



Installation Guide for SAP Mobile SDK

SAP Mobile Platform 2.3 SP02

Windows

DOCUMENT ID: DC01917-01-0232-01

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.

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

CHAPTER 1: Getting Started	1
Obtaining a License	1
Determining Host IDs	2
Comparing License Deployment Models	3
Generating Licenses at SMP	4
Generating License Keys	4
Generating Licenses at SPDC	5
Logging in to SPDC and Beginning License Generation	6
Locating Information in a License File	9
Setting a Fixed Port Number for SySAM License Server	11
 CHAPTER 2: Installing SAP Mobile SDK	 13
Preparing for Installation	13
Entering License Information	14
Selecting Installation Options	18
Completing the Installation	19
Verifying the Installation	20
Verifying an Installation on Same Host as SAP Mobile Server	20
Verifying a Remote Installation	21
Protecting the Installation	22
 CHAPTER 3: Installing an Upgraded SAP Mobile SDK	 25

CHAPTER 4: Installing the Agency Components of the SAP Mobile SDK	27
Installing the Agency Server for Windows	29
Installing the Agency Editor Plug-In and Eclipse Platform	36
Agency Editor and Eclipse Platform Configuration Overview	43
Creating a Connection Profile for the Agency Connector Studio	44
Creating Java Projects for Agency Java API Development	54
Configuring Eclipse File Associations for Agency Projects	60
Configuring Eclipse File Encoding for Agency Projects	66
Installing the Agency Test Environment	68
Agency SAP Framework Foundation Installation Overview	70
Agency SAP Framework Foundation Component	71
Installing Agency SAP Framework Foundation	71
Creating the JavaBE.ini File for SAP Systems	73
Installing the Agency SDK	78
 CHAPTER 5: Uninstalling SAP Mobile SDK	 81
 CHAPTER 6: Uninstalling Agency Editor	 83
 CHAPTER 7: Scripting Silent Installation or Uninstallation	 85
Silent Installation	85
Planning a Silent Installation	86

Modifying Properties in the SilentInstall_Win.txt File	86
Running a Silent Installation	86
Silent Uninstallation	87
Planning a Silent Uninstallation	87
Running a Silent Uninstallation	88
 CHAPTER 8: Troubleshooting	91
Updating the Database Connection Profile	91
 APPENDIX A: Locations of Code Libraries	93
 Index	97

CHAPTER 1 **Getting Started**

Before you begin installing SAP® Mobile SDK, familiarize yourself with system requirements and installation prerequisites.

Although SAP Mobile SDK works closely with the SAP Mobile Server component in SAP Mobile Platform Runtime, the installation of SAP Mobile SDK is independent of the SAP Mobile Server installation. You can install SAP Mobile SDK on a server or developer workstation, with or without SAP Mobile Server. If SAP Mobile Server has already been installed, you do not need to shut down the SAP Mobile Server services before installing or uninstalling SAP Mobile SDK.

For information about system requirements for SAP Mobile SDK, see *Supported Hardware and Software*.

For information about installing an SAP Mobile Server instance to work with your SAP Mobile SDK installation, see *Installing SAP Mobile Platform Runtime on a Single Server* in *Installation Guide for Runtime*.

New Installations

If you are installing SAP Mobile SDK on a system where there is no earlier version of either SAP Mobile SDK or SAP Mobile Server installed, you must provide a license. See *Obtaining a License* on page 1.

If you are installing SAP Mobile SDK on a system where a version of SAP Mobile Platform Runtime is already installed, the SAP Mobile SDK installer installs using the SAP Mobile Server license. See *Chapter 2, Installing SAP Mobile SDK* on page 13.

Upgrade Installations

If you are installing SAP Mobile SDK on a system where an earlier version has been installed, see *Chapter 3, Installing an Upgraded SAP Mobile SDK* on page 25.

Obtaining a License

There are two ways to obtain your SAP Mobile Platform license, depending on how you purchased the product.

When you purchase SySAM 2-enabled SAP products, you must generate, download, and deploy SySAM product licenses.

- If you ordered your product under an SAP® contract and were directed to download from SAP Service Marketplace (SMP), you can use SMP at <http://service.sap.com/licensekeys>

CHAPTER 1: Getting Started

(login required) to generate license keys for SAP products that use SySAM 2-based licenses.

- If you purchased your product from Sybase® or an authorized Sybase reseller, go to the secure Sybase Product Download Center (SPDC) at <https://sybase.subscribenet.com> and log in to generate license keys. The license generation process may vary slightly, depending on whether you ordered directly from Sybase or from a Sybase reseller.

For more complete information about SySAM, see:

- *SySAM 2 Users Guide* online at <http://infocenter.sybase.com/help/topic/com.sybase.infocenter.dc00530.0220/doc/html/title.html>
- *Fast Track to SySAM 2.0* white paper, available at *Fast Track to SySAM 2.0*
- *SySAM FAQ* online at <http://www.sybase.com/detail?id=1038615>
- *SySAM* product page online at <http://www.sybase.com/products/allproductsa-z/sysam>

Determining Host IDs

When you generate licenses at SMP or SPDC, you must specify the host ID of the machine where the licenses will be deployed.

- For unserved licenses, specify the host ID of the machine where the product will run. If you are running a product with per-CPU or per-CHIP licensing that supports SySAM sub-capacity, and you want to run the product in a virtualized environment, see *SySAM Sub-capacity Licensing* in the *SySAM Users Guide* for information about determining the host ID for unserved licenses.
- For served licenses, specify the host ID of the machine where the license server will run.

SMP or SPDC remembers the host information so that you can select the same license server when generating additional licenses.

To determine the host ID of the machine, run the **lmutil** utility from a terminal window or the Windows command prompt. For example:

```
lmutil lmhostid
```

Note: You can download the **lmutil** utility from the Flexera Software Web site at http://www.globes.com/support/fnp_utilities_download.htm.

You can also determine the host ID using native operating system commands. See the Frequently Asked Questions topic "What is my Host ID?":

- SMP: <https://websmp208.sap-ag.de/~sapidb/011000358700001006652011E> (requires login)
- SPDC: <https://sybase.subscribenet.com/control/sybs/faqs#30-4>

Comparing License Deployment Models

Learn about license deployment models.

Unserviced Licenses	Served Licenses
Licenses can be used only on the machine for which the license was generated.	Licenses can be distributed from a network license server to products running on any network machine.
Generate licenses at SMP or SPDC for each machine that will run the product: 1. Specify the host ID of the machine where the product will run. 2. Generate a license for that machine. 3. Save the license to the specified machine. 4. Repeat steps 1 – 3 for each machine where the product will run.	Generate licenses at SMP or SPDC for products running on multiple machines: 1. Specify the host ID of the license server. 2. Specify the number of required licenses. 3. Save the licenses to the license server host machine.
No license administration is required. However, when new licenses are required for product updates, you must update and deploy each license for each machine where the product update will run.	The license server requires administration. When new licenses are required for product updates, SMP or SPDC lets you update all licenses for a specific license server in a single step.
No license reporting or asset management capabilities are available.	Allows license monitoring and reporting of license use, capacity planning, and asset management using SAMreport.
Installed locally and always available.	Requires a functioning license server and network. If the license server or network fails, you must fix the problem or install an alternate license server before the product grace period expires.
If a machine where the product is running fails, you must regenerate all of its licenses and deploy those licenses to the replacement machines.	If a machine where the product is running fails, you can move the product to a new machine, and it will acquire licenses from the running license server. If the license server host machine fails, use the Manage License Hosts functionality at SMP or SPDC to move its licenses to a new network license server host.
License files are distributed across each machine running a product, and therefore they are difficult to manage and control.	License files are centrally located and managed.
Unserviced Standalone Seat (SS) licenses do not allow product use via Remote Desktop Connection or other terminal services clients.	Products can be used via Remote Desktop Connection or other Terminal Services clients, irrespective of the type of license in use.

Generating Licenses at SMP

Before you log in to SMP and generate licenses, gather the necessary information and complete these tasks.

Table 1. Information Needed Before License Generation

Required Information or Action	License Model		Description
	Served	Un-served	
License deployment model	X	X	Decide whether to use a served or unserved license deployment model. Typically, this is a company-wide decision that is made only once. Therefore, this is one of the most important things to determine before license generation.
Product machine host ID		X	Determine the host ID of the machine, or machine partition where the product will run.
License server – download and install	X		Download and install the SySAM license server before you generate the product’s licenses, and before you install the product.
License server host ID	X		Determine the host ID of the machine where the license server will run.
License server host name	X		Determine the host name of the machine where the license server will run.
License server TCP/IP port numbers	X		Determine the two port numbers on which the license server will listen for license requests.

Generating License Keys

If you have purchased SAP products that use SySAM 2-based licenses under SAP contract and are directed to download from SAP Service Marketplace (SMP), you can use SMP to generate license keys.

1. Go to the SAP Marketplace main page at <http://service.sap.com/licensekeys>.
2. Log in using your SMP credentials.
3. Select **SAP Support Portal**.
4. Select **Keys & Requests > License Keys**.

- Follow the instructions in the "How to generate license keys for SAP Sybase products" presentation available under the "Documentation and Helpful Resources" quick access link.

Generating Licenses at SPDC

Before you log in to SPDC and generate licenses, gather the necessary information and complete these tasks.

Table 2. Information Needed Before License Generation

Required Information or Action	License Model		Description
	Served	Un-served	
License deployment model	X	X	Decide whether to use a served or unserved license deployment model. Typically, this is a company-wide decision that is made only once. Therefore, this is one of the most important things to determine before license generation.
Product machine host ID		X	Determine the host ID of the machine, or machine partition where the product will run.
License server – download and install	X		Download and install the SySAM license server before you generate the product licenses, and before you install the product.
License server host ID	X		Determine the host ID of the machine where the license server will run.
License server host name	X		Determine the host name of the machine where the license server will run.
License server TCP/IP port numbers	X		Determine the port numbers on which the license server will listen for license requests. Note: If you do not specify the license server port numbers during license generation, the license server uses the first available ports in the range 27000 to 27009. If a firewall exists between the server and the client machines, fix the license server port numbers to allow access to the ports. See <i>Access Through a Firewall or VPN</i> in the <i>SySAM Users Guide</i> .

Logging in to SPDC and Beginning License Generation

Once you log in to SPDC and begin license generation, complete your product's license generation using the procedure that applies to the license deployment model you chose, either for generating served licenses or for generating unserved licenses.

If your product has been ordered from a SAP reseller, perform the “Web Key step” instructions, where noted.

1. Go to the SPDC login page at <https://sybase.subscribenet.com>.
 - Go to the SPDC Web Key Registration page at <https://sybase.subscribenet.com/webkey>.
2. Enter the login ID and password, then click **Login**. If you forget your password, click **Password Finder**.

Your password is sent to you in an e-mail message.

- Enter the authorization string from the Web Key Certificate provided by the reseller when your SAP product was purchased, then click **Submit Web Key**.

Note: If you do not know your SPDC account login ID and password, or Web Key Certificate authorization string, contact the person who ordered your product.

- At the Web Key Registration page, enter your account information, then click one of these options:
 - **Submit My Registration Information** – to register your product using your direct account information.
 - **Anonymous Activation** – to activate your product anonymously.
3. Select the product family that includes the product for which you want to generate a license; for example, Adaptive Server Enterprise.
 4. Depending on the product family you choose, you may see an additional Product Information page.
 - a. Product suites – if your product is included in one or more suites, select the suite that includes your product; for example, ASE Small Business Edition.
 - b. Product version and platform – select the product version, name, and operating system that matches your product order.
 5. If this is the first time you selected a specific product edition and platform, you must accept the SAP License Agreement before you are allowed to generate a license for that product.
 6. If your product software installation requires a license key (also referred to as the license file), click **License Keys** on the Product Download page.
 7. On the License Information page:
 - a. Select the option button to the left of the product for which to generate a license; for example, “CPU License (CP), ASE Enterprise Edition 15.7 for Sun Solaris SPARC 64-bit.”

- b. Scroll down and click **Select to Generate**.
8. In the Generate Licenses wizard, choose a license deployment model:
- **Served license** – go to *Generating Served Licenses* in the *Sybase Software Asset Management (SySAM) 2 Users Guide* to complete the license generation and download process.
 - **Unserved license** – go to *Generating Unserved Licenses* in the *Sybase Software Asset Management (SySAM) 2 Users Guide* to complete the license generation and download process.

Note: Some SAP products or license types do not offer a choice of license deployment models and do not display this page. If this is the case, continue through the Generate Licenses wizard to generate your license.

Click **Next**.

Generating Unserved Licenses

Generate and download an unserved license for your product.

1. Enter the number of machines (up to 10) for which to generate unserved licenses, and click **Next**.
2. Enter:
 - Node Host ID – enter the host ID of the machine where the product will be run. If you do not know the host ID, select **What Is My Host ID?** or see *Determining Host IDs* in the *Sybase Software Asset Management (SySAM) 2 Users Guide* at <http://infocenter.sybase.com/help/topic/com.sybase.infocenter.dc00530.0220/doc/html/title.html>.
 - Host Name – enter the machine host name.

For some license types, you must enter the number of licenses to generate. If you are unsure of the license quantity, select **How Many Licenses Should I Generate?**.

3. Click **Generate**.
4. When the license has generated, review the information on the View Licenses page, then, if the license information is correct, select one of:
 - If you generated only one license, click **Download License File**.
 - If you generated several licenses, click **Download All Licenses for Host**.

Note: Before you download and save generated licenses, you can select **Print Friendly** to print a paper copy of the license, or **License Overview** to return to the License Information page and generate additional licenses.

- To correct license information before downloading the license, click **License Overview**, select the license to be corrected and click **Check In** to reset the license to its original state. Repeat the license generation process.

5. If you chose to download licenses, when the File Download dialog box opens, click **Save**.
6. Save the generated licenses with a `.lic` file name extension. Although `$$SYBASE / SYSAM-2_0/licenses` is typically the directory where unserved licenses are located, your product may have a different requirement. See your product installation guide and release bulletin for product-specific information.

Note: If you do not save the license files with a `.lic` extension, SySAM does not recognize the licenses.

Next, install your licensed product using the instructions in the product installation guide and release bulletin.

Generating Served Licenses

Generate and download a served license for your product.

1. Enter the number of licenses to generate and click **Next**.

If you are unsure of the license quantity, select **How Many Licenses Should I Generate?**.

2. Select an existing license server host, or enter the host ID, and an optional host name and port number, of a new license server host.

When completing this step, use these recommendations:

- If you do not know the license server host ID, select **What Is My Host ID?** or see *Determining Host IDs* in the *Sybase Software Asset Management (SySAM) 2 Users Guide* at <http://infocenter.sybase.com/help/topic/com.sybase.infocenter.dc00530.0220/doc/html/title.html>.
- Although the host name is optional, SAP recommends that you provide the host name to ease future license administration.
- The port number is optional unless your product configuration is going to use three-server redundancy (see the next bullet item). Any unused port number between 0 and 64000 is valid. On UNIX, choose a port number greater than 1024; most port numbers less than 1024 are privileged port numbers. If you do not specify a TCP/IP port number, a default port between 27000 and 27009 is used.
- To generate a license for a three-server redundant configuration, enter the required information: license server host ID, host name, and port number for all three machines. Enter a port number outside of the 27000 to 27009 range. You must enter the fully qualified domain name (FQDN) as the host name if any client machine requires an FQDN to contact a license server machine.

Note: You cannot upgrade SySAM 1.0 licenses and use them in a three-server redundant configuration.

3. Click **Generate**.

4. When the license has generated, review the information on the View Licenses page, then, if the license information is correct and you do not need to generate additional licenses, select one of:

- If you generated only one license, click **Download License File**.
- If you generated several licenses, click **Download All Licenses for Host**.

Note: Before you download and save generated licenses, you can select **Print Friendly** to print a paper copy of the license.

- a) To correct license information, click **License Overview**, select the license to be corrected and click **Check In** to reset the license to its original state. Repeat the license generation process, starting with step 1 of this procedure.
 - b) To generate additional licenses, click **License Overview** and repeat the generation process for the additional product licenses.
5. When the File Download dialog box opens, click **Save**.
 6. Save the license files with a `.lic` file name extension to the `SYSAM-2_0/licenses` directory of the license server installation.

Warning! If you do not save the license files with a `.lic` extension, SySAM does not recognize the licenses.

7. After you save the license files to the license server, enter in a command prompt (Windows) or a shell window (Linux/UNIX) on the machine where your license server is running:

```
sysam reread
```

Your new licenses are registered with the license server.

Locating Information in a License File

After you download a license file, you may need to get some information from it to complete your installation.

1. Use a text editor to open your license file.
2. Locate the uncommented line that begins with the string for your SAP Mobile Platform edition:
 - Enterprise Edition – `INCREMENT SUP_ENTSRVR`
 - Enterprise Developer Edition – `INCREMENT SUP_ENTDEV`
 - Personal Developer Edition – `INCREMENT SUP_DEVELOPER`

For example:

- For Enterprise Edition:

```
...
INCREMENT SUP_ENTSRVR SYBASE 2011.11150 permanent uncounted \
  VENDOR_STRING=PE=EE;LT=CP HOSTID=000c29d300bd
```

CHAPTER 1: Getting Started

```
PLATFORMS="i86_n \  
...
```

- For Enterprise Developer Edition:

```
...  
INCREMENT SUP_ENTDEV SYBASE 2011.11150 permanent uncounted \  
    VENDOR_STRING=PE=EE;LT=CP HOSTID=000c29d300bd  
PLATFORMS="i86_n \  
...
```

- For Personal Developer Edition:

```
...  
INCREMENT SUP_DEVELOPER SYBASE 2011.11150 permanent uncounted \  
    VENDOR_STRING=PE=EE;LT=CP HOSTID=000c29d300bd  
PLATFORMS="i86_n \  
...
```

The rest of the examples in this section show the beginning of this line as it appears for Enterprise Edition; the details equally apply to all editions.

3. Determine whether the server license is served or unserved.

If the line you located in step 2 ends with "uncounted", it is an unserved license. For example:

```
...  
INCREMENT SUP_ENTSRVR SYBASE 2011.11150 permanent uncounted \  
    VENDOR_STRING=PE=EE;LT=CP HOSTID=000c29d300bd PLATFORMS="i86_n  
\  
...
```

If that line ends with a number immediately following a date, it is a served license. For example:

```
...  
INCREMENT SUP_ENTSRVR SYBASE 2011.11150 permanent 10 \  
    VENDOR_STRING=PE=EE;LT=CP HOSTID=000c29d300bd PLATFORMS="i86_n  
\  
...
```

4. Determine the product edition and license type for the license.

For both served and unserved licenses, note the value of PE (product edition) and LT (license type) in the line following the line you located in step 2. For example:

```
...  
INCREMENT SUP_ENTSRVR SYBASE 2011.11150 permanent uncounted \  
    VENDOR_STRING=PE=EE;LT=CP HOSTID=000c29d300bd PLATFORMS="i86_n  
\  
...
```

The PE value is the license product edition value; "EE" in the example above.

The LT value is the license type value; "CP" in the example above.

5. If you are installing Enterprise Edition, determine the number of client licenses.

If your license type is Development and Test (DT), you can change this number later.

- a) Locate the uncommented line, beginning with INCREMENT SUP_ENTCLIENT.

For example:

```
INCREMENT SUP_ENTCLIENT SYBASE 2011.11150 permanent uncounted \
    VENDOR_STRING=PE=EE;LT=ST HOSTID=000c29d300bd
PLATFORMS="i86_n \
...
```

- b) Determine whether the client licenses are served or unserved.

If the line beginning with INCREMENT SUP_ENTCLIENT ends with "uncounted", the client licenses are unserved. For example:

```
INCREMENT SUP_ENTCLIENT SYBASE 2011.11150 permanent uncounted \
    VENDOR_STRING=PE=EE;LT=ST HOSTID=000c29d300bd
PLATFORMS="i86_n \
    x64_n" ISSUER="CO=Sybase,
Inc.;V=2.0;AS=A;MP=3120;CP=100;EGO=" \
...
```

If that line ends with a number immediately after a date, the client licenses are served. For example:

```
INCREMENT SUP_ENTCLIENT SYBASE 2011.11150 permanent 100 \
    VENDOR_STRING=PE=EE;LT=ST HOSTID=000c29d300bd
PLATFORMS="i86_n \
...
```

- c) Determine the number of client licenses.

For unserved client licenses, the number of client licenses is the value of CP two lines below the line beginning with INCREMENT SUP_ENTCLIENT. For example:

```
INCREMENT SUP_ENTCLIENT SYBASE 2011.11150 permanent uncounted \
    VENDOR_STRING=PE=EE;LT=ST HOSTID=000c29d300bd
PLATFORMS="i86_n \
    x64_n" ISSUER="CO=Sybase,
Inc.;V=2.0;AS=A;MP=3120;CP=100;EGO=" \
...
```

For served client licenses, the number of client licenses is the value at the end of the line beginning with INCREMENT SUP_ENTCLIENT. For example:

```
INCREMENT SUP_ENTCLIENT SYBASE 2011.11150 permanent 100 \
    VENDOR_STRING=PE=EE;LT=ST HOSTID=000c29d300bd
PLATFORMS="i86_n \
...
```

Setting a Fixed Port Number for SySAM License Server

To use a served license when there are no ports available, you can edit the license file to specify a fixed port number.

The VENDOR SYBASE daemon is a license management process that runs on SAP Mobile Platform hosts. It normally uses a dynamically assigned port to communicate with the SySAM license server.

CHAPTER 1: Getting Started

This task is required when SAP Mobile Platform is deployed in an environment in which there are no available ports in the 27000 – 27009 range for the licensing server to dynamically assign.

1. Obtain a port number that SAP Mobile Platform can use to communicate with the SySAM license server.
2. Back up the license file.
3. Use a text editor to open the license file, and locate the `VENDOR SYBASE` line, near the top of the file.
4. At the end of that line, type `PORT=` followed by the port number you obtained.

For example, if you use port 27010, enter:

```
VENDOR SYBASE PORT=27010
```

You must leave a space between "SYBASE" and "PORT."

5. Save and close the license file.

CHAPTER 2 Installing SAP Mobile SDK

SAP Mobile SDK works closely with SAP Mobile Server, but its installation is independent of the SAP Mobile Server installation.

