR_PRIVILEGE, ID_CUSTOMER_CREATION)
VALUES ('WS_SPM_DELETE_CUSTOMER', 'Allows to call the SPM web service
deleteCustomer', 0);
```

```
INSERT INTO MOB_UMGR_ROLE_PRIVILEGES (ID_ROLE, ID_PRIVILEGE, ID_CUSTOMER_CREATION)
VALUES ('STANDARD_CONSUMER', 'WS_SPM_DELETE_CUSTOMER', 0);
```

--

GetCustomerByIdentificationResponse getCustomerByIdentification(

```
INSERT INTO MOB_UMGR_PRIVILEGES (ID_PRIVILEGE, STR_PRIVILEGE, ID_CUSTOMER_CREATION)
VALUES ('WS_SPM_GET_CUST_BY_ID', 'Allows to call the SPM web service
getCustomerByIdentification', 0);
```

```
INSERT INTO MOB_UMGR_ROLE_PRIVILEGES (ID_ROLE, ID_PRIVILEGE, ID_CUSTOMER_CREATION)
VALUES ('STANDARD_CONSUMER', 'WS_SPM_GET_CUST_BY_ID', 0);
```

--

PickupMoneyResponse pickupmoney(final PickupMoney request);

```
INSERT INTO MOB_UMGR_PRIVILEGES (ID_PRIVILEGE, STR_PRIVILEGE, ID_CUSTOMER_CREATION)
VALUES ('WS_SPM_PICKUP', 'Allows to call the SPM web service pickupmoney', 0);
```

```
INSERT INTO MOB_UMGR_ROLE_PRIVILEGES (ID_ROLE, ID_PRIVILEGE, ID_CUSTOMER_CREATION)
VALUES ('STANDARD_CONSUMER', 'WS_SPM_PICKUP', 0);
```

--

SetCredentialResponse setCredential(final SetCredentialRequest request);

```
INSERT INTO MOB_UMGR_PRIVILEGES (ID_PRIVILEGE, STR_PRIVILEGE, ID_CUSTOMER_CREATION)
VALUES ('WS_SPM_SET_CREDENTIAL', 'Allows to call the SPM web service
setCredential', 0);
```

```
INSERT INTO MOB_UMGR_ROLE_PRIVILEGES (ID_ROLE, ID_PRIVILEGE, ID_CUSTOMER_CREATION)
VALUES ('STANDARD_CONSUMER', 'WS_SPM_SET_CREDENTIAL', 0);
```

--

CreateIdentificationResponse createIdentification(

```
INSERT INTO MOB_UMGR_PRIVILEGES (ID_PRIVILEGE, STR_PRIVILEGE, ID_CUSTOMER_CREATION)
VALUES ('WS_SPM_CREATE_IDENT', 'Allows to call the SPM web service
createIdentification', 0);
```

```
INSERT INTO MOB_UMGR_ROLE_PRIVILEGES (ID_ROLE, ID_PRIVILEGE, ID_CUSTOMER_CREATION)
VALUES ('STANDARD_CONSUMER', 'WS_SPM_CREATE_IDENT', 0);
```

--

UpdateIdentificationResponse updateIdentification(

```
INSERT INTO MOB_UMGR_PRIVILEGES (ID_PRIVILEGE, STR_PRIVILEGE, ID_CUSTOMER_CREATION)
VALUES ('WS_SPM_UPDATE_IDENT', 'Allows to call the SPM web service
updateIdentification', 0);
```

```
INSERT INTO MOB_UMGR_ROLE_PRIVILEGES (ID_ROLE, ID_PRIVILEGE, ID_CUSTOMER_CREATION)
```

```

VALUES ('STANDARD_CONSUMER', 'WS_SPM_UPDATE_IDENT', 0);
--

CaptureResponse capture(final Capture request);
INSERT INTO MOB_UMGR_PRIVILEGES (ID_PRIVILEGE, STR_PRIVILEGE, ID_CUSTOMER_CREATION)
VALUES ('WS_SPM_CAPTURE', 'Allows to call the SPM web service capture', 0);
INSERT INTO MOB_UMGR_ROLE_PRIVILEGES (ID_ROLE, ID_PRIVILEGE, ID_CUSTOMER_CREATION)
VALUES ('STANDARD_CONSUMER', 'WS_SPM_CAPTURE', 0);
--

