

Extension Services v5.7



Embedded Standard Widget Toolkit (eSWT) For Eclipse * Beta *

Contents

Chapter 1. Samples	1
Embedded SWT Controls	1
Creating the project	1
Running in WebSphere Studio	1
Running on a device.	2
Using the application	4
Embedded SWT TextEditor	4
Creating the project	4

Running on a device.	4
Using the application	6
Chapter 2. Reference	7
Embedded SWT	7
Sources of information	7
Windows Mobile 2003	8

Chapter 1. Samples

Embedded SWT Controls

The Embedded SWT Controls sample provides an Embedded SWT example Extension Services application which demonstrates configuration of the various eSWT widgets. This sample is included to show how to take Embedded SWT Controls sample project and run it within WebSphere Studio. Steps are also provided to demonstrate how to use the Extension Services Platform Builder to generate and run a target platform which includes this sample application.

Creating the project

Perform the following steps to create the Embedded SWT Controls sample application project:

1. Select **File > New > Example... > Extension Services > Embedded SWT Controls**. Click **Next**.
2. Click **Finish**.

Note: If prompted to switch to the SMF perspective, click **OK**.

Running in WebSphere Studio

Setting up

The Embedded SWT Controls sample requires the use of native libraries which WebSphere Studio must be able to locate. If you have not already completed the steps in the Getting Started section of this guide, perform the following steps to set up the environment for the Embedded SWT Controls sample.

Note: eSWT for Extension Services 5.7 only provides native libraries for Windows and Windows Mobile 2003 platforms so it is not possible to run this example inside of WebSphere Studio on Linux in this release. Please refer to the Running on Device section for information on creating Windows or Windows Mobile 2003 platforms for target devices.

1. On Windows, add the following entry to the end of your PATH environment variable:

```
WSDD_HOME\wsdd5.0\technologies\eswe\files\eswt\win32\x86
```

Where WSDD_HOME is the location where you installed WebSphere Studio Device Developer 5.7.

2. Restart WebSphere Studio.

Adding the Embedded SWT bundles to the bundle server: To add the eSWT bundles to the bundle server, perform the following procedure:

1. In the SMF perspective, select **Run > Run...**
2. Select SMF Bundle Server. Click **New**. Provide a name for the bundle server and keep all other default settings. Click **Run** to start the bundle server. For more information on the available bundle server options, refer to the *Service Management Framework Bundle Developer Tools User's Guide*.
3. In the SMF Perspective switch to the **SMF Bundle Servers** tab, then right click on the **Admin@localhost:8080/smf** bundle server and select **Import Bundles**.

4. Select **Add Directory** and browse to <WSDI_INSTALL_DIR>\wsdd5.0\technologies\eswe\bundlefiles (where WSDI_INSTALL_DIR is where WSDI was installed, eg. c:\Program Files\IBM\DeviceDeveloper). Click **OK**.
5. Select the + symbol next to the new entry that was just created to list all available bundles in that directory.
6. Click the box next to eswt-win32.jar and eswt-wm2003.jar to select them both, then click **Finish** to import the bundles into the bundle server.

Deploying the application

Perform the following steps to submit the Embedded SWT Controls sample application bundle to an SMF bundle server:

1. In the SMF perspective, select **Run > Run...**
2. Select **SMF Bundle Server**. Click **New**. Provide a name for the bundle server and keep all other default settings. Click **Run** to start the bundle server. For more information on the available bundle server options, refer to the Service Management Framework Bundle Developer Tools User's Guide.
3. Right click on the Embedded SWT Controls sample project and select **SMF > Submit Bundle...**
4. In the resulting Target Submission window, select **Submit Jar**.
5. Select **Admin@localhost:8080/smf** in the Export Targets box.
6. Click **Finish** to submit the bundle to the bundle server.
7. Switch to the Bundle Server view and open **Admin@localhost:8080/smf > Bundles**. Verify that a bundle named eSWTControlsSample is listed.