1. *Preparing for Installation*

Ensure that the host on which you are installing SAP Mobile SDK meets the system requirements and is ready for you to begin the installation.

2. *Entering License Information*

Start the SAP Mobile SDK installer and enter license information.

3. *Selecting Installation Options*

Specify the installation directory.

4. *Completing the Installation*

Review the installation summary and launch the installation process.

5. *Verifying the Installation*

After you complete the installation, verify that the installation was successful before proceeding.

6. *Protecting the Installation*

Proper functioning of the SAP Mobile WorkSpace component of SAP Mobile SDK is critically dependent on a specific version of Eclipse components.

Preparing for Installation

Ensure that the host on which you are installing SAP Mobile SDK meets the system requirements and is ready for you to begin the installation.

1. If User Access Control (UAC) is present in the version of Windows where you are installing, disable it. Restart the system for this change to take effect.
2. Verify that the target system meets the system requirements for SAP Mobile SDK.
See *Supported Hardware and Software*.
3. Verify that you have Administrator privileges on the installation target host.
4. If the JAVA_TOOL_OPTIONS environment variable is set, remove it before you start the installation.
Check for the JAVA_TOOL_OPTIONS environment variable in both User Variables and System Variables panes of the Environment Variables dialog.
 - a) Right-click **My Computer** and select **Properties**.

- b) Select the **Advanced** tab, then click **Environment Variables**.
 - c) Select `JAVA_TOOL_OPTIONS` and click **Delete**.
 - d) Click **OK** to exit all dialogs.
5. If there is an earlier version of SAP Mobile SDK installed on the same computer where you are installing this version, back up any `workspace` directory trees used by SAP Mobile WorkSpace.

Although installing this version of SAP Mobile SDK does not overwrite those files, you should create backups that have been opened only in the earlier version. If you open an earlier version project in the later version of SAP Mobile SDK, you are committed to migrating the project to the later version. Backups keep open the option to go back to the earlier version.

6. To use the sampleDB database that is packaged with SAP Mobile Platform, be sure that the SAP Mobile Server that you plan to connect to from your SAP Mobile SDK installation has the SAP Mobile Platform SampleDB service running.

If the Windows service that runs the sampleDB server, SAP Mobile Platform SampleDB, has not been created:

- a) In the file system where the SAP Mobile Server is installed, go to the `SMP_HOME\Servers\UnwiredServer\bin` directory, where `SMP_HOME` is the SAP Mobile Platform installation directory, down to the `MobilePlatform` folder.
- b) Run the `sampladb.bat` script.

To create the service to start automatically, enter:

```
sampledb.bat install auto
```

To create the service to be started manually, enter:

```
sampledb.bat install manual
```

- c) Start the SAP Mobile Platform SampleDB service you just created, either by using the Windows Services control panel, or by entering, at the command prompt:

```
sampledb.bat start
```

For more information on the **sampledb.bat** script, see *Create or Remove the Windows Service for sampledb Server (sampledb) Utility* in *System Administration*.

Entering License Information

Start the SAP Mobile SDK installer and enter license information.

1. Start the SAP Mobile SDK installer using one of these methods:
- Insert the SAP Mobile SDK installation media.
 - Install from a Web download. Extract all the archive files to the same temporary directory on a local disk, not a shared drive.

Note: Use a short path, preferably a folder directly below the root directory, such as `C:\temp`. The path can include only ASCII alphanumeric characters, underscore (`_`), hyphen (`-`), and period (`.`) (two consecutive period characters are not allowed).

- Install from an image on the network. You must access the installation image from a mapped drive, not a UNC path.

Browse to the location of the `setup.exe` file, and:

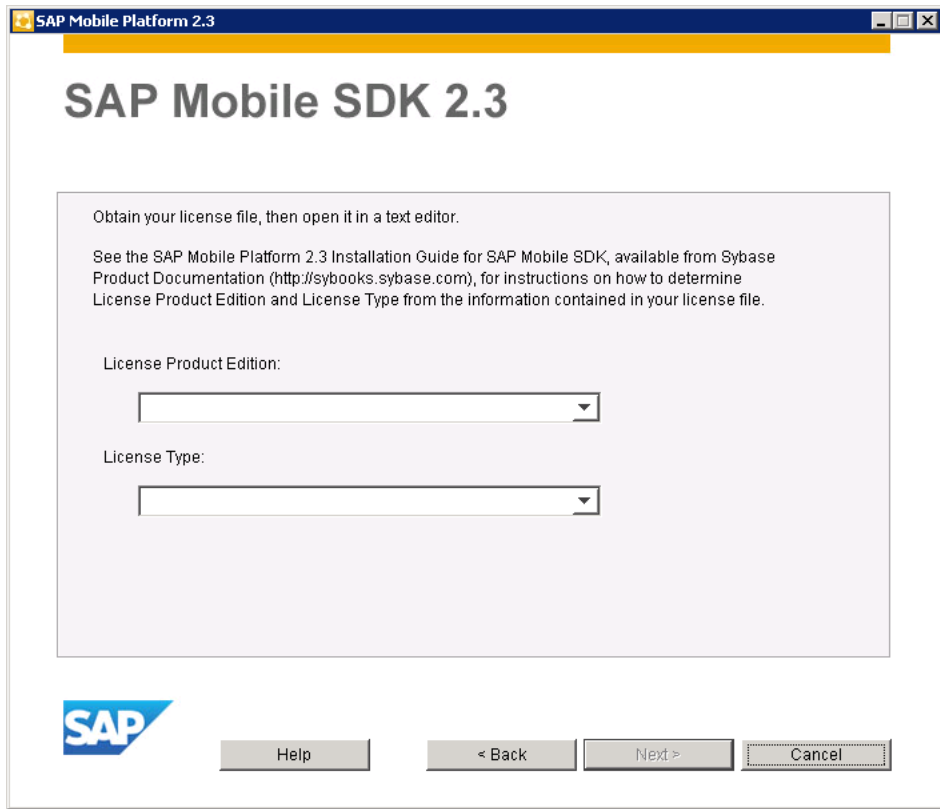
- On Windows 7 and Windows Server 2008 R2, right-click the `setup.exe` file and select **Run as Administrator**.
 - On earlier supported versions of Windows, double-click the `setup.exe` file.
-

Note: The installer displays an "Initializing wizard..." message while it verifies the integrity of the installation image. It may take a few minutes for the first installer panel to display.

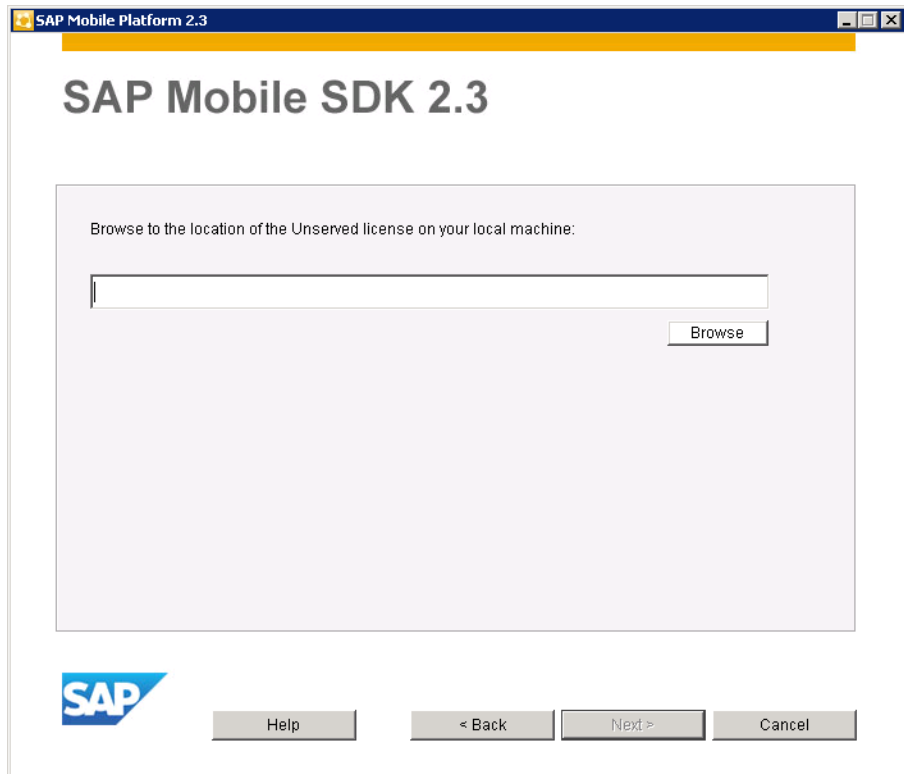
2. On the installer welcome page, click **Next**.
3. If you are installing on a system that does not have a licensed SAP Mobile Platform installation, select your license model and click **Next**.

Evaluation	Allows you to evaluate SAP Mobile Platform for 30 days. A license file is not required.
Unserviced (local) license	Standalone license managed locally by the host.
Served (remote) license	Standalone license managed by a license server.

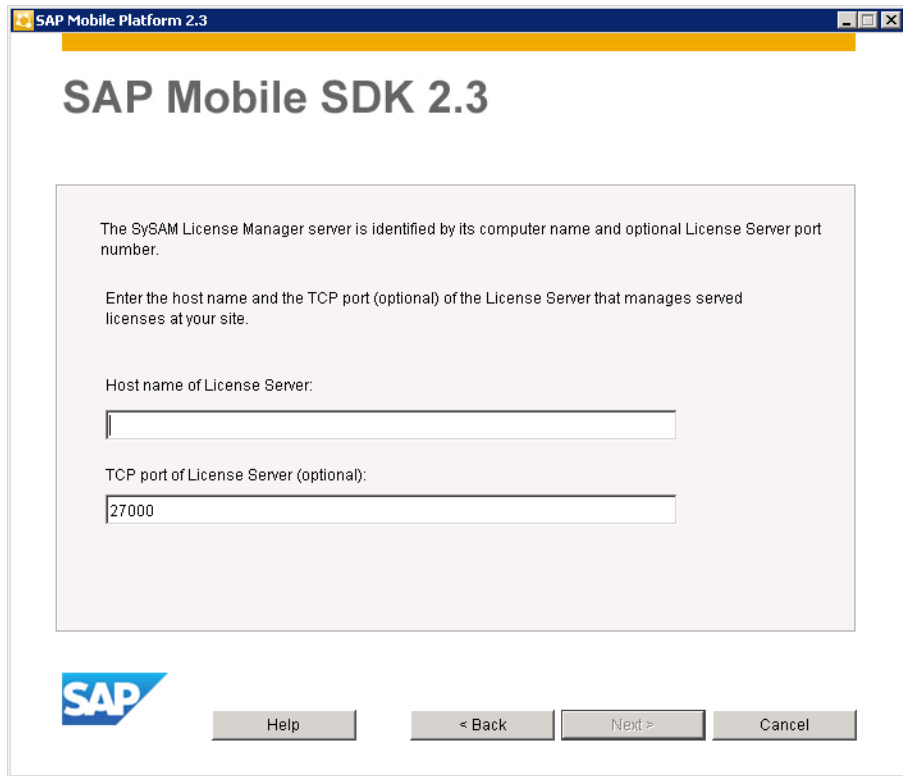
4. On the end-user license agreement page, select your country, accept the terms of the license agreement and click **Next**.
5. If you are installing on a system that does not have a licensed SAP Mobile Platform installation, on the license details page, select your License Product Edition and License Type.



6. If you are installing on a system that does not have a licensed SAP Mobile Platform installation, enter the location of your license file.
 - If you selected **Unserviced (local) license**, enter the absolute path to the license file on the installation target host. The path can include only ASCII characters.



- If you selected **Served (remote) license**, enter the Host Name of License Server and TCP Port of License Server.



If you get an error, such as `Failed to check out license`, click **OK**, then click **Back** to confirm the license model (served or unserved), license product edition, and license type.

Selecting Installation Options

Specify the installation directory.

1. Specify the absolute path to the installation location, which must reside on a local drive on the target host.

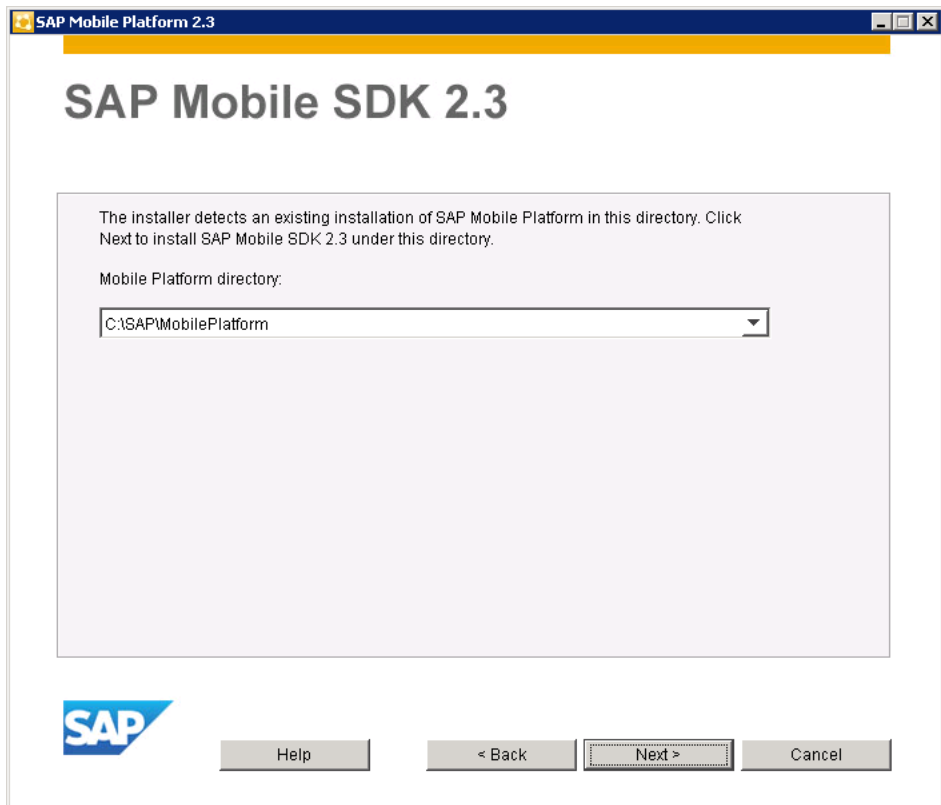
The total length of the path must be 30 characters or less.

Directory names in the path can contain only:

- ASCII alphanumeric characters
- Underscore (_), hyphen (-), and period (.) characters (two consecutive period characters are not allowed)

2. If you are prompted that an existing installation of SAP Mobile Platform has been detected:

- Click **Next** to install SAP Mobile SDK under that directory.
- If the path to that location would be too long (more than 30 characters), click **Browse** to select a different location, then click **Next**.



Completing the Installation

Review the installation summary and launch the installation process.

1. On the summary information page, verify the installation features and click **Install**.
2. (Optional) Click **View Release Bulletin**.
3. Click **Finish**.

You can now access this SAP Mobile SDK installation through the Windows Start menu: **Start > (All) Programs > SAP > Mobile Platform > Mobile WorkSpace 2.3**. If an earlier version of Sybase Mobile SDK was installed on the same system, its menu option remains: **Start > (All) Programs > Sybase > Sybase Mobile WorkSpace**, or ... **Sybase Mobile WorkSpace x.x**, where "x.x" is the earlier version number.

For information about configuring SAP Mobile SDK, see *SAP Mobile WorkSpace - Mobile Business Object Development*.

Verifying the Installation

After you complete the installation, verify that the installation was successful before proceeding.

See also

- *Protecting the Installation* on page 22

Verifying an Installation on Same Host as SAP Mobile Server

Start SAP Mobile WorkSpace and connect to SAP Mobile Server.

1. Start SAP Mobile WorkSpace.
 - a) From the Windows taskbar, select **Start > (All) Programs > SAP > Mobile Platform > Mobile WorkSpace 2.3**.
 - b) Accept the default workspace location.
2. Verify that the connection profile for the SAP Mobile Server to which you are connecting matches the settings for that server.

Change any connection profile settings as needed to match those for the target SAP Mobile Server.
3. Ping My SAP Mobile Server to be sure it is running:
 - a) In Enterprise Explorer, expand the **Mobile Servers** folder.
 - b) Right-click **My Mobile Server** and select **Ping**.
 - c) If you do not see a **Ping succeeded!** message, start the server (from Windows: **Start > (All) Programs > SAP > Mobile Platform > Start Mobile Platform Services** and ping the server again.

If you are connecting to a version 2.2.x Unwired Server, start the server (from Windows: **Start > (All) Programs > Sybase > Unwired Platform > Start Unwired Platform Services**) and ping the server again.
 - d) If the ping still fails, with exception `java.lang.ClassNotFoundException: com.sybase.jdbc3.jdbc.SybDriver`, see *Updating the Database Connection Profile* on page 91.
 - e) If the ping still fails, without a `ClassNotFoundException`, it is likely that port numbers were customized during SAP Mobile Server installation. In SAP Control Center, check the port numbers used for the SAP Mobile Server instance you are trying to connect to, and update the server's connection profile to match in SAP Mobile WorkSpace.

4. Connect to SAP Mobile Server:

In Enterprise Explorer, expand the **Mobile Servers** folder, right-click **My Mobile Server**, and click **Connect**.

5. Make sure sampleDB is installed on the SAP Mobile Server to which you connected.

See the last step in *Preparing for Installation* on page 13.

6. Connect to sampledb:

In Enterprise Explorer, expand **Database Connections**, right-click **My Sample Database**, and click **Connect**.

Verifying a Remote Installation

If you have installed SAP Mobile SDK on a system where no SAP Mobile Server is installed, start SAP Mobile WorkSpace and connect remotely to SAP Mobile Server.

Prerequisites

Ping the host name of the SAP Mobile Server you plan to use remotely with SAP Mobile SDK to be sure it is running.

Task

1. Start SAP Mobile WorkSpace:

- a) From the Windows task bar, select **Start > (All) Programs > SAP > Mobile Platform > Mobile WorkSpace 2.3**.
- b) Accept the default workspace location.

2. Modify the default connection profile for My Mobile Server:

- a) In Enterprise Explorer, expand the **Mobile Servers** folder.
- b) Right-click **My Mobile Server** and select **Properties**.
- c) From the left pane of the Properties for My SAP Mobile Server dialog, select **Mobile Server Connection Profile**.
- d) In the Host field, enter the host name for the SAP Mobile Server you want to connect to and click **Test Connection**.

If you do not see a `Ping succeeded!` message, check the spelling on the host name you entered; if you successfully pinged this server before starting this procedure, this ping should also succeed, if the same name is entered here.

- e) If the ping still fails, with exception `java.lang.ClassNotFoundException: com.sybase.jdbc3.jdbc.SybDriver`, see *Updating the Database Connection Profile* on page 91.
- f) If the ping still fails, without a `ClassNotFoundException`, port numbers were probably customized during SAP Mobile Server installation. In SAP Control Center, check the port numbers used for the SAP Mobile Server instance that you are trying to

connect to and update the server's connection profile to match in SAP Mobile WorkSpace.

g) Click **OK** to save the change.

3. Connect to SAP Mobile Server:

In Enterprise Explorer, expand **SAP Mobile Servers**, right-click **My SAP Mobile Server**, and click **Connect**.

4. Make sure sampleDB is installed on the SAP Mobile Server to which you connected.

See the last step in *Preparing for Installation* on page 13.

5. Modify the default connection profile for My Sample Database:

a) In Enterprise Explorer, right-click **My Sample Database** and select **Properties**.

b) In the left pane of the Properties for My Sample Database dialog, select **Sybase ASA Connection Profile**.

c) In the Host field, enter the host name for the SAP Mobile Server you want to connect to and click **Test Connection**.

If you do not see a `Ping succeeded!` message, check the spelling on the host name you entered; if you successfully pinged this server before starting this procedure, this ping should also succeed, if the same name is entered here.

d) Click **OK** to save the change.

6. Connect to sampledb.

In Enterprise Explorer, expand **Database Connections**, right-click **My Sample Database**, and click **Connect**.

If the ping still fails, with exception `java.lang.ClassNotFoundException: com.sybase.jdbc3.jdbc.SybDriver`, see *Updating the Database Connection Profile* on page 91.

Protecting the Installation

Proper functioning of the SAP Mobile WorkSpace component of SAP Mobile SDK is critically dependent on a specific version of Eclipse components.

In each version of SAP Mobile SDK, SAP Mobile WorkSpace uses specific versions of Eclipse components. These versions have been tested with SAP Mobile WorkSpace.

Note: Do not use the Eclipse update site to update any Eclipse plug-ins or contributed plug-ins like GMP, GEF, DTPs, and so on.

Before you start SAP Mobile WorkSpace, review the information in *Chapter 3, Installing an Upgraded SAP Mobile SDK* on page 25. To preserve the integrity of existing Eclipse workspaces, point to a new Eclipse workspace the first time you start SAP Mobile WorkSpace.

See also

- *Verifying the Installation* on page 20

Installing an Upgraded SAP Mobile SDK

You can install version 2.3 of the SAP Mobile SDK on the same system as an earlier version of the SDK. The earlier version of the SDK is not upgraded and you can use both versions as needed.

If you specify the same installation directory as the earlier SDK version, the installer places the new version SDK installation in a subdirectory at the same level as the earlier SDK version, for example:

```
<MobileSDK_InstallDir>
  MobileSDK213
  MobileSDK22
  MobileSDK23
```

Maintaining more than one version of SAP Mobile SDK allows you to defer making application code changes until you are ready to do so, or until you need to use features that are available only in the newer SDK version.

- Continue using the earlier SDK version to make minor modifications in applications that do not require features that are only available in the later version.
- Migrate applications to the newer SDK version to use features that are available only in the later version. See *Developer Guide: Migrating to SAP Mobile SDK*.
- When you have finished migrating all your applications to the newer SDK version, you can remove the older version using the Windows Add/Remove Programs option on the Control Panel.

Installing the Agentry Components of the SAP Mobile SDK

Within the SAP Mobile SDK there are included installers for Agentry development tools, as a part of the overall SAP Mobile SDK. Each of these components is installed to a system separate from the SAP Mobile Platform runtime environment and are intended for developers, implementors, and administrators.

With the exception of the Agentry SAP Framework Foundation, all the components described here can be found in the SAP Mobile SDK. This SDK is available on the SAP Service Marketplace as a part of the resources for the SAP Mobile Platform.