UpdateInvoiceConfigurationResponse updateInvoiceConfiguration(
INSERT INTO MOB_UMGR_PRIVILEGES (ID_PRIVILEGE, STR_PRIVILEGE, ID_CUSTOMER_CREATION)
VALUES ('WS_SPM_UPDATE_INVOICE', 'Allows to call the SPM web service
updateInvoiceConfiguration', 0);
INSERT INTO MOB_UMGR_ROLE_PRIVILEGES (ID_ROLE, ID_PRIVILEGE, ID_CUSTOMER_CREATION)
VALUES ('STANDARD_CONSUMER', 'WS_SPM_UPDATE_INVOICE', 0);
--

DeleteInvoiceConfigurationResponse deleteInvoiceConfiguration(
INSERT INTO MOB_UMGR_PRIVILEGES (ID_PRIVILEGE, STR_PRIVILEGE, ID_CUSTOMER_CREATION)
VALUES ('WS_SPM_DELETE_INV_CONFIG', 'Allows to call the SPM web service
WS_SPM_DELETE_INV_CONFIG', 0);
INSERT INTO MOB_UMGR_ROLE_PRIVILEGES (ID_ROLE, ID_PRIVILEGE, ID_CUSTOMER_CREATION)
VALUES ('STANDARD_CONSUMER', 'WS_SPM_DELETE_INV_CONFIG', 0);
--

AuthorisationCancelResponse authorisationCancel(
INSERT INTO MOB_UMGR_PRIVILEGES (ID_PRIVILEGE, STR_PRIVILEGE, ID_CUSTOMER_CREATION)
VALUES ('WS_SPM_AUTH_CANCEL', 'Allows to call the SPM web service
authorisationCancel', 0);
INSERT INTO MOB_UMGR_ROLE_PRIVILEGES (ID_ROLE, ID_PRIVILEGE, ID_CUSTOMER_CREATION)
VALUES ('STANDARD_CONSUMER', 'WS_SPM_AUTH_CANCEL', 0);
--

CaptureCancelResponse captureCancel(final CaptureCancel request);
INSERT INTO MOB_UMGR_PRIVILEGES (ID_PRIVILEGE, STR_PRIVILEGE, ID_CUSTOMER_CREATION)
VALUES ('WS_SPM_CAPT_CANCEL', 'Allows to call the SPM web service
captureCancel', 0);
INSERT INTO MOB_UMGR_ROLE_PRIVILEGES (ID_ROLE, ID_PRIVILEGE, ID_CUSTOMER_CREATION)
VALUES ('STANDARD_CONSUMER', 'WS_SPM_CAPT_CANCEL', 0);
--

```

```

UpdateWalletEntryResponse updateWalletEntry(UpdateWalletEntryRequest request);
INSERT INTO MOB_UMGR_PRIVILEGES (ID_PRIVILEGE, STR_PRIVILEGE, ID_CUSTOMER_CREATION)
VALUES ('WS_SPM_UPDATE_WALLET', 'Allows to call the SPM web service
updateWalletEntry', 0);
INSERT INTO MOB_UMGR_ROLE_PRIVILEGES (ID_ROLE, ID_PRIVILEGE, ID_CUSTOMER_CREATION)
VALUES ('STANDARD_CONSUMER', 'WS_SPM_UPDATE_WALLET', 0);

```

Some services in Smartphone Mobiliser overwrite fields with default data to prevent a hacker from creating unwanted (privileged) accounts in Mobiliser Platform. You can overwrite the default values in the Preferences configuration.

Please make sure that you understand this behavior and that the default values make sense for you.

The following fields get overwritten inside the Smartphone Mobiliser endpoint:

updateCustomer Service

```

customer.customerTypeId = defaultCustomerType
customer.feeSetId = defaultFeeSetId
customer.feeVatPercentage = null
customer.limitSetId = defaultLimitSetId
customer.riskCategoryId = defaultRiskCategoryId

```

createFullCustomer Service

```

customer.customerTypeId = defaultCustomerType
customer.feeSetId = defaultFeeSetId
customer.feeVatPercentage = null
customer.limitSetId = defaultLimitSetId
customer.riskCategoryId = defaultRiskCategoryId

```

createWalletEntry Service

```
walletEntry.limitSetId = null
```

The default values are set in the Preferences.

The node is:

```
com.sybase365.mobiliser.money.services.smartphone.SmartphoneEndpoint
```

The keys are:

```

defaultCustomerType (default="2")
defaultFeeSetId (default=null)
defaultLimitSetId (default=null)
defaultRiskCategoryId (default=null)

```

Cleanup Expired Authorization Task Configuration

The currently installed `CancelExpiredTransactionsTask` is missing some configuration to work properly, and this information must be added manually.