Launching the application

Perform the following steps to install and run the Embedded SWT Controls Sample application on an SMF runtime:

1. In the SMF perspective, select **Run > Run...**
2. Select **SMF Runtime**. Click **New**. Provide a name for the runtime and keep all other default settings. Click **Run** to start the runtime. In the SMF Runtime view, an active runtime is shown that contains only an SMF System Bundle. For more information on the available runtime options, refer to the Service Management Framework Bundle Developer Tools User's Guide.
3. Switch to the Bundle Server view and open **Admin@localhost:8080/smf > Bundles**. Right click the **eSWTControlsSample bundle** and select **Install**. The eSWTControlsSample bundle, along with all of its dependent bundles, will be installed onto the SMF runtime.

For instructions on how to access and use the Embedded SWT Controls sample application, see "Using the application" on page 4.

Running on a device

Deploying the application

Perform the following steps to submit the Embedded SWT Controls sample application bundle to the filesystem:

1. In the SMF perspective, right click on the **Embedded SWT Controls project** and select **SMF > Submit Bundle...**
2. In the resulting Target Submission window, select **Submit Jar** and click **Add Directory...**

3. Select a folder in which to submit the Embedded SWT Controls sample application bundle.
4. Select the newly added directory in the Export Targets box.
5. Click **Finish** to submit the bundle to the filesystem.
6. Verify that the bundle into the specified directory. The bundle will be named eSWTControlsSample+5_7_0.jar.

Creating a Platform Builder project

Perform the following steps to create a Platform Builder project that contains the Embedded SWT Controls sample application bundle. This project can be used to build a target platform for a specific device type that runs the Embedded SWT Controls sample application.

1. Select **File > New > Project > Extension Services > Platform Builder**. Click **Next**.
2. Enter eSWTControlsPlatformBuilder for the project name. Click **Next**.
3. On the Platform Options page, select the desired device type. Select the desired output format and specify an output location. If the output location is left blank, the build output will be placed in the eSWTControlsPlatformBuilder/output/ directory.

Note: eSWT for Extension Services 5.7 only provides libraries for Windows and Windows Mobile 2003 platforms.

Click **Next**.

4. On the J9 Options page, select the **jclFoundation** class library and the desired package format.
5. On the Startup Options page, leave all options with their default values. Click **Next**.
6. On the Platform Languages page, select the desired platform language support. Click **Next**.
7. On the Custom Bundles page, click **Add...** and select the **eSWTControls+5_7_0.jar** file from the filesystem location where it was deployed. Click **OK**.
8. Click the **Compute** button to determine the dependencies for the Embedded SWT Controls sample application bundle. A Resolve Dependencies dialog will appear. Select **Embedded SWT for Win32 (Beta)** from the list and then click **Resolve...**
9. A dialog will appear stating that all dependencies have been resolved. Click **OK**. Click **Next**.
10. On the System Bundles page, the system bundles that were added from the dependency resolution are listed. Click **Next**.
11. On the Summary page, make sure that the Build platform checkbox is selected. Review the summary and click **Finish**. The wizard will create the eSWTControlsPlatformBuilder project in the workspace and will immediately launch a build to generate a target platform.
12. When the build is complete, verify that the target platform package is placed in the previously specified output directory.

Launching the application

To run the target platform on your device, perform the following steps:

1. Install the target platform onto your device. This procedure may differ by device, depending on which output format was selected:

- If the output format was zip, tar or tar.gz, unpackage the platform package into a location on the device filesystem
 - If the output format was ipk, copy the ipk file to the device and install with the appropriate iPKG tools
2. Execute the platform startup script. The startup script is named StartSMF and is located in the smf directory of the target platform. Running this script will launch the Extension Services platform.

For instructions on how to access and use the Embedded SWT Controls sample, see "Using the application."

Using the application

Once the application has been launched, the Embedded SWT window will be created. In this new window there is a selection of available widgets, select one and click **Run** to display the widget's style configuration window.

Embedded SWT TextEditor

The Embedded SWT TextEditor sample provides an Embedded SWT example Extension Services application for Windows Mobile 2003 which provides a simple text editor. This sample is included to show how to use the Extension Services Platform Builder to generate and run a target platform for Windows Mobile 2003 which includes this sample application.