The Agentry SAP Framework Foundation is an ABAP add-on installed to the SAP system to which the mobile application is to connect and with which it will synchronize data. Since this component is not installed to the same system as the other components of the Agentry development tools, but instead to the SAP system, it is provided as a separate download within the list of SAP Mobile Platform components on the SAP Service Marketplace website.

Each of these components is described briefly below in the context of the SAP Mobile SDK. For those familiar with the Agentry Mobile Platform provided previously by Syclo, this content will be familiar. However, there are certain differences related to installation and configuration which are called out in these descriptions. From a functional and developer experience standpoint, there have been no changes to Agentry-based application development.

For information on the SAP Mobile Platform runtime, including how Agentry operates within it, see information provided on configuration, clustering, and similar topics within the guides provided for the SAP Mobile Platform runtime environment.

Note: The resources discussed here are also available on DVD's, which match the download images available on SAP Service Marketplace. Instructions refer only to the Service Marketplace. If you are installing from DVD's, the same structure information is applicable.

Agentry Server

The Agentry Server provided in the SAP Mobile SDK is installed to a host system for use by developers. Licensing for this component is limited to 5 devices, per the license keys provided in the plain text file found within the SAP Mobile SDK. See the procedure "Installing the Agentry Server for Windows" for details on locating and using this licensing information. This Agentry Server installation is used for development work, providing the Agentry Server by itself and outside of the overall SAP Mobile Platform implementation. When installed as a

CHAPTER 4: Installing the Agentry Components of the SAP Mobile SDK

Production Server, it is used for deployment preparation of a mobile application and all its resources, prior to being deployed to the SAP Mobile Platform runtime environment.

The installation program itself provides the options Development Server and Production Server, which should be familiar if you have worked with the Agentry Mobile Platform versions 6.0.x or earlier. However, the Production Server option, which was previously used for runtime environments with live users, is now provided only to support development and migration processes.

Agentry Editor

The Agentry Editor is the primary development tool for mobile applications built using the Agentry archetype within the SAP Mobile Platform. This is provided as a plug-in to the Eclipse IDE. The proper version of Eclipse must be installed and configured first, followed by the installation of the plug-in to the Eclipse implementation. The Agentry Editor provides a point and click development environment, within which the mobile application can be built or an existing mobile application can be modified for specific implementation needs.

For those familiar with the Agentry Mobile Platform versions 6.0.x or earlier this procedure is the same, and the Agentry Editor user interface experience has not been modified. The development concepts and architecture have also not been affected with the integration to the SAP Mobile Platform.

Agentry Test Environment

The Agentry Test Environment provides an Agentry Client which runs within a test and monitoring interface. This interface provides tools to inspect data stored on the client device, including objects, transactions, complex tables, and data tables. Actions and transactions can be debugged as they are executed or instantiated. Client devices hardware features including barcode scanners and GPS units can be mimicked within the Agentry Test Environment to support full functional testing of client-side behavior of the mobile application. The Agentry Test Environment can also mimic various client device types and form factors.

For those familiar with the Agentry Mobile Platform versions 6.0.x or earlier the installation of the Agentry Test Environment and its usage have not been affected by the integration to the SAP Mobile Platform.

Agentry SAP Framework Foundation and Related Resources

The Agentry SAP Framework Foundation is an ABAP Add-On that is installed to the SAP system with which the mobile application is to connect and synchronize. This component is only used for SAP systems and other enterprise systems will not use this component. The Agentry SAP Framework Foundation provides the framework within which the needed synchronization support for the mobile application is implemented. This includes exchange data model components such as triggers, mobile data objects, rules, filters, and other similar pieces.

This component of the SAP Mobile Platform is provided on SAP Service Marketplace as an available download. However, if the intention of your installation is to deploy a packaged

mobile application built on Agentry, this component should not be installed. Rather, the Agentry SAP Framework ABAP Add-On provided with that application, which will include application-specific components, should instead be installed. The Agentry SAP Framework Foundation is provided for user cases in which a new mobile application for an existing SAP system is being developed from the ground up.

Agentry SDK

The Agentry SDK installer is provided for two development scenarios. The first is when it is desired to develop ActiveX controls for use on Windows Mobile devices. The second is when it is necessary to develop interprocess communications between the Agentry Client and another process running on the mobile device. If neither of these scenarios are applicable to your environment, this component need not be installed. If the Agentry SDK is to be installed and used, see the installation procedure provided in the SAP Mobile SDK installation guide, as well as the information on the structure and usage of the API's within this SDK provided in the *Developer Guide: Agentry Applications*.

Installing the Agentry Server for Windows

Prerequisites

Address the following items prior to performing this procedure:

- Included in the SAP Mobile SDK 2.3 bundle is a text file named `DeveloperLicense_ReadMe.txt` containing a limited user license. This license information is needed for this installation process.
- Verify you have the proper installer for your target host system. There are two separate installers for the Agentry Server for Windows, which can be found in the SAP Mobile SDK 2.3 package:
 - `Agentry_6.1.0.x_Server_win32.exe` - For 32-bit builds of Windows
 - `Agentry_6.1.0.x_Server_x64.exe` - For 64-bit builds of Windows
- Establish connectivity between the host for the Agentry Server and the back end system or systems with which it is to connect. This includes network access, ODBC System DSNs, Oracle TNS names, account access for file system access, and any other system connectivity items related to any back end system to which the Server will connect.
- Record any connectivity information needed by the Agentry Server to connect to the back end system or systems. This can include user ID and password information, network addresses, network names or IP addresses, and port numbers or names.
- Place additional code libraries, Java resources, or similar items on the host and note their locations. These items are needed during the configuration of the Agentry Server.
- Determine if the Production Server, Development Server, or both are to be installed to the host system. This is necessary information required by the installer.

Task

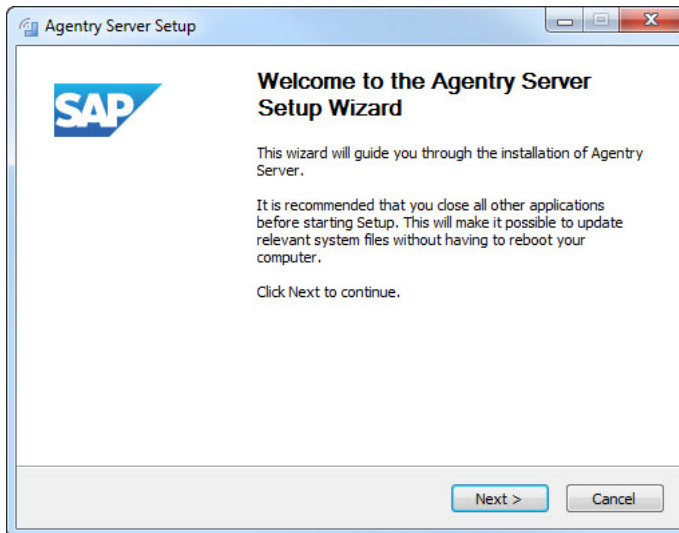
This procedure describes the steps to install the Agentry Server as provided in the SAP Mobile SDK 2.3 to a Windows host system. The installation wizard will prompt for information including licensing keys, the location to install the Agentry Server instance(s), and any desired shortcuts to the executable. The same installer can be used to install the Agentry Production Server, Agentry Development Server, or both. If both are selected, two separate instances of the Agentry Server will be installed.

Once the installation process is complete, there are several configuration tasks to accomplish specific to the implementation environment and mobile application needs. These are addressed separately from this installation procedure.

This server instance should never be installed to the same location as the SAP Mobile Platform. The purpose of the Agentry Server installer provided in the SAP Mobile SDK is to allow for development of mobile applications on Agentry without the need to install all of the SAP Mobile Platform components.

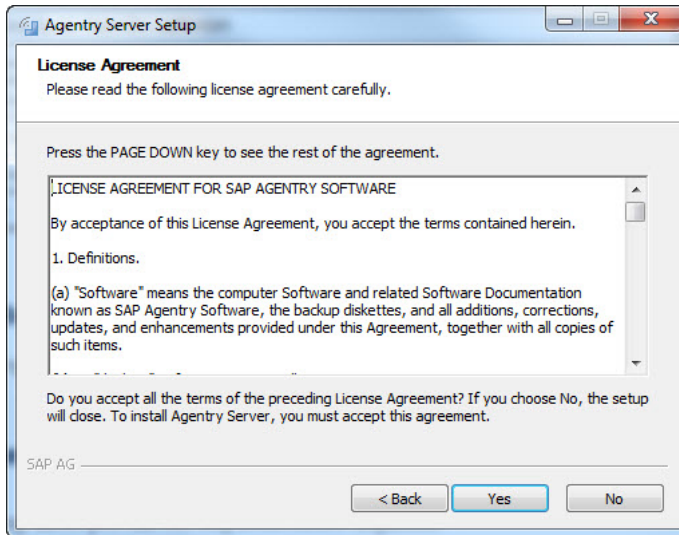
1. Launch the installer executable matching the needs of the host system; i.e. either the 32-bit or 64-bit Agentry Server installer.

The Welcome screen displays.



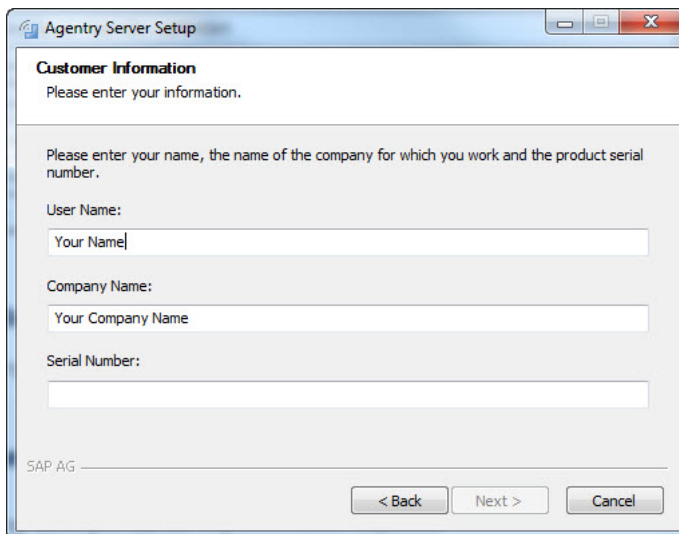
2. Click **[Next]** to begin the Agentry Server installation.

The License Agreement screen displays.



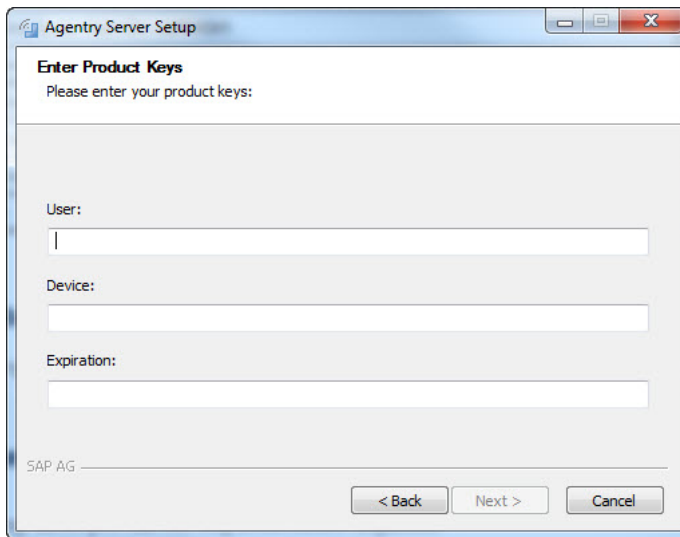
3. Review the license agreement text and click **[Yes]** to continue.

The Customer Information window displays.



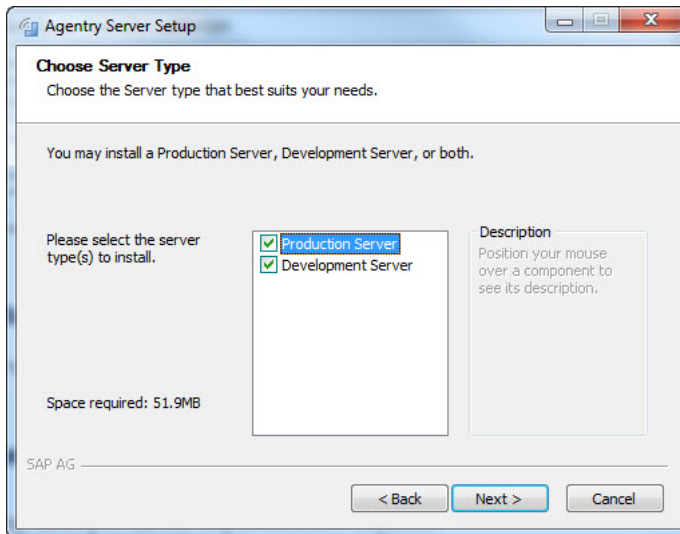
4. Enter the User Name of the administrator for the Agentry Server and the Company Name in the appropriate fields. In the Serial Number field, enter the serial number found in the license text file `DeveloperLicense_ReadMe.txt` provided with the SAP Mobile SDK 2.3. You will not be able to proceed until a valid serial number has been entered. Click **[Next]** to continue.

The Enter Product Keys window displays.



5. Enter the User key, Device key, and Expiration keys list in the license text file `DeveloperLicense_ReadMe.txt` provided in the SAP Mobile SDK 2.3. Click **[Next]** to continue.

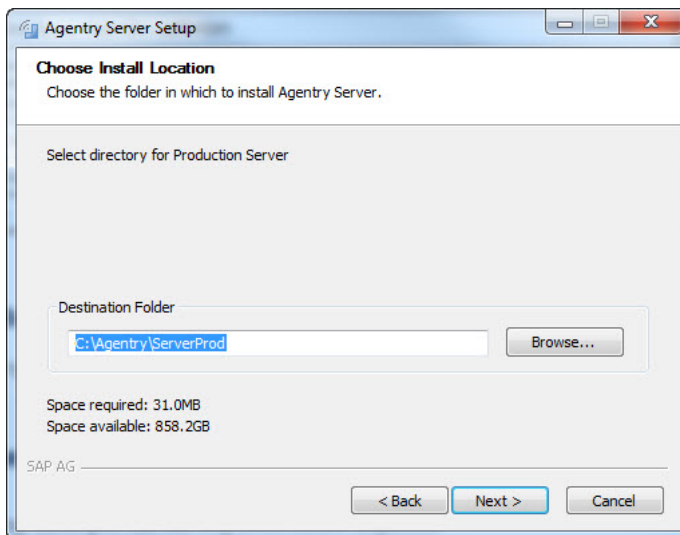
The Choose Server Type window displays.



6. Select the type of Server(s) to install. Select **Production Server** if it will be necessary to perform a production publish for the purposes of migration or upgrade. Otherwise, select **Development Server** for development work. Development is the more common selection and must be selected for development work, as a server configured for production should

not be used for development publishes. If both are selected, two separate Agentry Servers will be installed in different locations. Click **[Next >]** to continue.

The Choose Install Location window displays.



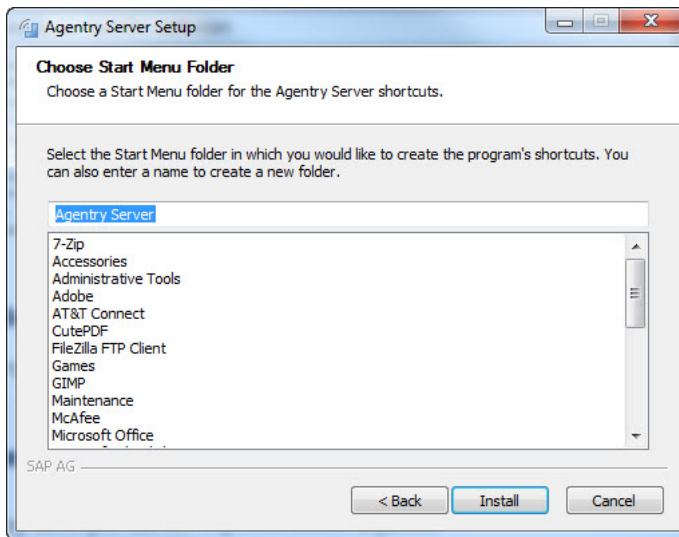
7. Specify the destination folder for the Agentry Server. To change the default folder, click **[Browse]** and navigate to the desired folder. Click **[Next]** to continue. At this point if both a product and development Server were selected for installation, the destination screen is displayed again. Select the destination for the development Server and click **[Next]** to continue.

The Shortcuts for [Production or Development] Server window displays.

8. You are prompted to add shortcuts to various locations of the Windows desktop. Make the desired selections and then click **[Next]** to continue.

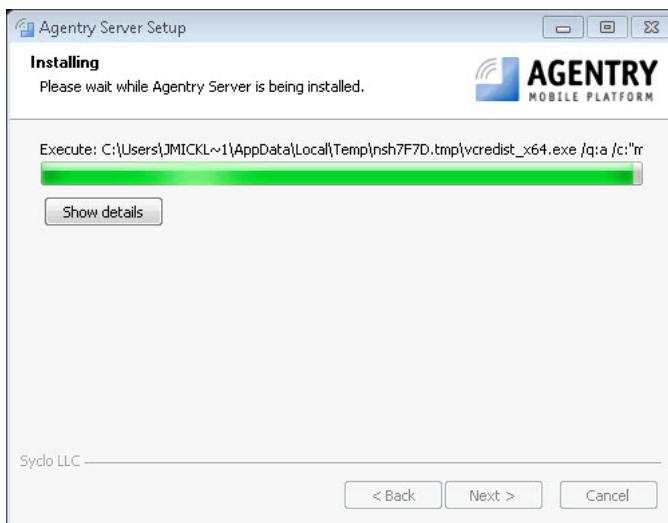
Note: This step and the previous step are repeated if you opted to install both a production and a development Server. Repeat these steps as necessary.

The Choose Start Menu Folder window displays.

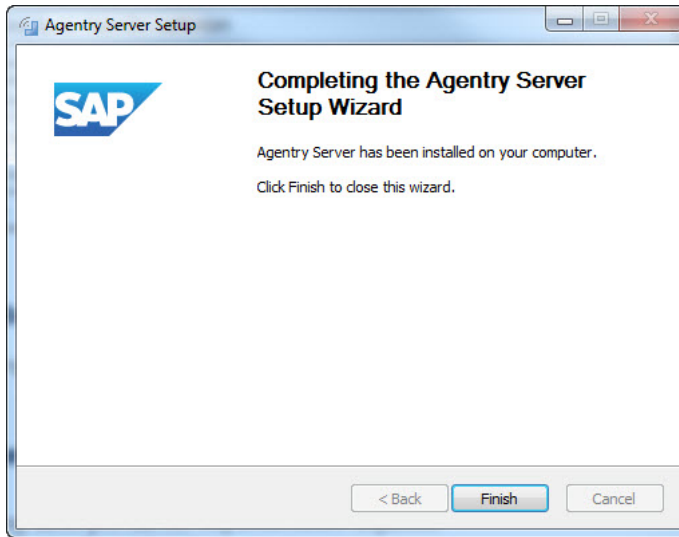


9. If you selected to create a shortcut in the Windows Start Menu, you are prompted for a folder where the shortcuts will be placed. You can select an existing folder or create a new one by entering the folder name in the field provided. If you did not select to create a shortcut in the Windows Start Menu, the previous screen is not displayed and the installation will begin. Otherwise, click **[Install]** to start the Server installation.

The Installation wizard installs the Agentry Server.



10. Click the **[Finish]** to complete the installation.



11. Depending on the security settings of the system to which the Server is installed, it may be necessary to modify how the Server and some of its related components are run; specifically they may need to be run with Administrator privileges. The following should be modified to run as an administrator if issues are encountered at startup:

- AgentryGUI.exe
- AgentryServer.exe
- AgentryCmd.exe

The Agentry Server is installed with the options selected within the installer. The installation may include one or two instances of the Agentry Server configured for production and/or development use. Additionally, certain command line utilities are installed into the Agentry Server's installation directory, as well as the Agentry Administration Client.

Next

Once the Agentry Server has been installed, address the following items:

- Configure the Agentry Server's connection options related to the back end systems with which it is to connect. These should match the system connections defined in the Agentry application project and can be created during a publish from the Agentry Editor.
- Configure the Client-Server connectivity options based on the specific needs of the implementation environment.

Installing the Agentry Editor Plug-In and Eclipse Platform

Prerequisites

The following items must be addressed prior to installing the Agentry Editor Eclipse plug-in:

- Determine if the proper installation for both Eclipse and the Agentry Editor plug-in is the 32-bit or 64-bit build.
- Access to the SAP Mobile SDK 2.3 package, as provided on the SAP Service Marketplace.
- DO NOT install the Agentry Editor plug-in into the SAP Mobile WorkSpace, which is an Eclipse implementation. Rather, obtain the proper compressed archive of Eclipse 3.7.2 (Eclipse Indigo) from one of the following two locations:
 - **32-Bit Build:** [*Click Here for Eclipse 32-Bit Download*](#) or enter the following URL in a web browser: `http://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/indigo/SR2/eclipse-java-indigo-SR2-win32.zip`
 - **64-Bit Build:** [*Click Here for Eclipse 64-Bit Download*](#) or enter the following URL in a web browser: `http://www.eclipse.org/downloads/download.php?file=/eclipse/downloads/drops/R-3.7.2-201202080800/eclipse-SDK-3.7.2-win32-x86_64.zip`
- Log into the intended host system as a user with Administrative privileges.
- Internet access to obtain the Eclipse plug-in installers is required.