```
insert into MOB_PREFERENCES
(ID_PREFERENCE,STR_PATH,STR_NAME,STR_VALUE,STR_TYPE,STR_DESCRIPTION) values
(350,'/businesslayer/system/com/sybase365/mobiliser/money/jobs/tasks/cancelexpired/CancelExpiredTransactionsTask/','internal.service.username','mobiliser','java.lang.String','internal service user');
insert into MOB_PREFERENCES
(ID_PREFERENCE,STR_PATH,STR_NAME,STR_VALUE,STR_TYPE,STR_DESCRIPTION) values
(351,'/businesslayer/system/com/sybase365/mobiliser/money/jobs/tasks/cancelexpired/CancelExpiredTransactionsTask/','internal.service.password','XXX','java.lang.String','internal service password');
```

(Replace XXX with the actual password of the Mobiliser Platform user in your installation.)

It is recommended to use the Preferences management section in the Operations Dashboard to add the missing Preferences entries.

Broken Oracle Trigger

The Oracle Trigger that creates history entries for the table `MOB_CUSTOMERS_CREDENTIALS` is broken in the current SQL scripts.

If you rely on data created by this trigger, please run the update script located in `upgrade/scripts/oracle/009_MONEY_fix_history_trigger.ddl`

Setting up a Local Development Environment

The current service pack comes with additional files, which help setting up a local development environment.

Customization Project Template

A sample customization project that holds samples for the most common customization tasks is located in the customization project in the distribution at

`tools/com.sybase365.mobiliser.dist.project-<VERSION>-customization.zip`.

If you extract that archive, you get a ready-to-build project that generates a dozen custom Mobiliser Platform bundles, and builds a Mobiliser Platform container that holds these custom bundles on top of a plain Mobiliser Platform container.

Prerequisites

For the custom Mobiliser Platform build to work, you need to set up a full local development environment. Recommendations include a source control system such as Subversion, a continuous integration server such as Jenkins, and a Maven repository server such as Nexus to manage your dependencies.

However, you can use other tools to manage your build environment. You can use any other mechanism to implement against the Mobiliser Platform APIs and package bundles for inclusion in the Mobiliser Platform container. However, using the recommended tools gives you a head-start of having some ready-to-go templates.

Initializing a Local Nexus Repository Server

For the customization project to build successfully, you need to configure a Maven repository server that holds all the required Mobiliser Platform artifacts, which are not accessible through public Maven repositories. Use the provided tool to load a Nexus server with all Mobiliser Platform artifacts that are part of the distribution. The upload script is located at `tools/com.sybase365.mobiliser.dist.tools-<VERSION>.jar`.

To run the upload, execute:

```
java -jar com.sybase365.mobiliser.dist.tools-<VERSION>.jar
./dependencies.properties ../ http://localhost/nexus repo admin
password
```

Tip: You can also run the tool without any parameters to get help on the parameters printed on the console.

1. The first argument must point to the property file holding the metadata on all Mobiliser Platform artifacts. It is included in the tools directory as well.
2. The second parameter must point to the root directory of where you have extracted the Mobiliser Platform distribution ZIP file.
3. The third parameter points to your Nexus server URL.
4. The fourth parameter is the name of the Nexus repository to which to upload the Mobiliser Platform dependencies. Create a dedicated repository on your server to host the Mobiliser Platform artifacts.
5. The fifth parameter is the Nexus user name that has upload privileges
6. The sixth parameter is the user's password

In addition, you need to include other public repositories for common open source libraries. The easiest way is to create a new group on your Nexus server that lists the following in the specified order. Be sure your local Maven settings configuration point to that Nexus group.

1. Maven Central (<http://repo1.maven.org/maven2/>)
2. Spring Enterprise External (<http://repository.springsource.com/maven/bundles/external/>)
3. Mobiliser Repository

Known Issues in 5.0 SP02

Bugzilla #	Description
4715	Issue: DB2::[Reports]::Error Message is displayed while generating Transaction detail report. Workaround: None.