Take the note from the TextEditor example in the current doc, the one that refers to "A Small cup of java" and insert it here.

Creating the project

Perform the following steps to create the Embedded SWT TextEditor sample application project:

1. Select **File > New > Example... > Extension Services > Embedded SWT TextEditor**. Click **Next**.
2. Click **Finish**.

Note: If prompted to switch to the SMF perspective, click **OK**.

Running on a device

Deploying the application

Perform the following steps to submit the Embedded SWT TextEditor sample application bundle to the filesystem:

1. In the SMF perspective, right click on the **Embedded SWT TextEditor** project and select **SMF > Submit Bundle...**
2. In the resulting Target Submission window, select **Submit Jar** and click **Add Directory...**
3. Select a folder in which to submit the Embedded SWT TextEditor sample application bundle.
4. Select the newly added directory in the Export Targets box.
5. Click **Finish** to submit the bundle to the filesystem.
6. Verify that the bundle into the specified directory. The bundle will be named eSWTextEditorSample+5_7_0.jar.

Creating a platform builder project

Perform the following steps to create a Platform Builder project that contains the Embedded SWT Controls sample application bundle. This project can be used to build a target platform for a specific device type that runs the Embedded SWT Controls sample application.

1. Select **File > New > Project > Extension Services > Platform Builder**. Click **Next**.
2. Enter `eSWTTextEditorPlatformBuilder` for the project name. Click **Next**.
3. On the Platform Options page, select PocketPC for the device type. Select the desired output format and specify an output location. If the output location is left blank, the build output will be placed in the `eSWTTextEditorPlatformBuilder/output/` directory. Click **Next**.
4. On the J9 Options page, select the **jclFoundation** class library and the desired package format.
5. On the Startup Options page, leave all options with their default values. Click **Next**.
6. On the Platform Languages page, select the desired platform language support. Click **Next**.
7. On the Custom Bundles page, click **Add...** and select the `eSWTTextEditor+5_7_0.jar` file from the filesystem location where it was deployed. Click **OK**.
8. Click the **Compute** button to determine the dependencies for the Embedded SWT Controls sample application bundle. A Resolve Dependencies dialog will appear. Select the **Embedded SWT for Windows Mobile 2003 (Beta)** bundle from the list and click **Resolve...**
9. A dialog will appear stating that all dependencies have been resolved. Click **OK**. Click **Next**.
10. On the System Bundles page, the system bundles that were added from the dependency resolution are listed. Click **Next**.
11. On the Summary page, make sure that the **Build platform** checkbox is selected. Review the summary and click **Finish**. The wizard will create the `eSWTTextEditorPlatformBuilder` project in the workspace and will immediately launch a build to generate a target platform.
12. When the build is complete, verify that the target platform package is placed in the previously specified output directory.

Launching the application

To run the target platform on your device, perform the following steps:

1. Install the target platform onto your device. This procedure may differ by device, depending on which output format was selected. If the output format was zip, tar or tar.gz, unpackage the platform package into a location on the device filesystem.
2. Execute the platform startup script. The startup script is named `StartSMF` and is located in the `smf` directory of the target platform. Running this script will launch the Extension Services platform.

For instructions on how to access and use the Embedded SWT TextEditor sample application, see “Using the application” on page 6.

Using the application

Once the Extension Services platform has been started the Embedded SWT window will be created. The text widget comes up in an editable state, to change the behavior of the text editor use the Options menu at the bottom of the application.

Chapter 2. Reference

Embedded SWT

The Standard Widget Toolkit (SWT) is the part of the open source Eclipse project that provides widget functionality to the Eclipse platform. Similarly, the Embedded Standard Widget Toolkit (eSWT) is a pure subset of SWT that provides widget functionality for embedded platforms.

eSWT supports only those widgets and user interface behaviors that are common to embedded platforms. For example, since drag-and-drop functionality and tabbed folders are not common to embedded platforms, eSWT does not support them.