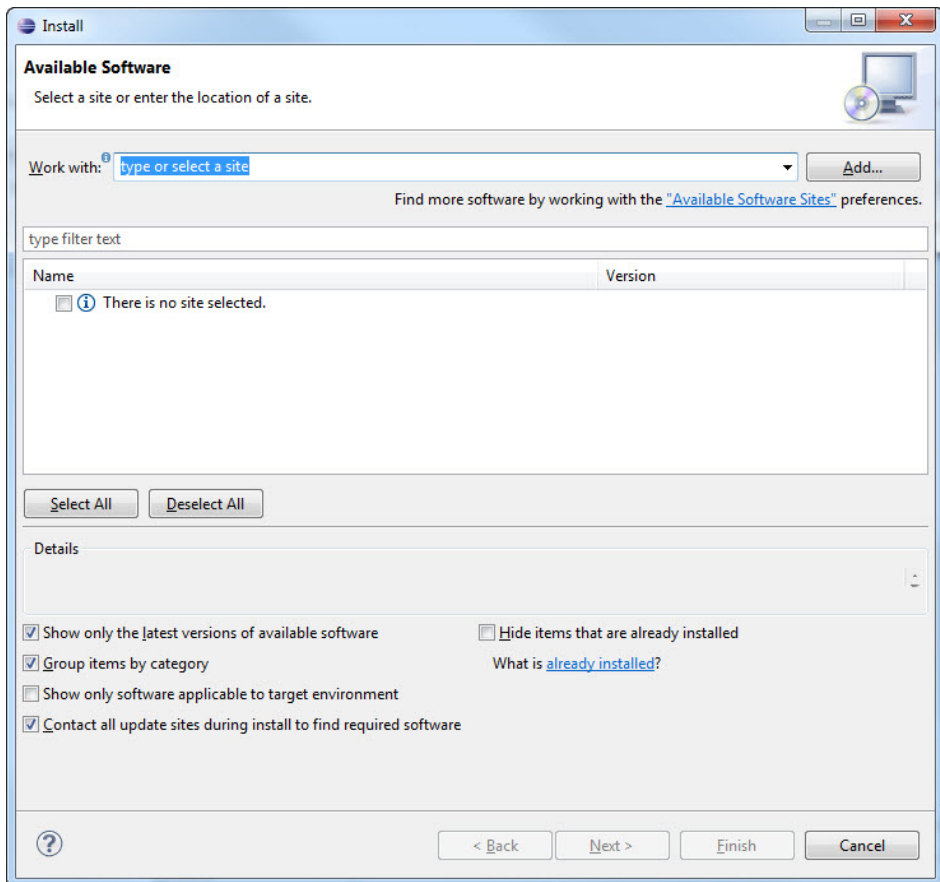
Task

The Agentry Editor is provided as a plug-in to Eclipse. This procedure provides the steps necessary to first install Eclipse and then to install the Agentry Editor to this Eclipse implementation. As noted in the prerequisites, the Eclipse installer should be obtained from the `eclipse.org` website, with the proper 32-bit or 64-bit build downloaded from that site. Eclipse is provided in a ZIP archive which is extracted to the desired installation location.

Once this is complete, the Agentry Editor plug-in can be added to the Eclipse implementation according to the instructions provided here. The build of the Agentry Editor plug-in must match the Eclipse build, and both 32-bit and 64-bit Agentry Editor builds are available.

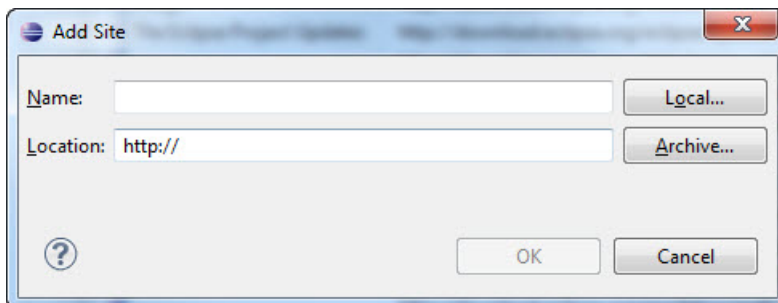
1. The Agentry Editor Eclipse plug-in is bundled within the SAP Mobile SDK 2.3 package provided on the SAP Service Marketplace. There are two files available and the one corresponding to either a 32-bit or 64-bit installation should be used for this procedure (as listed below). Make the ZIP archive available on the system where Eclipse is to be installed:

- **32-Bit:** Agentry_6.1.0.x_EditorPlugin_x86.zip
 - **64-Bit:** Agentry_6.1.0.x_EditorPlugin_x86_64.zip
2. Navigate in a web browser to one of the two URL's provided in the prerequisites to install either the 32-bit or 64-bit build of the Eclipse platform. Version 3.7.2 ("Eclipse Indigo") must be installed. Installation of Eclipse involves extracting the contents of the archive to the desired installation location. No installer is run for this process.
 3. Install the Java Runtime Environment for Eclipse. This must be Java 6 (Java 7 is available, but Eclipse requires the JRE for Java 6). This can be installed to the host system using the default location for the JRE; alternately it can be installed to the same location as the Eclipse instance, in a directory named `jre`. For example, if Eclipse is installed to `C:\eclipse`, the JRE can be installed to `C:\eclipse\jre`. This is the recommended location, as installing the JRE here does not require any environment variables to be modified on the Windows host system, as Eclipse looks to this location first for the JRE. If installed to a different location, the paths for the `jre\bin` and `jre\lib` directories of Java installation must be added to the **Path** environment variable in Windows before attempting to start Eclipse.
 4. Start Eclipse by executing the `eclipse.exe` executable file found in the extracted folder.
 5. During startup you are prompted to select or create an Eclipse workspace. The proper selection is based on the following:
 - New Agentry Installation: Create a new workspace
 - Upgrading an Agentry 5.x or earlier installation: Create a new workspace
 - Upgrading an Agentry 6.0.x installation: Either select the existing workspace or create a new one and import the projects after the Agentry Editor plug-in has been installed.
 6. When Eclipse is running, select the menu item **Help | Install new Software...**
The Available Software dialog displays.



7. Click the **[Add]** button located to the right of the **Work with** field.

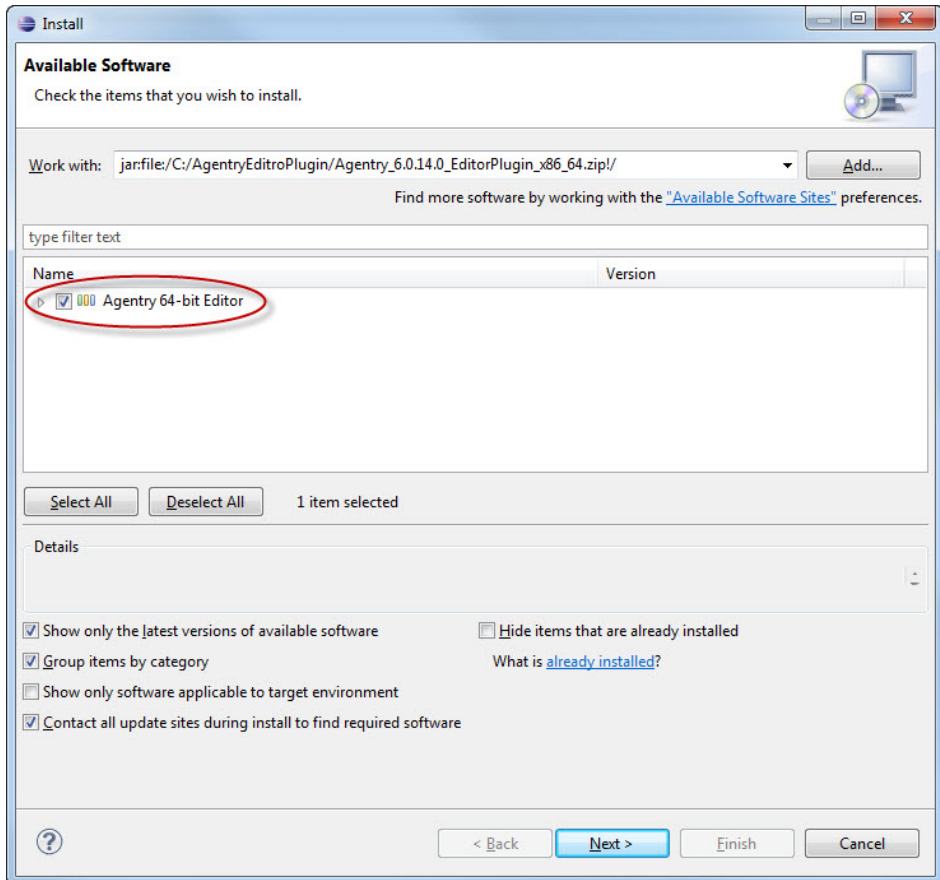
The Add Repository dialog displays.



8. Click the **[Archive]** button to the right of the Location field. In the file explorer window now displayed, navigate to and select the archive file for the Agency Editor plug-in from the first step in this procedure. Click **[OK]** after returning to the Add Repository screen.

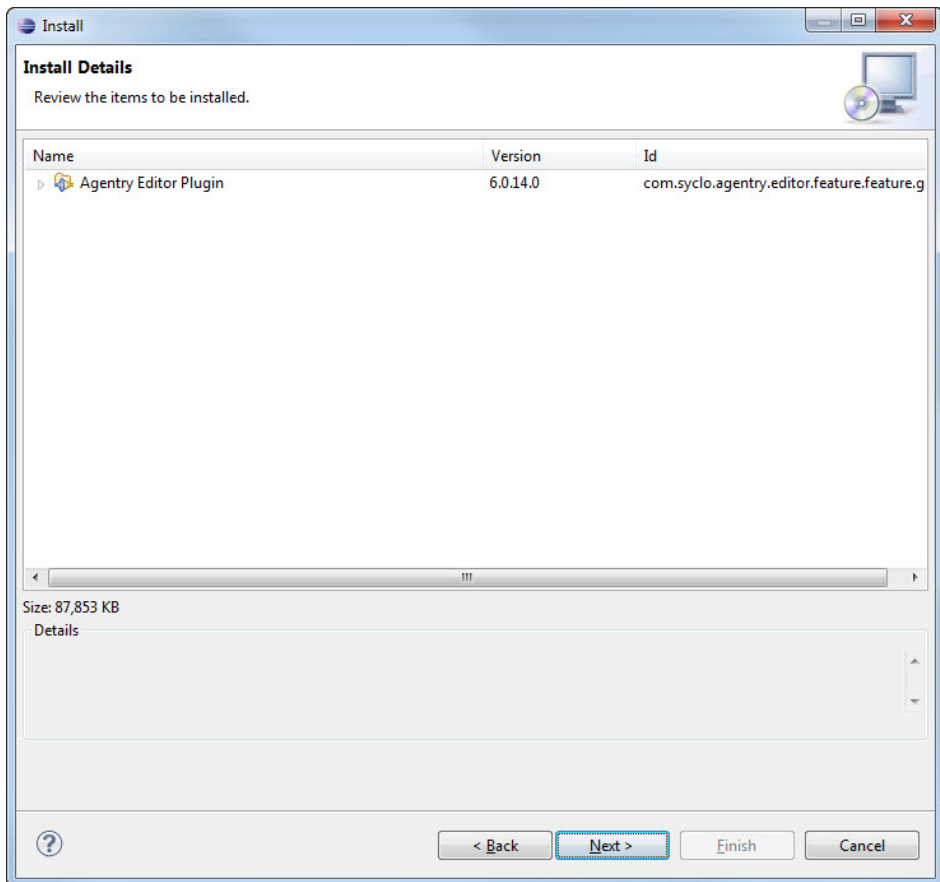
CHAPTER 4: Installing the Agency Components of the SAP Mobile SDK

This returns you to the Install dialog, where the Agency Editor plug-in (either 32-bit or 64-bit) is listed.



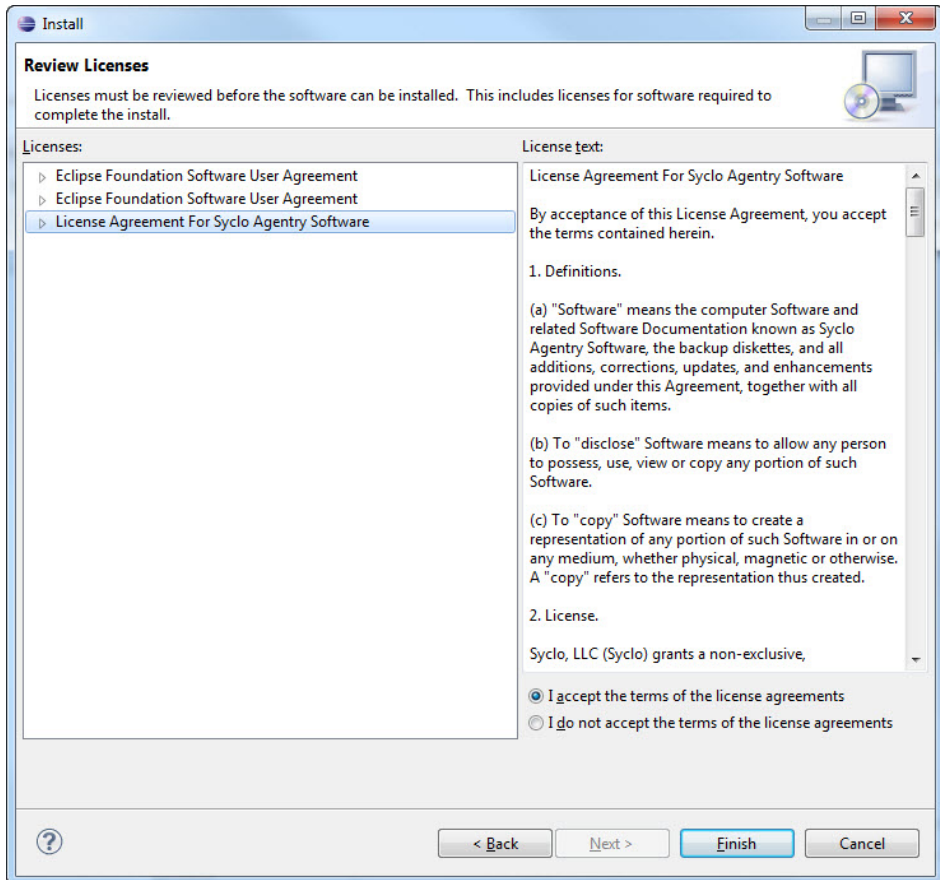
9. Check the box for the Agency Editor and click the **[Next >]** button.

The Install Details dialog displays.

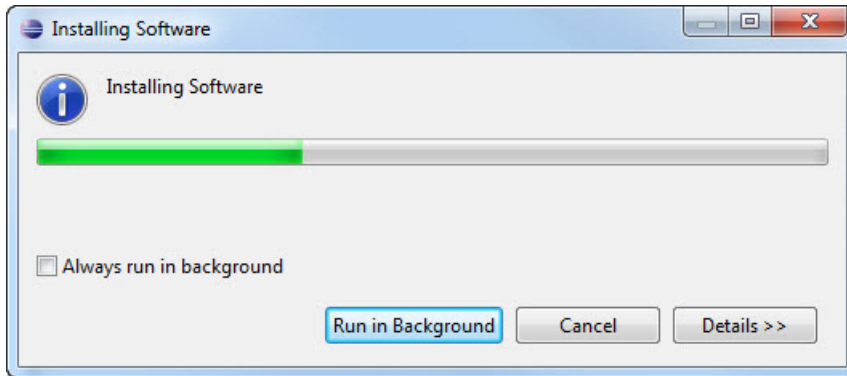


10. Click the **[Next >]** button to proceed.

The License Agreement dialog displays.

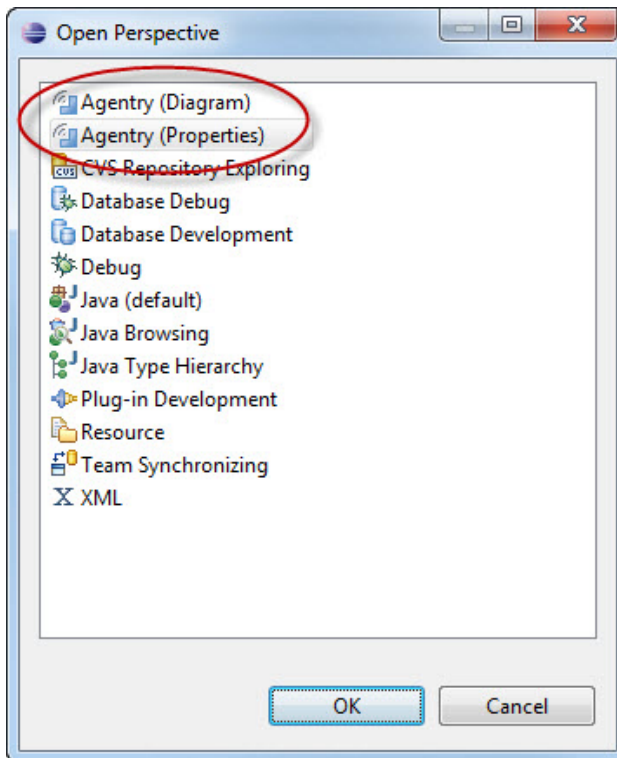


11. Select the radio button below the license information to accept the agreement and click the **[Finish]** button. During this process a security warning is displayed related to unsigned content. This is a result of certain `.jar` files not having been signed. You can safely continue with the installation by selecting the option to continue in the warning dialog. The installation will commence and a status dialog is displayed.



12. When the plug-in has been installed, a prompt is displayed to restart Eclipse. Select the **[Restart Now]** button. When Eclipse restarts, go to the workbench. Select the menu item **Window | Open Perspective | Other...**

The Open Perspective dialog displays, where the two Agentry perspectives are now listed.



13. Either of these may be selected to begin working with the Agentry Editor.

The Eclipse environment and Agentry Editor plug-in are installed on your system.

Next

Additional configuration of Eclipse is needed as it relates to the Agentry Editor plug-in. This configuration is performed within the Eclipse Platform and includes the following general items:

- Agentry projects work with several file types. You must create file associations within Eclipse for these types to allow the platform to properly handle, display, and edit them.
- Script files, including SQL and shell or batch scripts created in Agentry Editor are saved with a Unicode encoding. The default file encoding for an Eclipse workspace is different. You must modify file encoding options within the Eclipse preferences.
- When working with a database back end system, you must configure the Data Source Tools installed with Eclipse in order for the Agentry Editor connector studio to work with a database back end.

It may also be necessary to complete the configuration of one or more Agentry Servers. A publish from the Editor is a part of the Server configuration process as it relates to the Agentry application project's system connections. If connectivity was already configured between the Agentry Server and the back end system, the publish is not necessary for connection configuration.

Agentry Editor and Eclipse Platform Configuration Overview

After the Agentry Editor plug-in has been installed has been installed to Eclipse there are configuration tasks to be accomplished. All tasks are performed from within the Eclipse Platform itself. These tasks relate to several areas of the Agentry Editor and its management of application projects.

The information provided concerning these configuration tasks should be considered a recommended best practice. For a particular development environment, different options or configurations can be implemented based on need. Contributing factors and items of consideration include the overall uses for the Eclipse implementation.

Connector Studio and the Data Tools Platform

The Data Tools Platform (DTP) is a project for the Eclipse platform that provides several powerful tools for work with different data sources. The Agentry Editor leverages the power of this project by extending the tools provided in the Connectivity DTP subproject with the Agentry Connector Studio.

The Agentry Connector Studio is a tool within the Agentry Editor plug-in that allows for the development of object, transaction, and step definitions using the available schema information of a data source. When an Oracle or SQL Server database system is that data source there is some configuration needed within certain Connectivity components.

Specifically, connections must be configured to these database systems in order to access this schema information.

The configuration of a connection includes the creation of a Driver Definition and a Connection Profile within the Connectivity tools. The procedure provided on configuring these items will be an example on configuring these tools for the Agentry Editor. For complete information on all configuration options as well as descriptions on the functionality and uses for these tools see the Eclipse.org website.

Java Projects

If the application being developed or configured includes a Java Virtual Machine system connection, a Java project should be created within the Eclipse platform. This project should include the Agentry Java API version 5, as well as any other needed resources to properly build the Java logic for the mobile application's data synchronization. This will also expose the data members of the Java resources to be used to the Agentry Connector Studio. This will then allow for objects, transactions, synchronization steps, and properties to be defined based on the information provided by the Java packages in the Java project.

This process is performed using the Eclipse New Java Project wizard. The new Java project can be created before or after the Agentry application project is created, but does require the Agentry Development Server to be installed and accessible to the host system of the Eclipse platform, as this location includes the Agentry Java API JAR file to be used in the Java project.

File Associations and File Encoding

An Agentry application project can contain several different file types as a part of the project definitions. These can include scripts for SQL, batch files, and shell scripts, Java source files, bitmap files and large markup files related to XML processing. Configuration of the Eclipse Platform and the tools outside of, but still used by the Agentry Editor plug-in for Eclipse, can be needed in order for these different files to be handled correctly.

Also, most of the above mentioned files are stored as text files, with the obvious exception being bitmap files for image definitions. The encoding of these text files when created by the Agentry Editor is a unicode format of UTF-16. Files for other purposes outside of the Agentry application project may not be encoded in this format. Therefore the configuration related to file encoding must be performed in a manner that supports the different file encoding formats that may be needed.

Creating a Connection Profile for the Agentry Connector Studio

Prerequisites

The following items must be addressed prior to performing this procedure:

- The connection between the Data Source Tools and the target database uses JDBC. The desired JDBC drivers should be selected and the associated resource files (jar files or

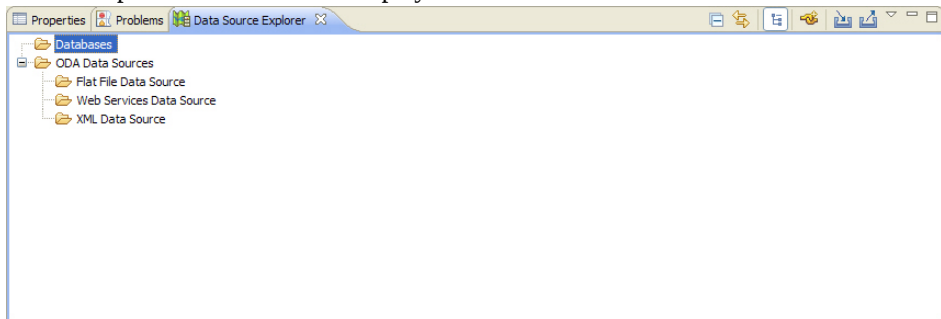
otherwise) should be located and noted. This information will be needed when creating a Driver Definition.

- If a suitable Driver Definition has already been created for the Data Source Tools, the name of the definition should be noted for use in the following procedure. In this case, a new Driver Definition will not need to be created.
- Information about the selected JDBC drivers should be gathered, specifically the attributes and values needed to configure a connection using the specific driver package. This will likely include the syntax for specifying the database host and server, login and password information, and the Java class or package within the JDBC driver to be used.

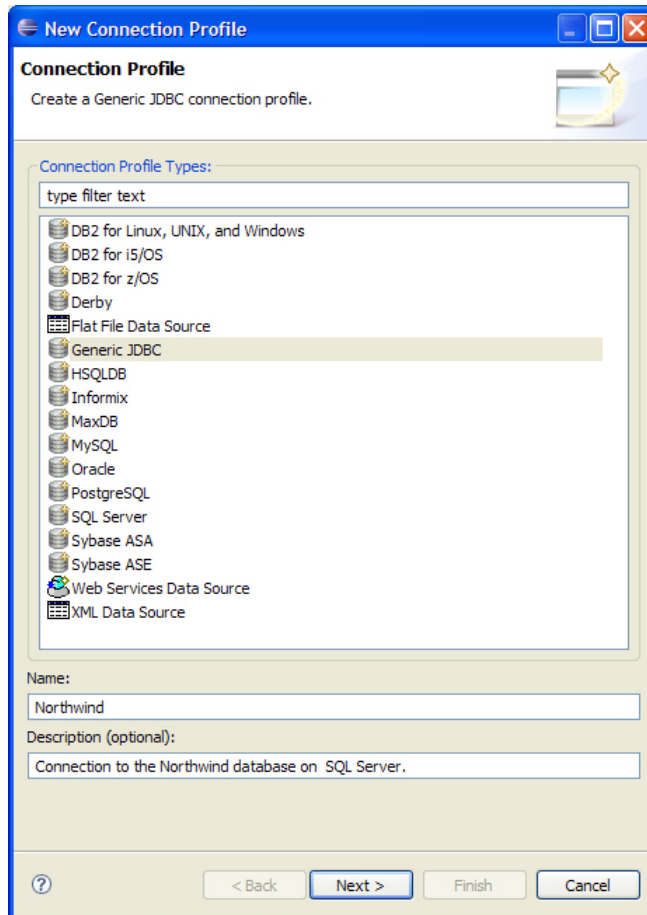
Task

The connection created in this process will be accessible through the Data Source Explorer view available in Eclipse. This view is not a part of the Agency Editor Eclipse Plug-in but rather is provided with the Eclipse Platform. Therefore, detailed instruction and information on its functionality and behavior is not provided by Syclo. This information can be obtained from the Eclipse.org website. Once this connection is created, it will be listed in the Data Source Explorer view. From here you will be able to perform all functionality supported by the Data Source Tools plus use the Connector Studio functionality provided with the Agency Editor.

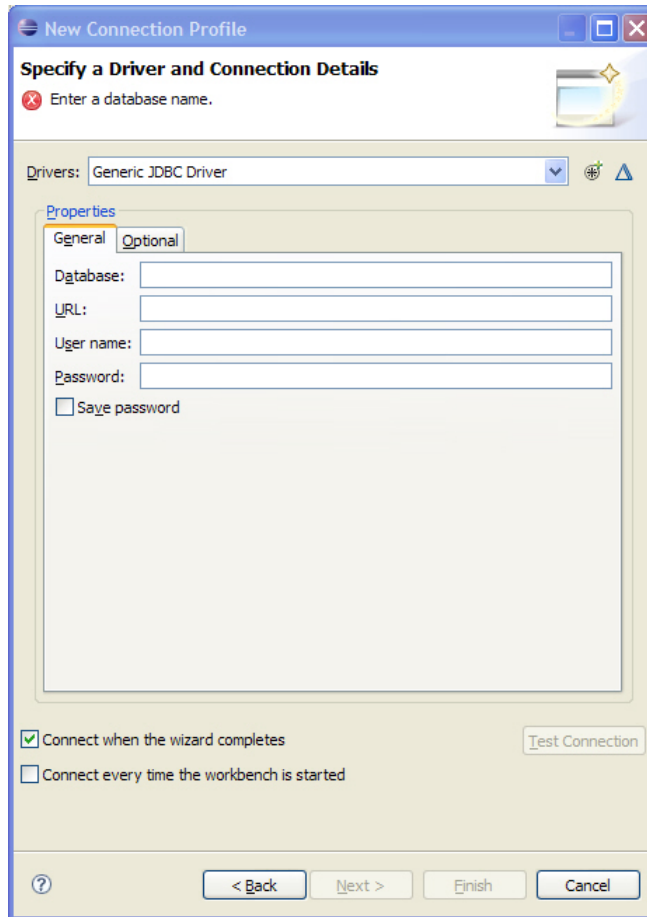
1. Open the Data Source Explorer view by selecting the menu item **Window | Show View | Other...**. In the tree control displayed, expand the Connectivity node and select the Data Source Explorer item. This will display the view as shown next:



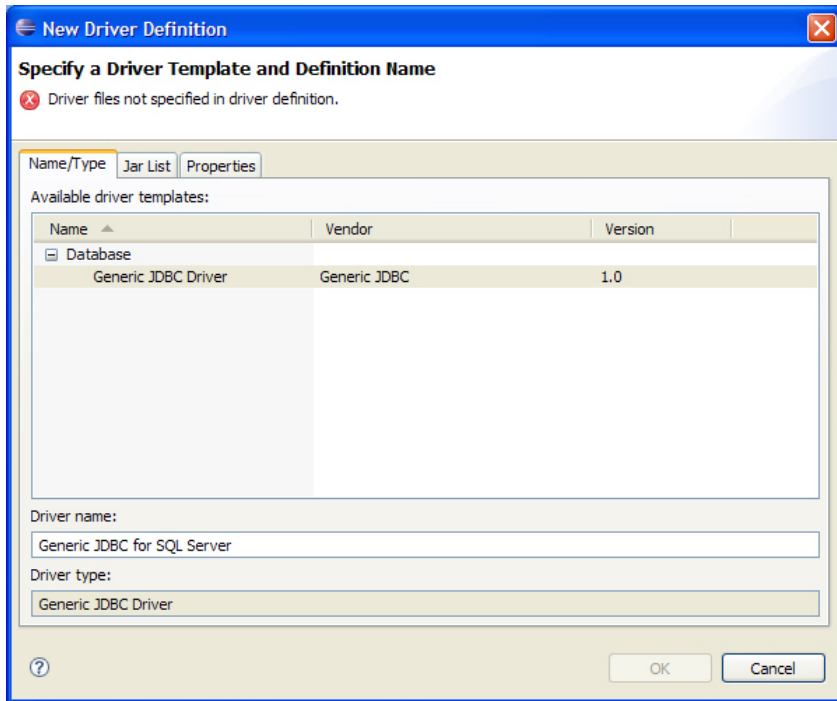
2. You can now create a new connection to a database using this view by either right-clicking the Databases node and selecting the menu item **New...**; or by clicking the tool bar button . Either will display the New Connection Profile Wizard.



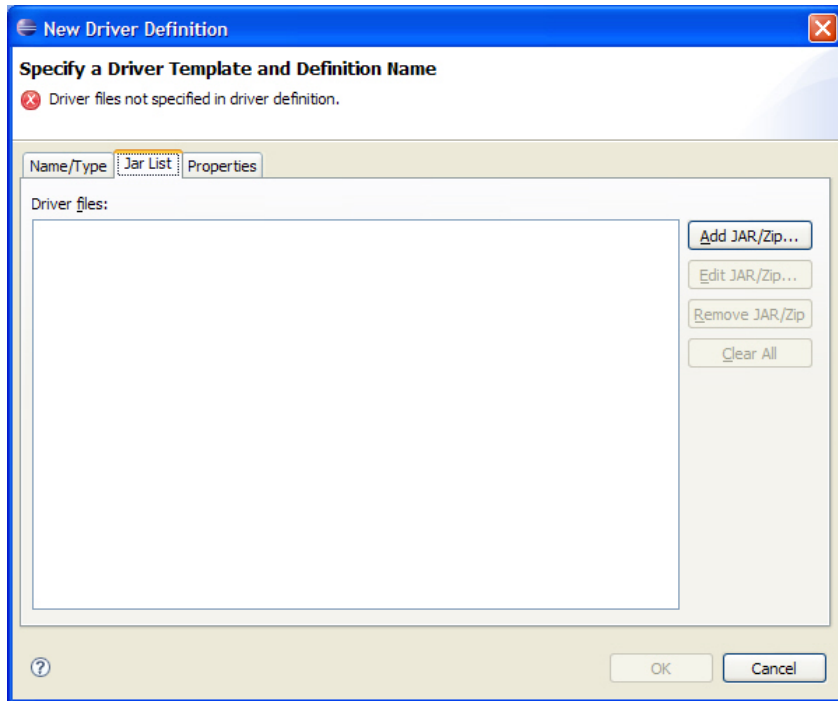
3. This screen lists the database types to which a connection can be made. This selection is largely based on not just the database type, but also on the selected JDBC driver package to be used. For this example, the Connection Profile Type to be used is Generic JDBC. Give the connection profile a name and a description and click the **[Next >]** button.



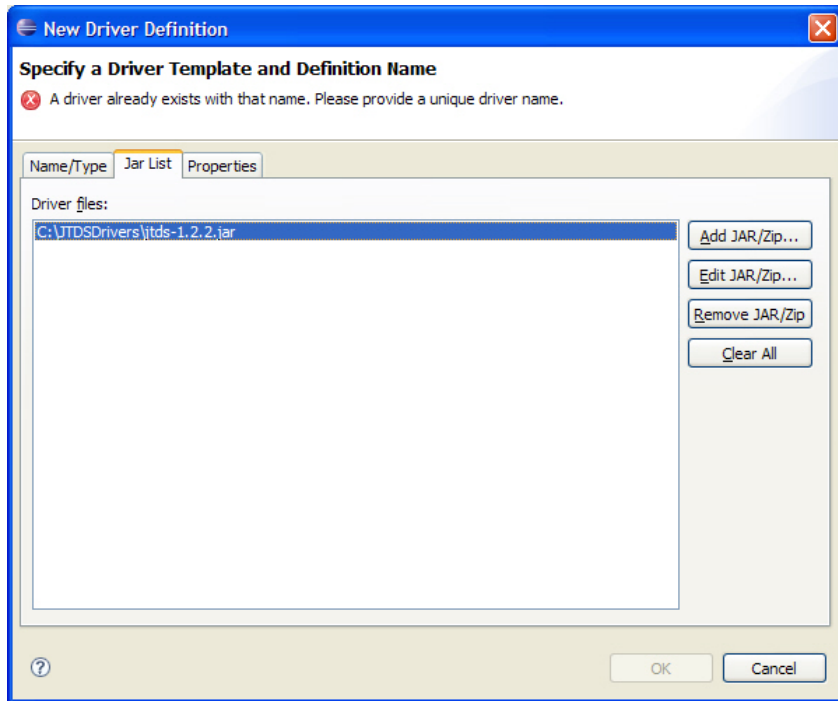
4. In this next screen the driver to be used for the connection is selected. If a driver profile exists that is suitable for the connection, select it from the Drivers list. In this case you can skip the next instructions on creating a new Driver profile.
5. To create a new Driver Profile, click the new Driver Profile Button  to the right of the Drivers list. This will display the New Driver Profile wizard.



6. In this first tab you can select the driver template. This list will vary depending on the type of Connection Profile you are creating. In this example there is only one option, Generic JDBC Driver. Enter a name for the Driver Definition.
7. Next select the Jar List tab. This list contains the jar files for the JDBC driver package to be used for this Driver Definition. For this example the list is empty initially.

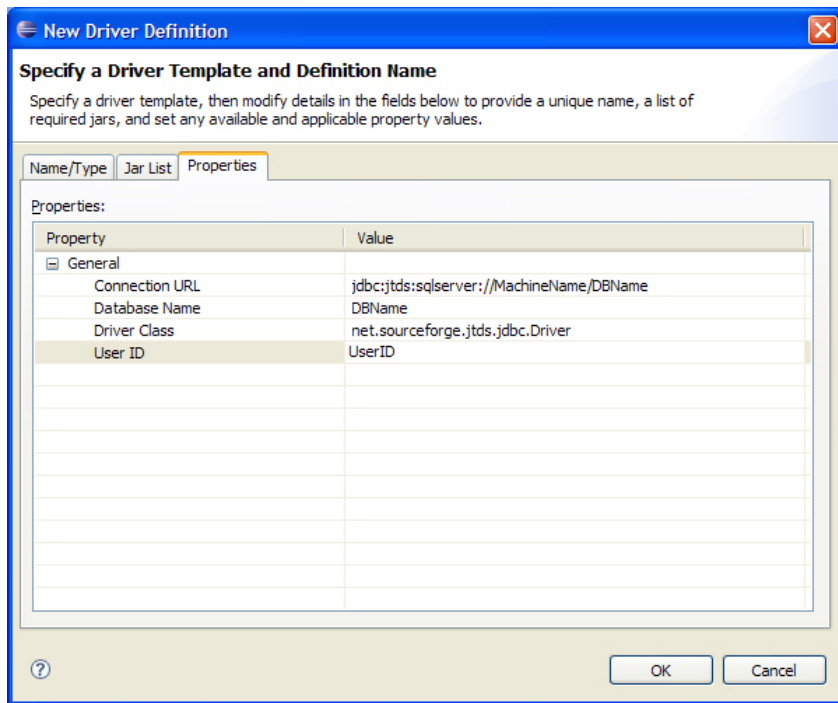


8. Add the proper jar file(s) to this list by selecting those provided with the selected JDBC driver package. This is done by clicking the Add Jar/Zip button and then browsing to the jar file on the file system.

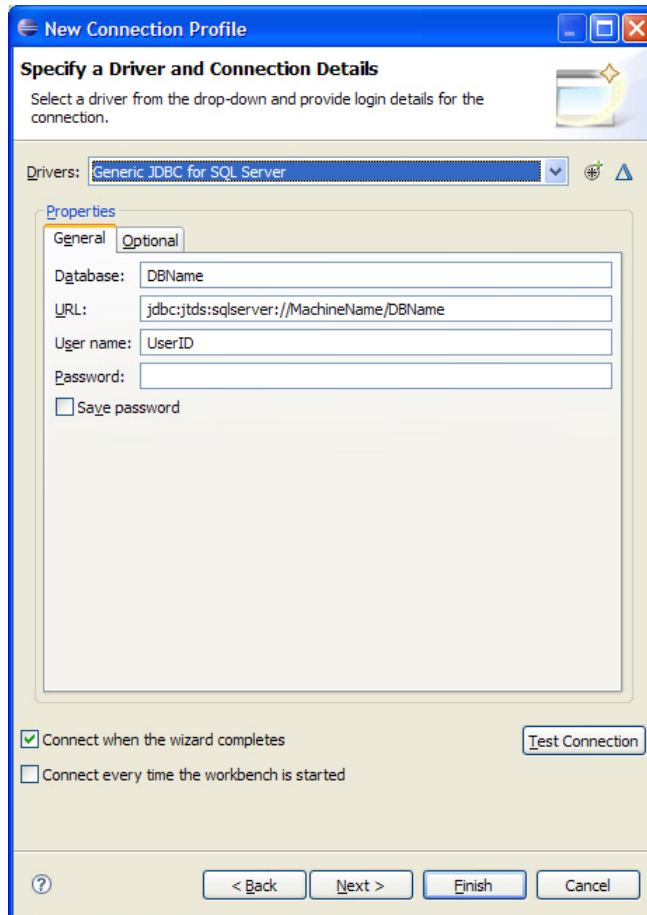


Here the selected item is the jar file for the jTDS 1.2.2 JDBC driver package provided by SourceForge.net. For other driver packages the selected jar file(s) will be different.

9. Next select the Properties tab. The items listed in this tab will be dependent on the selected jar file and are those items the JDBC driver requires to establish connections.

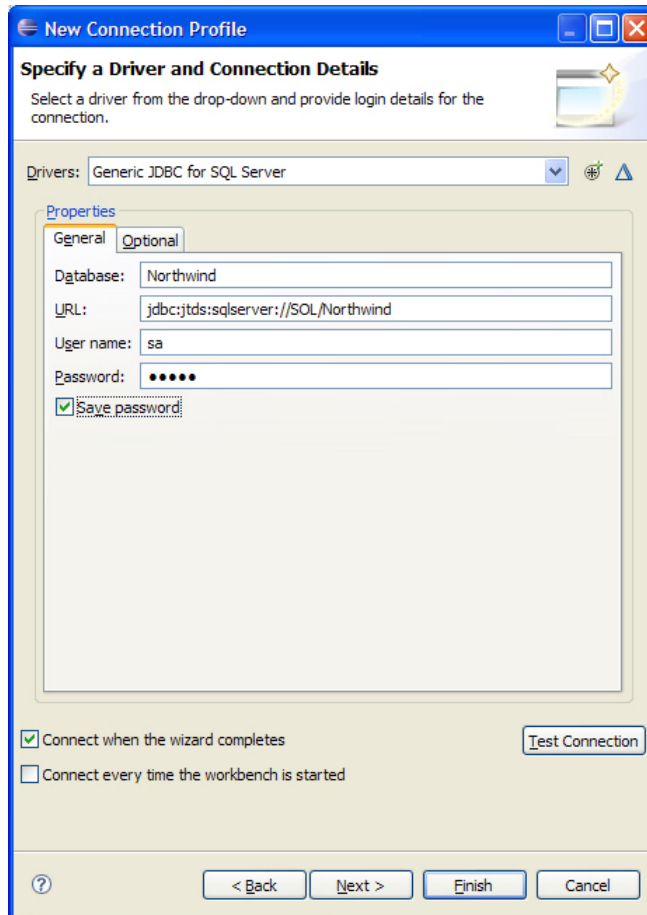


The values set here should be left to generic or template types of values. This driver definition, once created, can be reused for multiple actual connections to a database of the same type. The values entered here will be presented going forward when this driver definition is selected for a new Connection Profile and can be set at that time to the database specific values needed for a connection. Once these items are set, click the **[OK]** button.

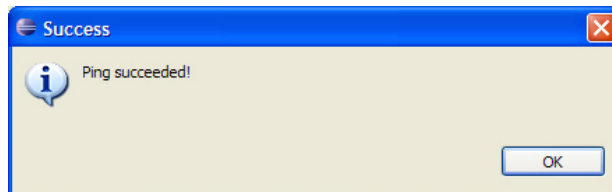


The New Connection Profile Wizard will now be displayed again. The values set in the fields on this screen will match those entered for the selected JDBC driver properties.

10. Now alter these settings by entering the proper values for the specific database connection to be created. In this example, the database name, URL, User name and Password are altered for a connection to a Northwind database in MS SQL Server.

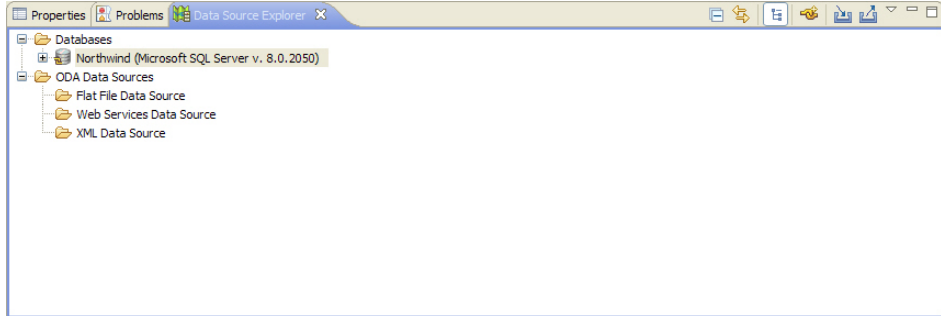


11. If the target database is available you can now test this connection by clicking the **[Test Connection]** button. If the connection is successful you will see the following pop-up screen.



12. Click OK on this prompt. Finally, choose whether to connect when this wizard closes and also whether or not to connect automatically when the Data Source Explorer view is displayed in Eclipse. Then click the **[Next >]** button to view a summary of this configuration, or the **[Finish]** button to close this wizard.

The Data Source Explorer View will now list the new Connection Profile under the Databases node with the name you entered.



The database connection created will allow for use of the schema information provided by that database in the object definition process. The developer can use the Agency Connector Studio, accessed through the Data Source Explorer view, to select a table and to create or modify an object definition, as well as its related transactions and SQL Step definitions, based on the structure of the database table.

Creating Java Projects for Agency Java API Development

Prerequisites

Before creating a Java project in Eclipse for use with Agency Java API development work, the following items must be addressed:

- The Agency Development Server with which development work will be performed must be installed and accessible to the host system of the Eclipse platform with the Agency Editor plug-in.
- Other Java resources needed for the development work should be accessible and available to the Eclipse platform host system. This can include other jar files as well as other code resources that may be needed for the development work.
- SAP Java Connector (SAP JCo) version 2.1.8, or any later 2.x version, is required for development and runtime environments in which the Agency SAP Framework Foundation is used. JCo version 3.0 or later are not supported at this time.
- Many mobile applications that make use of the Java Virtual Machine system connection type include multiple layers of Java code libraries that reside logically between the Agency Java API and the logic for the specific mobile application. In such situations these other libraries must exist in a location that is always available to the Eclipse platform.
- Determine the proper version of the JRE to be used for the Java development portion of the application. The Ganymede release of Eclipse requires the JRE version 1.6. The development project may need a different version of the JRE based on the requirements of the back end system with which synchronization will occur through the Java system

connection. A part of creating the Java project in Eclipse is the optional designation of the specific JRE version to use.

- The Java Perspective should be open in Eclipse. Note that by default this perspective is always open, though it can be closed at any time. To reopen this perspective, select the **Windows | Open Perspective | Other...** menu item in Eclipse. Select the Java (default) perspective in the list shown.

Task

This procedure provides the basic steps of creating a Java project to contain the development work performed with the Agentry Java API. When this procedure is complete a Java project will exist in the current Eclipse workspace that will include the resources needed for performing development work related to the Java Virtual Machine system connection type. This project will include:

- The Agentry Java API jar file as installed with the Agentry Development Server.
- A source folder location for Java logic created for the mobile application.
- Any other resources that may be needed for the project (e.g. `sapjco.jar`).