Table 1. List of widgets and UI behaviors that are supported by eSWT

• Button	• Label	• ScrollBar
• Canvas	• List	• Shell
• Combo	• Menu	• Slider
• Composite	• MenuItem	• Text

Including only commonly-used widgets allows eSWT to run more effectively on embedded devices.

To use Embedded SWT in your Windows[®] or Windows Mobile[™] 2003 project:

1. Include the appropriate `swt-win32-3049.dll` library on your library path. You can find the DLL in one of the following locations, where `<WSDD_HOME>` is the directory where WebSphere Studio Device Developer was installed:
 - `<WSDD_HOME>\wsdd5.0\technologies\eswe\files\eswt\win32` for the Windows platform
 - `<WSDD_HOME>\wsdd5.0\technologies\eswe\files\eswt\wm2003` for a Windows Mobile 2003 device
2. Reference the appropriate code JAR in your project:
 - `<WSDD_HOME>\wsdd5.0\technologies\eswe\bundlefiles\eswt-win32.jar` for the Windows platform
 - `<WSDD_HOME>\wsdd5.0\technologies\eswe\bundlefiles\eswt-wm2003.jar` for the Windows Mobile 2003 platform

Note: The eSWT code JARs do not include source code.

Sources of information

Since Embedded SWT (eSWT) is a subset of SWT, you can refer to the SWT resources for more information:

- **SWT Documentation** (<http://help.eclipse.org/help21/index.jsp>)
For a discussion of SWT concepts and how to build an SWT application refer to the *Standard Widget Toolkit* chapter in the **Platform Plugin Developer's Guide** of the online Eclipse documentation. This documentation covers SWT's core concepts - widgets, layouts and events.
- **SWT Articles** (<http://www.eclipse.org/articles/index.html>)

The Eclipse website includes articles on SWT. Particularly relevant articles include:

- "Understanding Layouts in SWT", which provides detailed FormLayout examples
- "A Small Cup of SWT", which discusses writing SWT applications for a Windows Mobile™ 2003 device

These articles provide in-depth explanations about various SWT concepts. Most of the covered concepts also apply to Embedded SWT.

- **SWT Website** (<http://www.eclipse.org/platform/index.html>)

The Eclipse SWT website offers a plethora of information about SWT.

Note: Pay special attention to the developer's page. This page includes code snippets and frequently asked questions.

- **SWT Newsgroup** (<http://www.eclipse.org/newsgroups/index.html>)

You can also subscribe to the SWT newsgroup. The newsgroup can be very helpful if you have specific questions about how to use SWT.

Windows Mobile 2003

The eSWT API is portable; however, platforms have different native behavior. The following table shows items that are not supported for a Windows Mobile 2003 device:

Table 2.

Item	Supported?	Notes/Workarounds
3D (button)	No	
setImage(Image) API (button)	No	Work-around: You can use a Label and set the image. Then hook mouse and keyboard events to it to emulate the button behavior.
SWT.TOGGLE (button)	Partially, see note	Note: You should change the background or foreground color for legibility. The default colors make it hard to read when the button is "pressed."
SWT.SIMPLE (combo)	No	Work-around - You can emulate this style by using a text widget and a list widget.
SWT.RIGHT_TO_LEFT (control)	No	
SWT.RIGHT_TO_LEFT (GC)	No	
setLineStyle(int) API (GC)	Partially, see note	Note: *Only SWT.LINE_SOLID and SWT.LINE_DASH are supported.
SWT.SEPARATOR (label)	No	
setDefaultItem(MenuItem) API (menu)	No	
setImage(Image) API (MenuItem)	Partially, see note	Note: This API is supported only when the MenuItem is on the top-level menu bar.
MouseTrack Listener	No	Note: MouseTrackListener events are not generated and do not apply to this device.
setEnabled(Boolean) API (scrollbar)	No	Note: Scrollbars are always enabled.

Table 2. (continued)

Item	Supported?	Notes/Workarounds
setVisible(Boolean) API (scrollbar)	Partially, see note	Note: Only supported when the size of the Scrollbar thumb is equal to the size of the Scrollbar range.
Shells	Partially, see note	Note: Shells cannot be minimized.