This procedure is not intended to be a comprehensive discussion on Java projects created in the Java Perspective in Eclipse, nor is it a discussion intended for the novice Java developer. A level of understanding and knowledge is assumed on the part of the reader related to these topics. Full information on the JDT project, which includes the Java Perspective, can be found at the Eclipse help site:

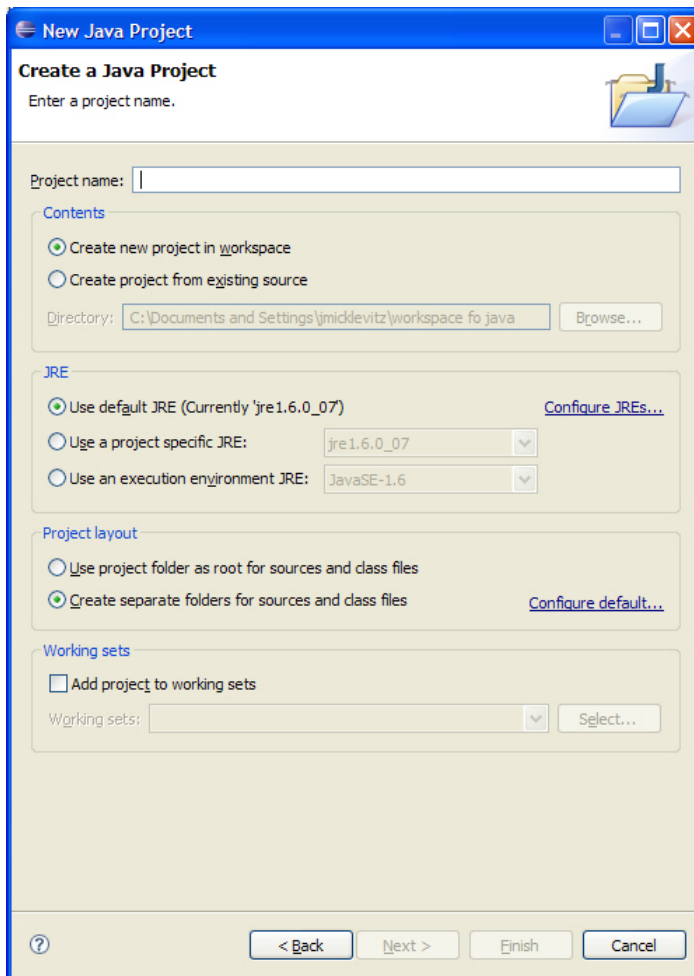
<http://help.eclipse.org/documentation>

At the above link see the documentation matching the Eclipse version you installed for the Agentry Editor plug-in.

1. Start the Java Project Wizard in Eclipse using one of the following methods:

- Select the menu item **File | New | Java Project**.
- Right click in the Package Explorer View in the Java Perspective and select the pop-up menu item **New | Java Project**.
- Right click in the Project Explorer View provided with the Eclipse Platform and select the popup menu item **New | Project**. In the screen displayed, select the tree control item **Java | Java Project**.

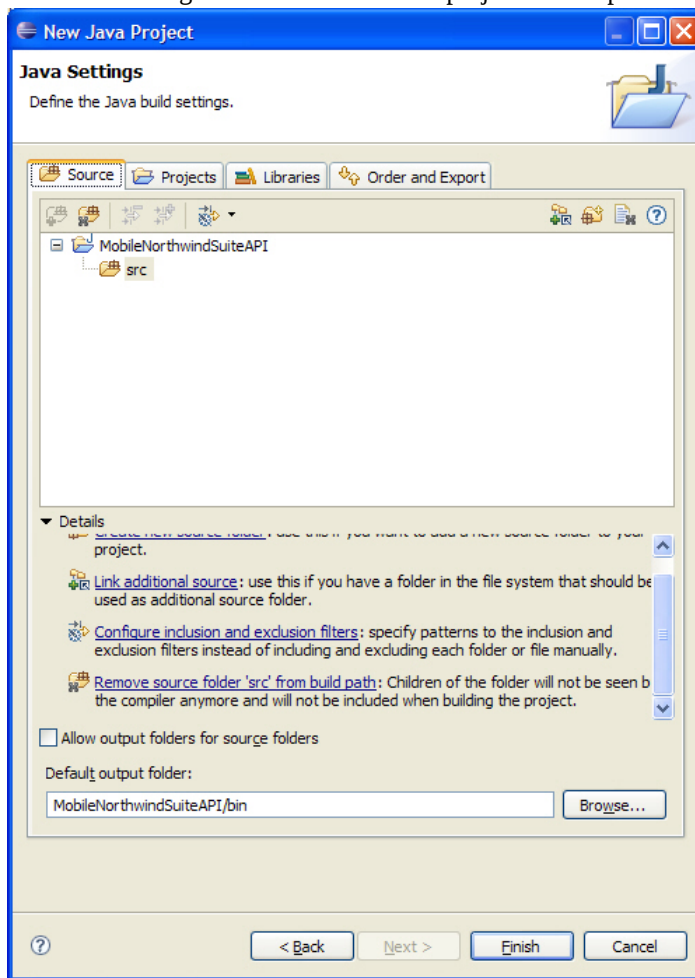
Any one of these actions will display the Java Project wizard:



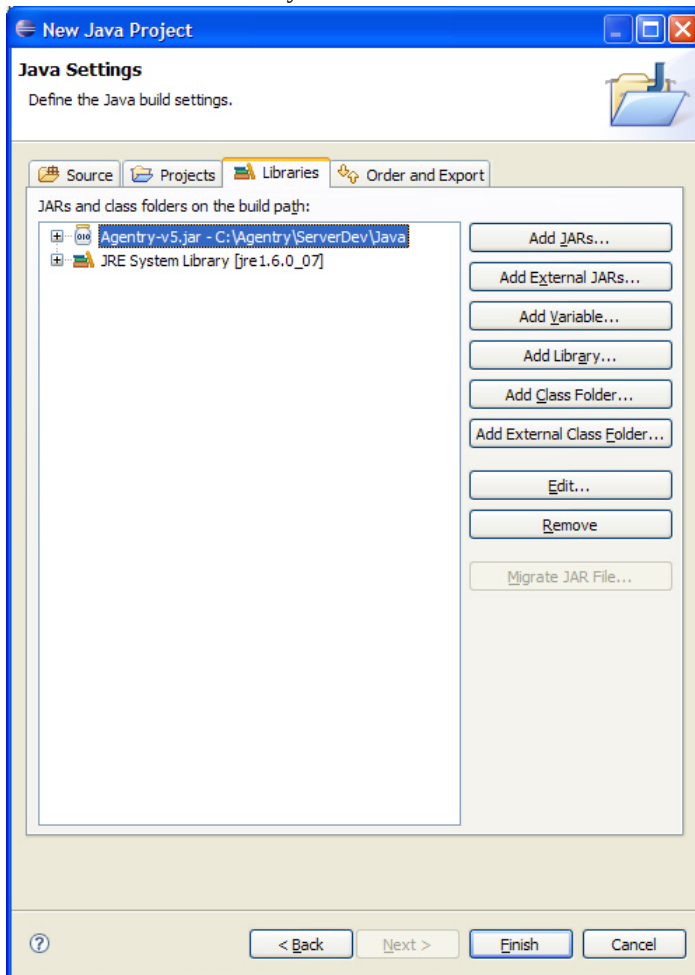
2. In this first wizard screen, begin by giving the project a name. This will be the identifier for the Java project in the Eclipse Project and Package Explorer views. Then:
 - a) Select the **Create new project in workspace** radio button in the Contents section to create a new project. Alternately, an existing source can be selected in the second radio button for this new project. This will create a project that references the specified source files in a location outside of the current Eclipse workspace.
 - b) In the JRE section specify the JRE version for the Java development project. This selection must match the needs of the back end system for which the mobile application development work will be performed. This need not be the same JRE version as Eclipse uses for its own execution. The version of the JRE to be used must be installed prior to making this selection. The third option is to select a Java execution environment.

- c) The Project Layout specifies the location of the source and built class files for the Java project. The default is the separate locations for source and output files. The default folders are `src` and `bin`, respectively, which will reside as sub-folders to the project folder in the Eclipse workspace. These locations can be modified by clicking the [Configure default](#) hyperlink in this section.
 - d) The new project can be added to a Working Set. If unfamiliar with work sets in Eclipse, see the help topic “Working Sets” in the Eclipse publication *Workbench User Guide*.
3. Once these options are set, click the **[Next >]** button to proceed.

The next screen of the Java Project wizard is displayed. This screen contains multiple tabs for further configuration of the new Java project’s build path:



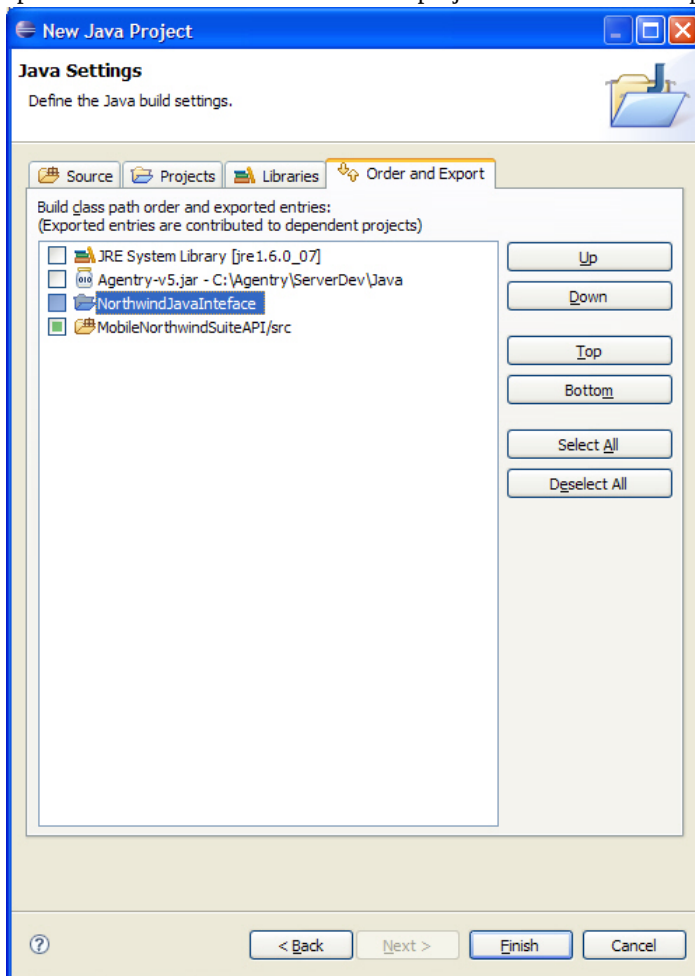
4. Set the Source options as needed for the given project. Options here include selecting additional source folders outside the workspace, adding new folders to the project within the workspace that will be used as source folders, and other source file-related tasks.
5. Select the Projects path to set other projects to be included on the build path for the current project. This project must reside in the same workspace as the new project being created. Once added, options exist related to which files or sources within the project are to be used. The build order for the project selected here is set in the Order and Export tab.
6. Select the Libraries tab. Here is where the libraries related to the Agentry Java API, as well as other libraries needed by the back end are selected and included in the project:



To add libraries to the project's build path, begin by clicking the button to the right to add the type of library needed. For the Agentry Java API, click the Add External JARs...

button. Navigate to the installation location of the Agentry Development Server and select the file `Java\Agentry-v5.jar`.

7. Once the libraries have been added, select the Order and Export tab. Here the build order is specified for the libraries and external projects for the new Java project:



The specific build order is completely dependent on the project being created. However, the JRE System Library will likely always be first. Within this order, the Agentry Java API library should be ordered before any libraries or projects that extend the Agentry classes or interfaces. Also, it is likely any resources from the back end system should be ordered before the mobile application-specific items, which may include “application suite” libraries, as it is likely these too will extend or access these items. Once the build order is set, click the **[Finish]** button to create the new Java project.

A new Java project will be created in the current Eclipse workspace. This project will now be accessible to the Agentry application project. When defining Java steps, or complex table or

data table Java synchronization components, it will be possible to select items contained in the Agentry Java API, as well as those in the other libraries and projects for the Java project.

Next

Changes to the project's configuration can be made by selecting the project in the Project Explorer or Package Explorer and select the Properties menu item.

Configuring Eclipse File Associations for Agentry Projects

Prerequisites

The following items must be addressed before performing this procedure:

- For Eclipse implementations to which the Agentry Editor plug-in was added, review the current file associations before making any modifications. Any current configuration of file associations for file types of `.sql`, `.bmp`, `.bat`, and `.sh` should be noted.
- Determine the desired editor within Eclipse to use for each of the file types that may be a part of the Agentry application project. This may be done now or during this configuration procedure.

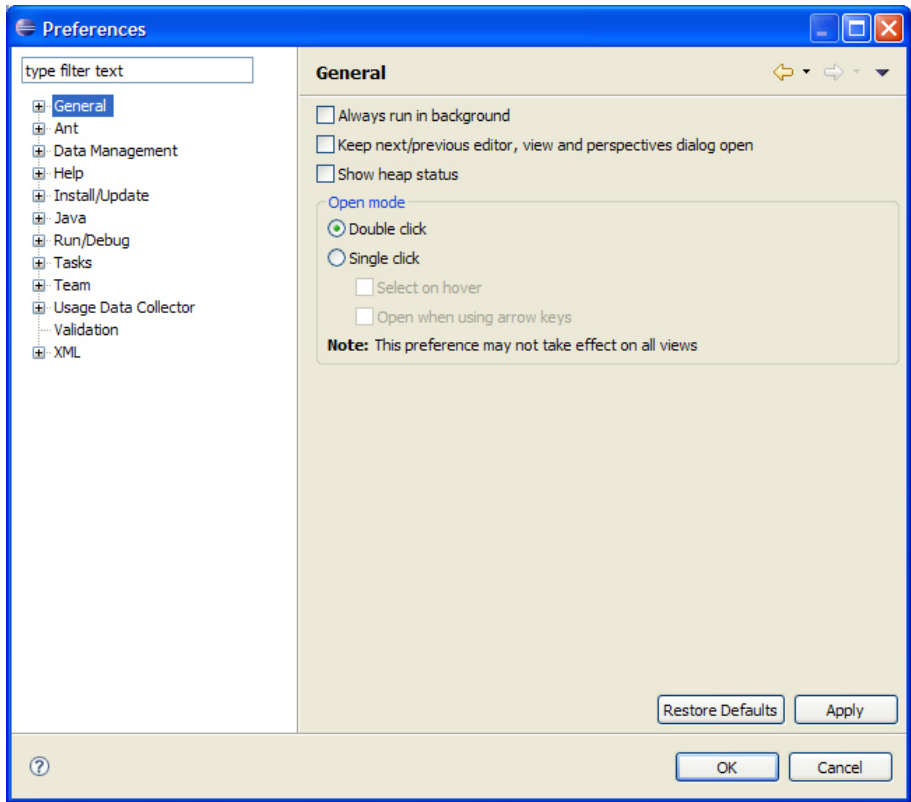
Task

In this procedure the file associations within Eclipse will be configured for the file types that may be a part of an Agentry application project. These file associations will determine which editor or view within Eclipse will be used to display the file contents and allow those files to be edited. The file associations configured in this procedure are those available in Eclipse as provided by the Agentry Editor installer. If there are other tools available for a particular implementation, those may be used as preferred or desired. The process for creating these file associations is the same. The selection of the particular tool within Eclipse for a particular file type is dependent on those available and preferred. Eclipse must be running to make these modifications.

Specifically the editors for bitmaps (`.bmp`), Windows batch files (`.bat`), SQL scripts (`.sql`), and Linux and Unix shell scripts (`.sh` for Agentry application purposes) will be configured in this procedure. For shell scripts and batch files the text editor provided with Eclipse will be configured in this procedure. Other tools exist for these types of files as plug-ins to the Eclipse Platform. If those are to be used, review the documentation and instructions provided with those tools for configuration.

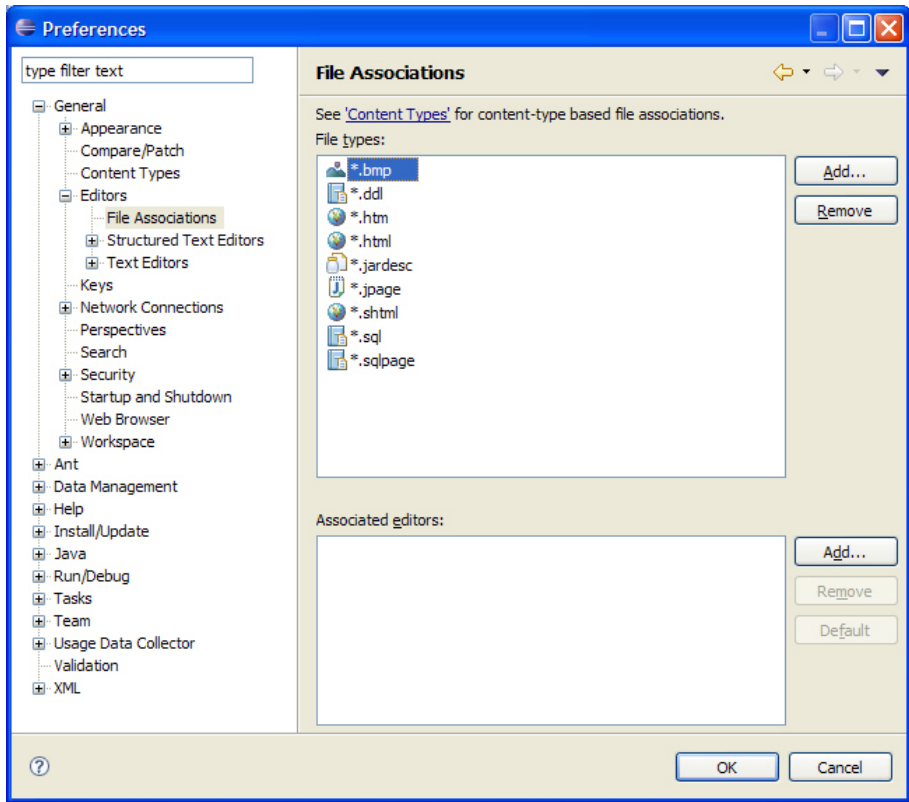
1. Select the Eclipse menu option **Window | Preferences**.

This will display the Preferences screen in Eclipse:



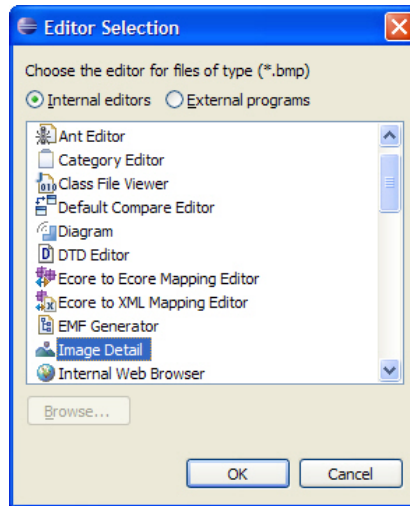
2. In the tree control on the left, select General | Editors | File Associations.

This will display the File Associations preference page:



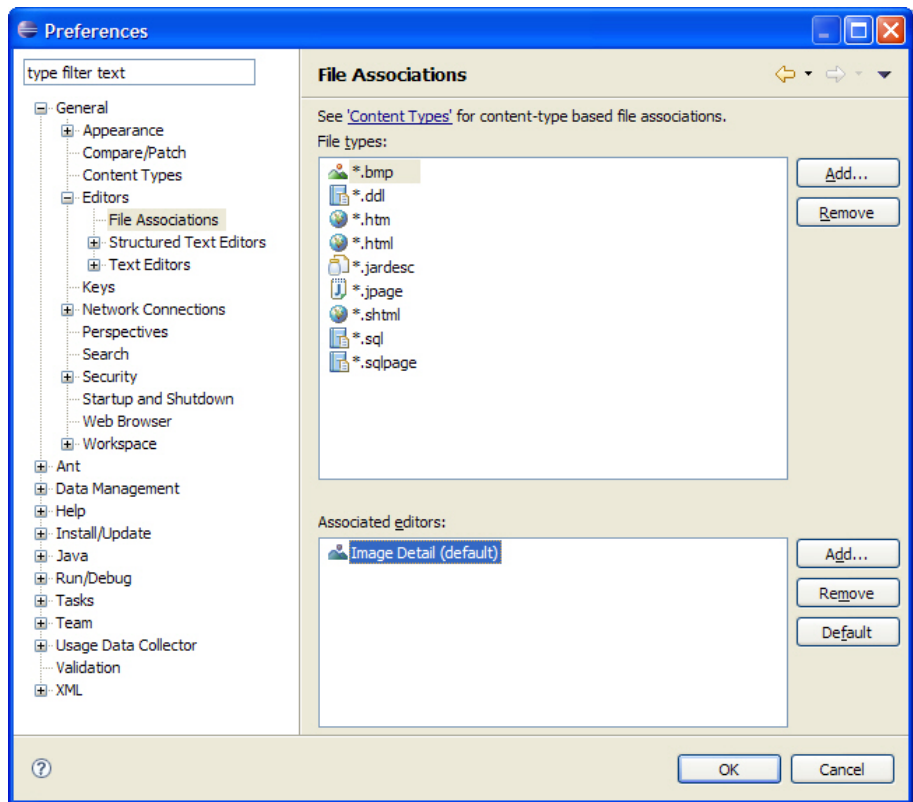
3. In the top list on this page are the file types for which associations can be made. The bottom list contains those editors or tools configured for the file type selected in the top list. File types can also be added to the top list. Begin by selecting the item * . bmp in the top list. The list of associated editors will be empty. Click the **[Add...]** button to the right of this empty list to create a file association.

This will display the Editor Selection list, where an internal or external editor can be selected. Here an internal editor will be used.



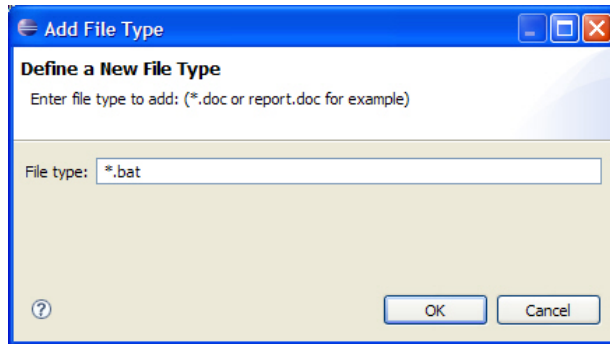
4. For a bitmap file you can select the Image Detail editor in this list. Then click the [OK] button to make this selection.

This returns you to the File Associations preference page, where the Image Detail editor is now configured as the default editor for bitmap files.



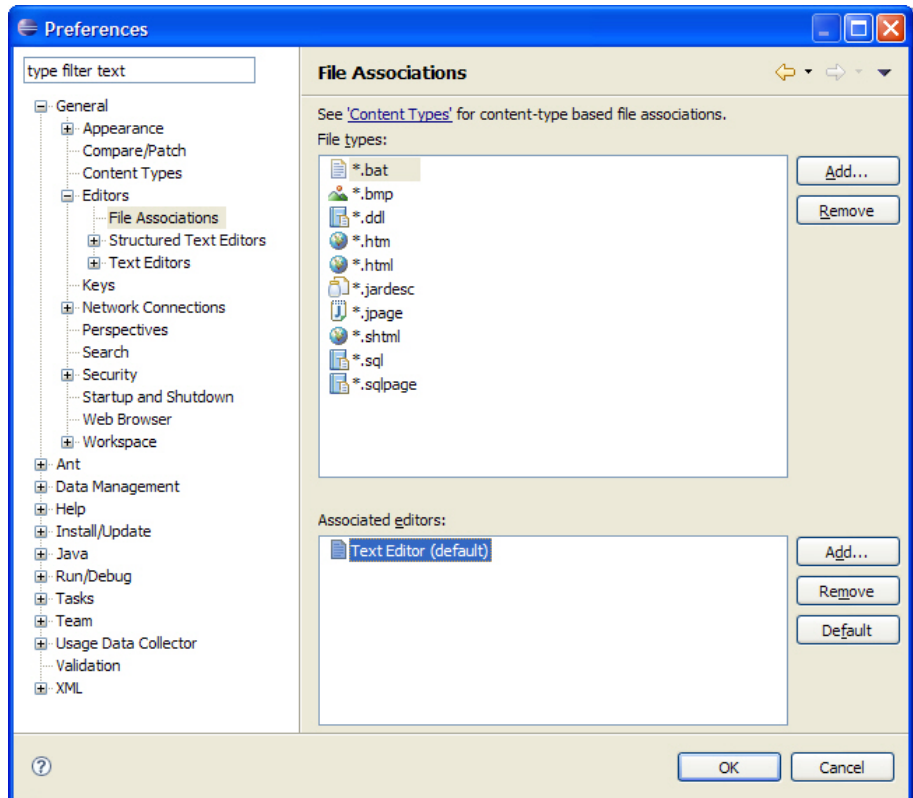
5. Next the remaining file associations can be created. Begin by checking the configuration of `.sql` files. By default the SQL File Editor is configured as the default for this file type. This is a part of the Data Tools package provided with Eclipse. You may select another editor if one is available, though this is the recommended editor for Agentry application projects.
6. The remaining file types that may need to be configured are those related to File System connections. These can be batch files for Windows deployments or shell scripts for Linux and Unix deployments of the Agentry Server. By default these file types are not listed in the Preference page for file associations. They can be added. To do this, click the **[Add...]** button to the right of the File types list.

This will display the Add File Type screen as shown next.



7. Enter the file extension in the format `*.ext` in the field provided and click the **[OK]** button.

This will return you to the File Associations preferences page where the new file type will now be listed.



8. This new file type can now be selected and an Editor can be associated with it, just as before. In the previous example Windows batch files are configured to have the Text Editor as the default editor.

9. This procedure can now be repeated for any other file types. In the case of Linux or Unix development, the recommended file type of `.sh` should be added and configured. While the file extensions are not required by these operating system types for proper execution, within the Agentry application project an extension is needed. This is so a file association can be configured for proper display and management of the files.

The file associations for those file types used in Agentry application projects have now been configured. In the examples provided in this procedure the following configurations have been made:

- Bitmap files will be displayed in the Image Detail editor.
- Windows batch files will be displayed in the standard Text Editor for Eclipse.
- Shell script files will be displayed in the standard Text Editor for Eclipse.

Configuring Eclipse File Encoding for Agentry Projects

Prerequisites

Before performing this procedure the following items must be addressed:

- The Agentry Editor plug-in and Eclipse Platform must be installed.
- The file associations should be configured for `.sql`, `.bat` or `.sh`, as needed, within the Eclipse Platform.
- Identify the file encoding format for any Java source files that may be opened within the same workspace as the Agentry application project.

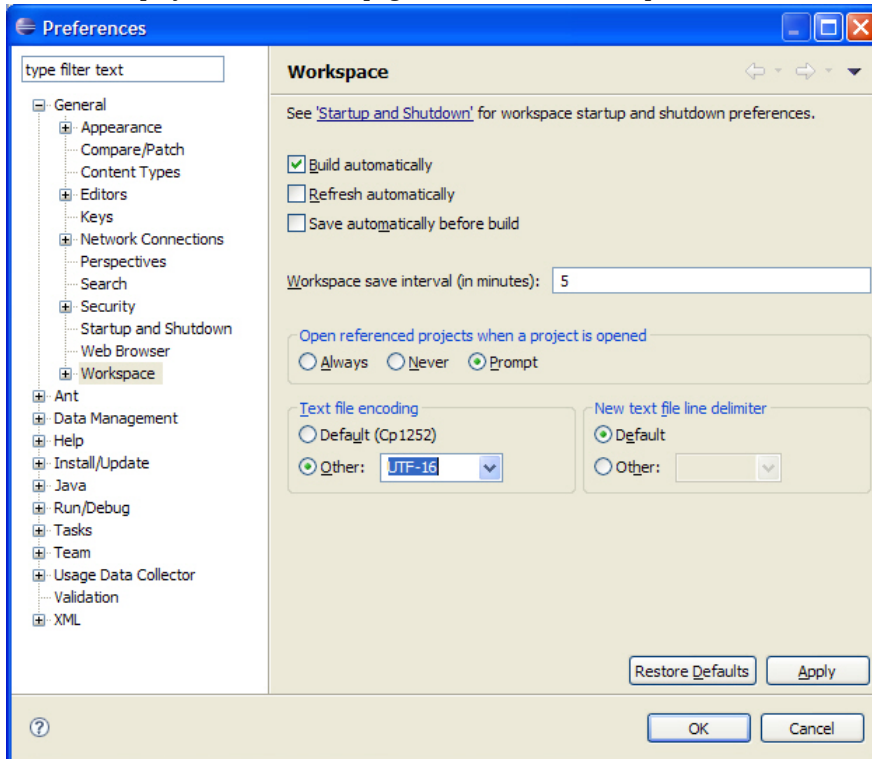
Task

This procedure describes the steps to configure the file encoding settings and preferences within the Eclipse platform as required by the Agentry Editor plug-in. The assumption is that the file editors for the various file types have been configured based on the recommendations provided by Syclo. If one or more different file editors are to be used for certain file types, the following procedure may or may not be valid. Those tools described here do not support or provide editor-specific file encoding, instead inheriting this behavior from the Eclipse workspace settings.

This procedure configures the current Eclipse workspace to use the file encoding format UTF-16, which is a unicode format. This will then result in all text editing tools treating all text files opened in any editor as if they are encoded in this format. Certain files external the Agentry application project but possibly a part of its overall implementation may not be encoded in this format. The primary example of such files would be Java files for a JVM system connection. The Java perspective within Eclipse does have a preference setting for file encoding. This is set for an individual Java project. The proper encoding setting for Java projects is dependent upon how those Java source files were encoded upon creation. The common option for such files is the UTF-8 encoding. However, other encoding formats are possible and this should be determined before making changes.

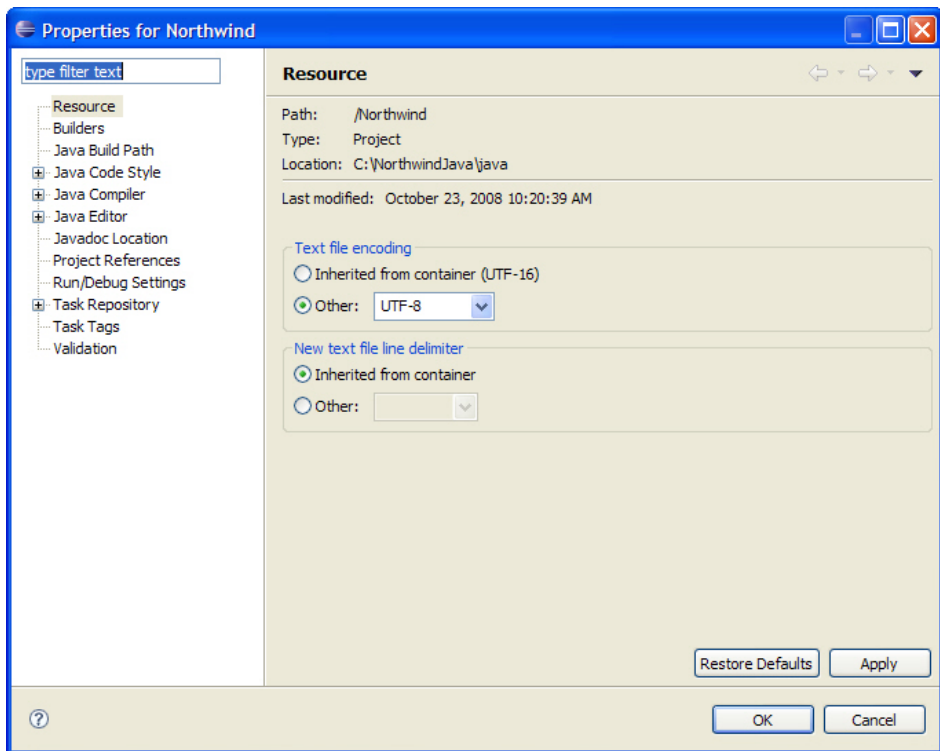
1. Start the Eclipse platform and open the workspace to be used for the Agency Editor. Once Eclipse is started, select the menu item **Window | Preferences**. In the tree control on the left of the Preferences screen select General -> Workspace.

This will display the Preferences page for the current workspace.



2. On this page look for the section **Text file encoding**. Select the **Other** radio button and then select the item UTF-16 in the associated drop down list.
3. Click the **[OK]** button to close the Preferences screen and apply the changes. If a Java project is not to be opened within this same workspace, or if the Java project's files have been encoded with UTF-16, no further actions are needed. Otherwise, continue on with this procedure.
4. Switch to the Java perspective in Eclipse and open the Java project for this same workspace. Select the menu item **Project | Properties**.

This will display the Properties page for the java project:



5. In the tree control to the left select the item **Resource**. Then look for the section **Text file encoding**. Select the radio button **Other** and then select the proper encoding option based on the encoding of the source files in the Java project. Note that if these files are encoded in either ASCII or UTF-8, then UTF-8 can be selected. For other file encoding formats select the proper item.
6. Click the **[OK]** button to close the Properties page and save these changes.

This procedure results in changing the Eclipse workspace's default file encoding for all text files to the unicode format UTF-16. This is the encoding in which all text files created by the Editor are formatted. The exception to this are java source files.

Installing the Agency Test Environment

Prerequisites

Address the following items prior to performing this procedure:

- The Agentry Test Environment installer is found in the SAP Mobile SDK package on the SAP Service Marketplace. Extract this installer from that package prior to performing this procedure.
- Determine the installation location of the Agentry Test Environment on the intended host system. By default this location is `C:\Agentry\Test Environment`. You can specify a different location during the installation if desired.
- Log into the host system as an administrator, with privileges to install software to the desired location.

Task

This procedure describes the steps necessary to install the Agentry Test Environment software component. This component is used as a client test tool during development of applications using the Agentry archetype. It is not intended for end users.

Perform this procedure to:

- Test modifications of the mobile application, when those modifications affect client-side behavior or client-server communications or synchronization.
- Test or debug run time issues in a Production environment. The Agentry Test Environment can be connected to the Agentry Production Server just as a normal client can.

When this procedure is complete, the Agentry Test Environment is available for use to test the client-side behaviors of any modifications made to the mobile application.

1. Start the Agentry Test Environment installer on the host system where you are installing this component by running the program `Agentry_6_1_0_x_TestEnvironment.exe` (NOTE: The value x represents the latest service pack or patch release of the system, depending on the version downloaded). This displays the **Welcome to the Agentry Test Environment Setup Wizard** screen.
2. Click the **[Next]** button to advance the wizard. This displays the **License Agreement** screen.
3. Click the **[Yes]** button to accept this agreement and advance the wizard. The **Choose Install Location** screen is displayed.
4. Enter the desired directory path in the Destination Folder field. This can be a new path, an existing location selected by clicking the **[Browse]** button, or the default location. Click the **[Next]** button to install the Agentry Test Environment to the specified directory. This displays the **Shortcuts for Agentry Test Environment** screen.
5. Select the desired shortcut locations by selecting or deselecting the listed shortcut options. Once the shortcuts are configured, click the **[Install]** button to begin the installation of the **Agentry Test Environment** software component.

The selected shortcuts are created by the installer. The **Installation Status** screen is displayed next indicating the installation progress. When the installation is complete, the **Completing the Agentry Test Environment Setup** wizard screen is displayed.

6. To start the Agentry Test Environment now, leave the box checked on this screen. Otherwise, deselect this box. Click the **[Finish]** button to close the wizard.

With the completion of this procedure, the Agentry Test Environment is installed to the selected location and is available for use in testing mobile applications in development, configuration, implementation, or production issue resolution scenarios.

Next

Just as with a standard client installation, the Agentry Test Environment needs to connect to an Agentry Server to perform an initial transmit. Prior to performing this transmit, the proper client platform to be mimicked, enabling or disabling scan and GPS simulations, and other similar selections should be made.

Agentry SAP Framework Foundation Installation Overview

The Agentry SAP Framework Foundation is implemented as an ABAP Add-On and is installed to the SAP system as a part of the implementation of the SAP Mobile Platform only when building or supporting mobile applications built in the Agentry archetype, and only when those mobile applications synchronize data with SAP systems. If both of these are not a part of your environment, this component is not installed. The framework includes both the Administration and Monitoring Portal, and the Integration Framework.

The Integration Framework provides the integration point to the SAP system for the Agentry Server. This then allows the Agentry Server to synchronize data with the SAP system for the mobile application. Included in the Integration Framework is the Configuration Portal used during implementation to create synchronization components of the mobile application.

The Administration and Monitoring Portal provides the tools and user interface to allow for the administration, configuration, and care and feeding tasks for the Agentry SAP Framework Foundation. These include security settings, log settings, user monitoring, and other similar tools.

There are also two additional resources needed in both the development and runtime environments when using the Agentry SAP Framework Foundation component. The first is an API contained in the Java file `SAPCommon-130881.jar`. The second is a configuration file, which must be created manually, named `JavaBe.ini`. Both of the files must reside within the folder where the Agentry Server has been installed. Additionally, the `SAPCommon-130881.jar` must be an available resource for any Java development projects related to the mobile application being developed.

The installation of the Agentry SAP Framework Foundation includes the following main tasks:

1. Verify the system requirements.

2. Install the Agentry SAP Framework Foundation software and support packages and place the `SAPCommon-130881.jar` file in the proper Agentry Server directory.
3. Create the `JavaBe.ini` configuration file and place it in the Agentry Server's installation folder.

Agentry SAP Framework Foundation Component

The following table lists the minimum version of the items required for the Agentry SAP Framework Foundation Component installation to the SAP system system. The software component for Agentry SAP Framework Foundation is SMFND release 600_700.

Software Requirements

Software Component	Release	Service Pack
SAP_ABA	700	SP14
SAP_BASIS	700	SP14

Installing Agentry SAP Framework Foundation

Prerequisites

The following items must be addressed prior to performing this procedure:

- Ensure you have the current versions of kernel, TP and R3trans.
- Current SPAM / SAINT Update - Compare the short text of the last SPAM / SAINT update you imported with that of the SPAM / SAINT Update in the SAP Service Marketplace. If the version of the SPAM / SAINT Update in the SAP Service Marketplace is more recent, import it.
- Verify the required software components have been installed, as listed in the system requirements for the Agentry SAP Framework Foundation.
- Verify the required support packages have been installed, as listed in the system requirements for the Agentry SAP Framework Foundation.
- No SAP password is required.

Task

Additional Information

- Space required in the transport directory: approximately 10 MB
- Total runtime: approximately 0.5 hour
- **SAP ABAP Add-Ons:**
 - Integration Foundation Add-On:
 - SMFND_600700_NW700.SAR

Language Support

The Agentry SAP Framework Foundation supports the following languages:

- English
- German
- Brazilian Portuguese
- French
- Spanish
- Korean
- Japanese
- Simplified Chinese

1. Log on to your SAP system as client **000** and as a user that has system administrative privileges. Do **not** use the SAP* or DDIC users.
2. Import the required user language(s) for all components that have already been installed. You must perform this language import prior to the installation of the Agentry SAP Framework Foundation.
3. Load the software package into your system via the Add-On Manager, using the transaction code **SAINT**.

For more information about this, see the online documentation for the Add-On Installation Tool. Select the help function in the application toolbar and navigate to **Online Documentation | Loading Installation Packages**.

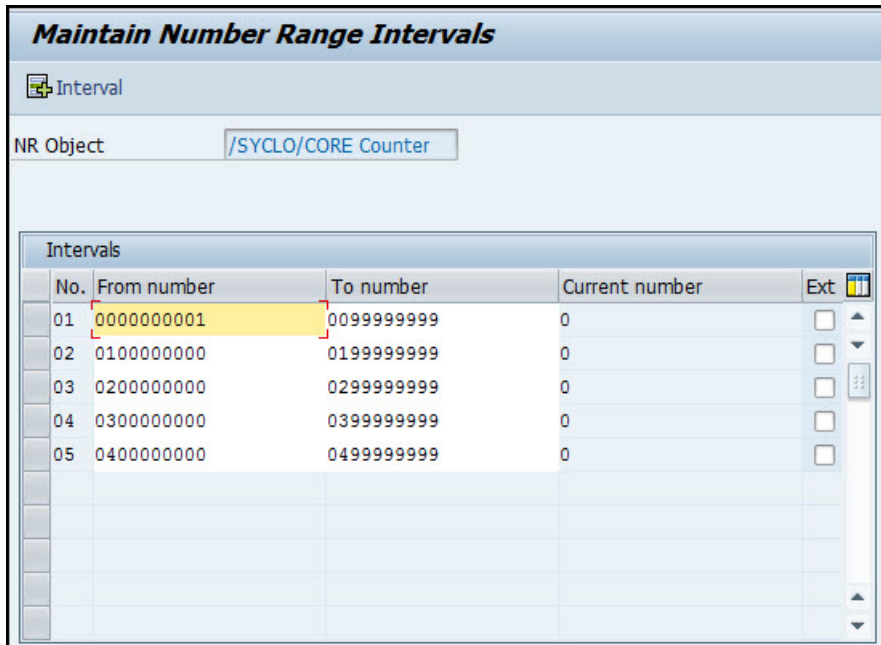
4. Start the installation of the Agentry SAP Framework Foundation using the Add-On Installation Tool, accessed from the transaction SAINT.
For more information about this, see the online documentation for the Add-On Installation Tool, selecting the help function on the toolbar.

5. Activate the Services for the Web Dynpro ABAP Applications

- a) Start transaction SICF.
- b) Activate all services under the node `default_host/sap/bc/webdynpro/Syclo`

6. Define Intervals for Number Range Objects /SYCLO/C_1 and /SYCLO/C_2

- a) Start transaction SNRO.
- b) Enter the number range object /SYCLO/C_1.
- c) Select **Number Ranges | Intervals**.
- d) Maintain or create the intervals 01 and 02 .
- e) Repeat these steps for the number range object /SYCLO/C_2 and maintain the intervals 01, 02, 03, 04, and 05. For example:



- Next, copy the file `SAPCommon-130881.jar` from the SAP Mobile SDK package to the installation location of the Agency Server.

With the completion of this procedure the Agency SAP Framework Foundation has been installed to the SAP system and is available for developers to begin creating the necessary components for a mobile application built using the Agency archetype.

Next

The final item to address is the creation of a configuration file for the Agency Server. See the instructions “Creating the JavaBE.ini File for SAP Systems” for details.

Creating the JavaBE.ini File for SAP Systems

Prerequisites

The following items must be addressed prior to performing this procedure:

- The `JavaBE.ini` file is needed only when using the Agency SAP Framework Foundation.
- This file is needed by the Agency Server. This component should be installed, per instructions found in the SAP Mobile SDK installation guide.

- This file should NOT be manually placed within the SAP Mobile Platform Runtime environment. Rather it should be packaged and deployed per documented procedures for deploying Agentry applications.

Task

This procedure documents the steps necessary to create the JavaBE.ini configuration file for use by the Agentry Server. The SAPCommon-130881.jar processes this file at runtime and expects certain sections and settings to be present. These revolve primarily around connectivity and authentication to and with the SAP System.

In addition to these required sections, in some implementations developers choose to add additional configuration sections. These additional sections are valid, but are outside the scope of this document. Typically these settings are consumed by the synchronization logic built for a specific application.

1. Create a new plain text file named JavaBE.ini and save it in the installation folder of the Agentry Server.
2. Using a plain text editor, open the JavaBE.ini file for editing.
3. Create the [HOST] section with the following settings:

```
[HOST]
```

```
server=<your sap host system's network name>
```

```
APPNAME=<name of your mobile application>
```

4. Create the [CONFIG] section exactly as shown below. Note there are other options for the source setting listed in this section, but the default is sufficient for initial configuration. Further information on this is provided in development guides.

```
[CONFIG]
```

```
source=SAP
```

5. Create the [CLIENT_NUM] section with the following setting:

```
[CLIENT_NUM]
```

```
CLIENT=<client number for Agentry Server to communicate with SAP  
Application server>
```

6. Create the [SYSTEM_NUM] section with the following setting:

```
[SYSTEM_NUM]
```

```
SYSNUM=<system number for Agentry Server to communicate with the
SAP Application Server>
```

7. Create the [LANGUAGE] section with the LANG option set to the two character SAP language key matching the language of the SAP system:

```
[LANGUAGE]
```

```
LANG=<two chracter SAP language key>
```

8. Create the [LOGGING] section, with the logging level set based on the information provided after the example:

```
[LOGGING]
```

```
Level=<logging level>
```

- 1 - Fatal Errors only
- 2 - Errors and above
- 3 - Warnings and above
- 4 - Informational messages and above
- 5 - Debugging messages and above
- 6 - Trace level debugging

9. Create the [LOGON_METHOD] section with the following settings, using the descriptions of settings provided below the example. The option set here will specify whether the [GLOBAL_LOGON] or [GROUP_LOGON] section is needed as well:

```
[LOGON_METHOD]
```

```
LOGON_METHOD=<logon method to SAP system>
```

- USER_AUTH - Standard User ID and password authentication is used.
- USER_AUTH_GLOBAL - Pooled connections using a single user ID and password; requires the section [GLOBAL_LOGON] to also be created.
- USER_AUTH_GROUP - User ID and password authentication with the SAP Message Server (load balancing) is used; requires the section [GROUP_LOGON] to also be created.

10. If the LOGON_METHOD is set to USER_AUTH_GLOBAL, create a new section named [GLOBAL_LOGON] as shown next:

```
[GLOBAL_LOGON]
```

```
UID=<User ID shared by all users>
```

```
UPASSWORD=<Password shared by all users>
```

```
SHAREDCONNECTIONS=<number of connections created for the shared pool>
```

- 11. If the LOGON_METHOD is set to USER_AUTH_GROUP, create a new section named [GROUP_LOGON] as shown next:**

```
[GROUP_LOGON]
```

```
MESSAGE_SERVER=<host name or IP of the SAP Message Server>
```

```
GROUP_NAME=<name of the SAP Application Server group>
```

```
SYSTEM_ID=<name or ID of the SAP system>
```

```
CLIENT=<client number to be used by the Agentry Server to connect to the SAP system>
```

- 12. Create the section [REQUIRED_BAPI_WRAPPER] with the settings and values listed exactly as shown next:**

```
[REQUIRED_BAPI_WRAPPER]com.synclo.sap.bapi.LoginCheckBAPI=/SYCLO/  
CORE_SUSR_LOGIN_CHECK
```

```
com.synclo.sap.bapi.RemoteUserCreateBAPI=/SYCLO/  
CORE_MDW_SESSION1_CRT
```

```
com.synclo.sap.bapi.RemoteParameterGetBAPI=/SYCLO/  
CORE_MDW_PARAMETER_GET
```

```
com.synclo.sap.bapi.SystemInfoBAPI=/SYCLO/CORE_SYSTINFO_GET
```

```
com.synclo.sap.bapi.ChangePasswordBAPI=/SYCLO/  
CORE_SUSR_CHANGE_PASSWD
```

```
com.synclo.sap.bapi.CTConfirmationBAPI=/SYCLO/  
CORE_OUTB_MSG_STAT_UPD
```

```
com.synclo.sap.bapi.DTBAPI=/SYCLO/CORE_DT_GET
```

```
com.synclo.sap.bapi.GetEmployeeDataBAPI=/SYCLO/  
HR_EMPLOYEE_DATA_GET
```

CHAPTER 4: Installing the Agentry Components of the SAP Mobile SDK

```
com.synclo.sap.bapi.GetUserDetailBAPI=/SYCLO/CORE_USER_GET_DETAIL
```

```
com.synclo.sap.bapi.GetUserProfileDataBAPI=/SYCLO/  
CORE_USER_PROFILE_GET
```

```
com.synclo.sap.bapi.PushStatusUpdateBAPI=/SYCLO/CORE_PUSH_STAT_UPD
```

```
com.synclo.sap.bapi.RemoteObjectCreateBAPI=/SYCLO/  
CORE_MDW_USR_OBJ_CRT
```

```
com.synclo.sap.bapi.RemoteObjectDeleteBAPI=/SYCLO/  
CORE_MDW_USR_OBJ_DEL
```

```
com.synclo.sap.bapi.RemoteObjectGetBAPI=/SYCLO/  
CORE_MDW_SESSION_GET
```

```
com.synclo.sap.bapi.RemoteObjectUpdateBAPI=/SYCLO/  
CORE_MDW_SESSION_UPD
```

```
com.synclo.sap.bapi.RemoteReferenceCreateBAPI=/SYCLO/  
CORE_MDW_USR_KEYMAP_CRT
```

```
com.synclo.sap.bapi.RemoteReferenceDeleteBAPI=/SYCLO/  
CORE_MDW_USR_KEYMAP_DEL
```

```
com.synclo.sap.bapi.RemoteReferenceGetBAPI=/SYCLO/  
CORE_MDW_SESSION_GET
```

```
com.synclo.sap.bapi.RemoteReferenceUpdateBAPI=/SYCLO/  
CORE_MDW_SESSION_UPD
```

```
com.synclo.sap.bapi.RemoteSessionDeleteBAPI=/SYCLO/  
CORE_MDW_SESSION1_DEL
```

```
com.synclo.sap.bapi.RemoteUserDeleteBAPI=/SYCLO/  
CORE_MDW_SESSION1_DEL
```

```
com.synclo.sap.bapi.RemoteUserUpdateBAPI=/SYCLO/  
CORE_MDW_SESSION_UPD
```

```
com.synclo.sap.bapi.TransactionCommitBAPI=WFD_TRANSACTION_COMMIT
```

```
com.syclo.sap.bapi.SignatureCaptureBAPI=/SYCLO/  
CS_DOBDSDOCUMENT_CRT
```

13. Save and close the file to the installation location of the Agentry Server.

With the completion of this procedure the `JavaBe.ini` configuration file for the Agentry Server is created, with settings to connect to the SAP system for which the mobile application will be developed.

Installing the Agentry SDK

Prerequisites

The following items must be addressed prior to performing this procedure:

- This SDK is provided for development purposes and it is assumed the person installing these components understands the development of ActiveX controls, and/or inter process communications in Windows Mobile environments. If such items are not needed for your implementation, then this SDK need not be installed and is not needed for any other mobile application development using the Agentry Editor or other components of the SAP Mobile SDK.
- This SDK should be installed to the location where needed for development work; i.e. onto the same host system as the IDE or other development toolset to be used for development; or to a file share or network drive accessible to these tools.

Task

This procedure provides the steps for installing the Agentry SDK, which can be found in the SAP Mobile SDK package provided on the SAP Service Marketplace. This is a simple installation procedure, consisting of executing the installation program and selecting the location for the files within the Agentry SDK.

Once the SDK has been installed, the resulting files are found in two distinct folders within the base installation location you will select:

- Agentry ActiveX
- Agentry External Process

For details on the contents of this SDK, and how to develop applications or controls using this SDK, see the *Developer Guide: Agentry Applications 2.3*, specifically the information beginning in the section “The Agentry SDK,” and the subsequent sections “Technical Overview - ActiveX Controls and the Agentry Client” and “Agentry Client API for External Processes Technical Overview.”

CHAPTER 4: Installing the Agentry Components of the SAP Mobile SDK

1. Launch the installer program by executing `Agentry_6.1.0.x_SDK.exe`. (NOTE: 'x' will vary depending on the service pack version currently available).
2. Click the **Next** button on the first screen **Welcome to the Agentry Software Development Kit Setup Wizard**.
3. Click the **Yes** button to accept the license agreement presented on the **License Agreement** screen.
4. Next specify the location where you would like the resources of the SDK to be installed in the **Choose Install Location** screen.
5. Click **Install** to install the Agentry SDK. Finish the wizard when prompted by the last screen,

With the completion of this procedure the resources of the Agentry SDK have been installed. These resources can be moved or copied to other locations as needed and can be included in development projects.

Next

If you have not done so already, review the information provided on the Agentry SDK found in the *Developer Guide: Agentry Applications*.

Uninstall SAP Mobile SDK using the Control Panel option to remove programs, which launches the uninstaller.

Prerequisites

- Shut down SAP Mobile WorkSpace.
- Back up any user-created files and log files you want to keep from the installation directories to another location.
- Prevent virus scans, backups, and system updates from interfering with the uninstallation by either excluding the SAP Mobile SDK installation directory from these processes or temporarily disabling these processes.

Task

1. From the Control Panel, begin removing the program.
This starts the uninstaller.
2. Click **Next** in the welcome window.
3. Click **Uninstall** to start uninstalling SAP Mobile SDK.
4. Click **Finish** when you see:

The InstallShield wizard has successfully uninstalled SAP Mobile SDK *<version>*.

Uninstall SAP Mobile SDK using the Control Panel option to remove programs, which launches the uninstaller.

Prerequisites

- Perform any backups needed of log files, business logic, and other resources that may be needed for archive or upgrade purposes.
- Archive the backed up resources.
- Log in as a user with administrative privileges.

Task

1. If it is not currently running, start the Eclipse instance where the Agency Editor plug-in is installed.
2. Select **Help > Install New Software**.
3. Click **What is already installed?**.
4. In the dialog that displays, select the **Installed Software** tab.
5. In the list of installed components, select the **Agency Editor plug-in** and click **Uninstall**.
6. In the Uninstall Details dialog, click **Finish** to remove the Agency Editor plug-in from the Eclipse instance.
7. When prompted, select **Restart Now** to restart Eclipse and apply the changes.

Note: This restart affects only Eclipse. It is not a full system restart.

Scripting Silent Installation or Uninstallation

You can automate installation, upgrading, or uninstallation of SAP Mobile SDK components by specifying settings in a text file, then running a script that silently runs the SAP Mobile SDK installer or uninstaller.

Silent Installation

Use the `SilentInstall_Win.bat` script to automate the installation or upgrade process. The SAP Mobile SDK installer runs silently, taking input from the `SilentInstall_Win.txt` file.

The silent installation capability allows you to perform any particular type of SAP Mobile SDK installation or upgrade, without entering information through the installer interface. There is no difference in the scripting process between a silent installation and a silent upgrade.

- Configure installer settings in a `SilentInstall_Win.txt` file.
- Run the installer from the command prompt, using the `SilentInstall_Win.bat` script.
- If User Access Control (UAC) is present in the version of Windows where you are installing, disable it. Restart Windows for the change to take effect.

Note: The SAP Mobile SDK installer can install only SDK components. The SAP Mobile Platform Runtime requires a different installer for the server components.

1. *Planning a Silent Installation*

Review the content of the default `SilentInstall_Win.txt` file and determine what changes you must make for the SAP Mobile SDK installation you want to automate.

2. *Modifying Properties in the SilentInstall_Win.txt File*

Edit a separate copy of the default `SilentInstall_Win.txt` configuration file for each type of SAP Mobile SDK installation you want to automate.

3. *Running a Silent Installation*

Run the `SilentInstall_Win.bat` script from a command prompt.

Planning a Silent Installation

Review the content of the default `SilentInstall_Win.txt` file and determine what changes you must make for the SAP Mobile SDK installation you want to automate.

1. Locate the `SilentInstall_Win.bat` script and `SilentInstall_Win.txt` file in the root directory of the SAP Mobile SDK installation media.
2. If you cannot modify those files on the installation media, create a copy of the installer image in a location from which you can run the installer.
3. Use a text editor to open the `SilentInstall_Win.txt` file, and review the current settings.

Comments preceding each group of settings describe the valid options, and any special requirements for the entry.

4. Determine what changes you want to make in the installation settings.

For example, you may want to change license information and installation location.

Modifying Properties in the SilentInstall_Win.txt File

Edit a separate copy of the default `SilentInstall_Win.txt` configuration file for each type of SAP Mobile SDK installation you want to automate.

1. From the root directory of the installer image, use a text editor to open `SilentInstall_Win.txt`.
2. Make the changes you have determined are necessary.
3. Save your changes.

Running a Silent Installation

Run the `SilentInstall_Win.bat` script from a command prompt.

Prerequisites

1. To accommodate SAP Mobile Platform internal communications, you may need to reconfigure intrusion detection/prevention systems (either hardware- or software-based). See the SAP Mobile Platform *Landscape Design and Integration* guide.
2. Run the installation script as an administrator on Windows 7 and Windows Server 2008 R2.

Task

1. Confirm that the root directory of the installer image contains the `SilentInstall_Win.txt` configuration file that you modified for a particular installation type.

2. From a command prompt on the installation target host, navigate to the root directory of the installer image and launch the silent installation script:

```
SilentInstall_Win.bat
```

3. Check the log files for errors.
 - Errors that cause a silent installation to fail before the installation directories are created are logged in `%SystemDrive%\SDKInstall.log`.
 - Errors encountered after the installation destination is created are logged in `SMP_HOME\InstallLogs\SDKInstall.log`, where `SMP_HOME` represents the SAP Mobile SDK installation directory, down to the `MobileSDKXXX` folder.

Silent Uninstallation

Use the `SilentUninstall_Win.bat` script to automate the uninstallation process. The SAP Mobile SDK uninstaller runs silently, taking input from the `SilentUninstall_Win.txt` file.

The silent uninstallation capability allows you to perform uninstallation of SAP Mobile SDK, without entering information through the uninstaller interface.

- Run the uninstaller from the command prompt, using the `SilentUninstall_Win.bat` script.
- There are no options with silent uninstallation.

When you run a silent installation, the `SilentUninstall_Win.bat` script and `SilentUninstall_Win.txt` configuration file are copied to the `SMP_HOME\Uninstaller` directory, where `SMP_HOME` represents the SAP Mobile SDK installation directory, down to the `MobileSDKXXX` folder. If you did not run a silent installation, you must:

- Copy those files to the location above from the SAP Mobile SDK installation media root directory.

Note: The SAP Mobile SDK uninstaller can only uninstall SDK components. The SAP Mobile Platform Runtime requires a different uninstaller for the server components.

1. *Planning a Silent Uninstallation*

Move and modify the `SilentUninstall_Win.bat` file.

2. *Running a Silent Uninstallation*

Run the `SilentUninstall_Win.bat` script from a command prompt.

Planning a Silent Uninstallation

Move and modify the `SilentUninstall_Win.bat` file.

1. Move the `SilentUninstall_Win.bat` to a temporary location outside of the SAP Mobile SDK installation tree.

For example, `C:\temp`.

Note: The current location of the `SilentUninstall_Win.bat` file should be the `SMP_HOME\Uninstaller` directory, where either a silent installation placed it or you copied it earlier. (`SMP_HOME` represents the SAP Mobile SDK installation directory, down to the `MobileSDKXXX` folder.)

2. Use a text editor to open the `SilentUninstall_Win.bat` file, and change the line beginning with `cd` as follows.

Line before change:

```
cd /d %~dp0
```

Line after change:

```
cd /d SMP_HOME\Uninstaller
```

where `SMP_HOME` represents the SAP Mobile SDK installation directory, down to the `MobileSDKXXX` folder.

Running a Silent Uninstallation

Run the `SilentUninstall_Win.bat` script from a command prompt.

Prerequisites

1. Shut down SAP Mobile WorkSpace.
2. Prevent backups and virus scans from interfering with the uninstallation. You can either exclude the SAP Mobile SDK installation directory from backups and virus scans, or temporarily disable them.
3. Run the uninstallation script as an administrator on Windows 7 and Windows Server 2008 R2.

Task

1. On the host where you want to run a silent uninstallation, confirm that the `SMP_HOME\Uninstaller` directory contains the correct version of the `SilentUninstall_Win.txt` configuration file that you modified for a particular uninstallation type. (`SMP_HOME` represents the SAP Mobile SDK installation directory, down to the `MobileSDKXXX` folder.)
2. From a command prompt on the installation target host, navigate to the directory to which you moved the `SilentUninstall_Win.bat` file and launch the silent uninstallation script:

```
SilentUninstall_Win.bat
```

3. Check for errors in *SMP_HOME*\InstallLogs\SDKUninstall.log, where *SMP_HOME* represents the SAP Mobile SDK installation directory, down to the *MobileSDKXXX* folder.
4. Delete any folders and files remaining in the SAP Mobile SDK installation directory, if the uninstaller could not remove them.
5. Restart the system to delete folders and files remaining in the SAP Mobile SDK installation directory.
6. Repeat these steps on the target systems for each uninstallation you are automating.

CHAPTER 8 **Troubleshooting**

Determine the cause of common problems and apply the recommended solution.

Updating the Database Connection Profile

Due to JDBC driver location changes, beginning in SAP Mobile Platform version 2.1. ESD #3, the existing database connection profile may fail to ping/connect with this exception:

```
java.lang.ClassNotFoundException:  
com.sybase.jdbc3.jdbc.SybDriver.
```

If you see this error, change the JDBC driver location. For example, for a SQL Anywhere® connection profile:

1. In Enterprise Explorer, right-click the database connection profile, and select **Properties**.
2. In the left pane, select **Sybase ASA Connection Properties**.
3. Click the upper-right triangle button to invoke the Edit Driver Definition dialog.
4. Select the **Jar List** tab, and select the driver file entry that contains the incorrect JDBC driver file path, then click **Edit JAR/Zip** to invoke the Select the file dialog.
5. Click **Look in** to specify the path. For example: *SMP_HOME*\MobileWorkSpace\Eclipse\sybase_workspace\framework\eclipse\plugins\com.sybase.jconnect_7.0.0.201212211040\lib., where *SMP_HOME* represents the SAP Mobile SDK installation directory, down to the MobileSDKXXX folder.
6. Click **OK** to complete the driver definition location change.
7. Click **OK** to complete the connection profile properties change.

APPENDIX A **Locations of Code Libraries**

Find the code libraries necessary to develop different types of applications on different device platforms.

The following table shows the top-level subdirectories that are created when you install SAP Mobile SDK. The directory names identify the type of application and the device platform that the different code libraries support.

By default, SAP Mobile SDK components are installed in the `C:\SAP\MobilePlatform\MobileSDKXXX` directory. In this guide, *SMP_HOME* represents the SAP Mobile SDK installation directory, down to the `MobiledSDKXXX` folder.

Table 3. SAP Mobile SDK Installation Subdirectories

Directory	Description
_jvm	JVM used by the uninstaller.
Eclipse	Eclipse files for SAP Mobile WorkSpace GUI
HybridApp	Libraries supporting Hybrid App applications
HybridApp\API\AppFramework	Hybrid App application framework files
HybridApp\API\Container	JavaScript files supporting Hybrid App applications
HybridApp\API\Container\android	Android-specific APIs for Hybrid App applications
HybridApp\API\Container\blackberry	BlackBerry-specific APIs for Hybrid App applications
HybridApp\API\Container\ios	iOS-specific APIs for Hybrid App applications
HybridApp\API\Container\wm	Windows Mobile-specific APIs for Hybrid App applications
HybridApp\Containers\Android	Android-specific libraries for Hybrid App applications
HybridApp\Containers\BB	BlackBerry-specific libraries for Hybrid App applications
HybridApp\Containers\iOS	iOS-specific libraries for Hybrid App applications

APPENDIX A: Locations of Code Libraries

Directory	Description
HybridApp\Containers\wm	Windows Mobile-specific libraries for Hybrid App applications
HybridApp\PackagingTool	Packaging tool files for Hybrid App applications
InstallLogs	Log files created each time the SAP Mobile SDK installer is used. Use these logs to troubleshoot installer issues.
JDKx.x.x_x	JDK required by SAP Mobile SDK components.
licenses	SySAM license file for SAP Mobile SDK and license files for third-party components included in SAP Mobile SDK.
MAKit	Mobile Analytics Kit-specific libraries for Hybrid Web Container
ObjectAPI	Libraries supporting Object API applications
ObjectAPI\Android	Android-specific files for Object API
ObjectAPI\apidoc	Reference documentation for Object API
ObjectAPI\BB	BlackBerry-specific files for Object API
ObjectAPI\DOE	Libraries supporting Object API applications that connect to SAP Data Orchestration Engine (DOE)
ObjectAPI\DOE\Android	Android-specific files using Object API with DOE
ObjectAPI\DOE\iOS	iOS-specific files using Object API with DOE
ObjectAPI\DOE\Win32	Win32-specific files using Object API with DOE
ObjectAPI\DOE\WM	Windows Mobile-specific files using Object API with DOE
ObjectAPI\iOS	iOS-specific files for Object API
ObjectAPI\Utils	Utilities supporting Object API applications
ObjectAPI\Win32	Win32-specific files for Object API
ObjectAPI\Win32\UltraLite	Win32 libraries for Object API applications using UltraLite database on device
ObjectAPI\WM	Windows Mobile-specific files for Object API

Directory	Description
ObjectAPI\WM\PocketPC	Windows Mobile-specific files for Object API on Pocket PC
ObjectAPI\WM\Smartphone	Windows Mobile-specific files for Object API on Smartphone
ObjectAPI\WM\Ultralite	Windows Mobile libraries for Object API applications using UltraLite™ database on device
OData	Libraries supporting OData applications
OData\Android	Android-specific files for OData
OData\BB	Blackberry-specific files for OData
OData\iOS	iOS-specific files for OData
sapjco	SAP Java Connector files
ThirdParty	License terms of third-party components included in SAP Mobile SDK
Uninstaller	SAP Mobile SDK uninstaller
MobileWorkspace	SAP Mobile Workspace development environment

Index

C

code library locations 93

D

database connection profile
 connecting to SAP Mobile Server on same host 20
 connecting to SAP Mobile Server remotely 21
 updating 91

G

generating
 served licenses 8
 unserved licenses 7
 getting started 1

H

host IDs
 determining 2
 lmutil utility 2

I

installation directories 93
 installing SAP Mobile SDK 13

L

license deployment models
 comparison of 3
 recommendations 3
 license file
 locating information 9
 obtaining 1
 licenses
 license files 11
 models 1
 obtaining 1
 recommendations for license models 3

locating code libraries 93

P

ports, communication
 SySAM license server 11
 prerequisites 1
 protecting the installation 22

R

recommendations
 license models 3

S

SAP Software Asset Management for licenses 1
 served license
 model 1
 served licenses 8
 silent installation 85
 configuration file 86
 planning 86
 running 86
 silent uninstallation 87
 planning 87
 running 88
 silent upgrade 85
 planning 86
 SPDC
 generating served licenses 8
 generating unserved licenses 7
 SySAM license server 1
 communication port 11

T

troubleshooting 91

U

uninstalling 81, 83
 silently 87

Index

- unserved license
 - model 1
- unserved licenses 7
- updating database connection profile 91
- upgrading 25

V

- verifying the installation 20
 - SAP Mobile SDK installed remotely 21
 - SAP Mobile SDK on same host as SAP Mobile Server 20