WebSphere® Studio Device Developer

Application Tools for Extension Services Developer's Guide

Third Edition (May 2004)

This edition applies to Application Tools for Extension Services 5.7 and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Chapter 1. Getting Started

This chapter contains introductory information for Extension Services.

Introducing the Application Tools for Extension Services

The Application Tools for Extension Services provide tooling that enables you to develop, test, and deploy Web applications for pervasive devices. Specifically, the Application Tools target the Extension Services for WebSphere® Everyplace® platform. This platform provides a component-based Java™ runtime that implements the OSGi Alliance, an open standards platform that supports the development, deployment, and management of services to Internet-enabled devices.

In the standard Web application model, applications run on a Web server and clients access these applications using Hyper Text Transfer Protocol (HTTP) through a browser. The standard model works well when you connect the client and server through a stable Internet connection. However, this model does not address problem domains in which the client is disconnected or sporadically connected to the Internet, such as mobile workforce applications.

Extension Services for WebSphere Everyplace extends Web applications to the device through a programming model that includes the following advantages:

- Created with open standards

  Extension Services is based on Java, the OSGi Alliance, and the J2EE programming model.
- Provides a component-based framework

  Service Management Framework™, the IBM® OSGi implementation, provides a framework above Java that enables you to manage multiple applications and services as components on the platform. SMF enables you to install, start, stop, and uninstall components, or bundles, without restarting the Java Virtual Machine (JVM). The platform runs only those components that are currently needed, instead of loading all of the libraries during the JVM setup.
- Provides platform management over the Internet

  You can manage the components running on the platform over the Internet through a management agent.
- Supports small footprint JVMs

  Extension Services supports WebSphere Everyplace Micro Environment JCL Foundation, WebSphere Everyplace Custom Environment JCL Resource Managed Gateway Plus, and WebSphere Everyplace Custom Environment JCL Max.
- Provides a familiar programming model and familiar development tools.

  Extension Services uses the server-side J2EE Web application programming model. The Application Tools extend WebSphere Studio's existing Web application development tools, which provides a familiar development environment.

Extension Services for WebSphere Everyplace is not a full implementation of the J2EE runtime. Instead, Extension Services supports portions of the J2EE programming model as OSGi services. The Application Tools support the Servlet

2.3 and JSP 1.2 Web application support provided by Extension Services for WebSphere Everyplace. Refer to “Supported Web development features” for more information about the features Extension Services supports.

Supported Web development features

Extension Services Web projects support the Servlet 2.3 and Java Server Pages (JSP) 1.2 specifications. A small set of restrictions are documented in the **Comparison to the Servlet 2.3 and JSP 1.2** section, in the Reference chapter of the Service Management Framework Runtime documentation. Enterprise Java Beans (EJB) and Enterprise Application Archives (EAR) are not supported.

You can use WebSphere Studio Servlet and JSP development tools on Extension Services Web projects. However, when you use these tools on an Extension Services project, consider the following:

- Extension Services only supports converting J2EE 1.3 projects to Extension Services projects. It does not support J2EE 1.2 projects.
- Only run Extension Services Web projects on Extension Services servers.
- Use Web Application Bundles (WAB) as the import and export archive file instead of Web Application Archive (WAR) files.
- Modify the project’s Platform Profile and Application Services when you want to update an Extension Service project’s build path. You can also update the Java Build Path to include other Extension Services projects or projects that contain Service Management Framework bundle folders. The project can access only these components while running on an Extension Services platform.

  Note: If you update the Java Build Path in any other way, your build time classpath will not be available to the application at runtime.
- Verify that you selected the appropriate services for your project. If you use an unsupported J2EE feature, the feature is automatically flagged as an error in your code and reported through the Tasks view. In most cases, a ″class cannot be resolved″ error is reported.
- Extension Services projects only support the following set of properties from the project’s Web properties page:
  - You can change the Context Root and Web content folder name.
  - Extension Services supports J2EE level 1.3. It does not support J2EE level 1.2.
  - Extension Services supports JSP tag libraries. However, the specific JSP tag libraries on the **Web Project Features** page have not been tested with Extension Services.
  - Extension Services does not support Struts or JavaServer Faces.

Supported Service Management Framework Bundle Developer features

Extension Services Web projects are Web applications and OSGI bundles. As such, you can use many of the Service Management Framework (SMF) Bundle Developer tools under the following guidelines:

- Application Tools does not support the **SMF Submit Bundle...** operation for Extension Services Web projects. Instead of using this operation, you can run an Extension Services project directly on an Extension Services server using the instructions in “Running and debugging Extension Services Web projects” on page 16.

If you prefer, you can export the project as a Web Application Bundle file, as described in “Exporting Web Application Bundle files” on page 16, and then the file can be imported by an SMF bundle server.

- When a Web Application Bundle (WAB) file is created from any Extension Services Web project, it will not include the bundle folders from other projects, even if they are listed in the `.bndldesc` file.

For more information

For more information about technologies related to Extension Services, refer to the following documentation:

- Extension Services for WebSphere Everyplace Runtime Developer’s Guide

  Refer to this documentation for a description of Extension Services for WebSphere Everyplace, descriptions of Extension Services for WebSphere Everyplace concepts, and instructions to complete tasks with Extension Services for WebSphere Everyplace.
- WebSphere Studio Application Developer documentation

  Refer to the Web development information in this document for more detailed information about the Web development tools.
- Service Management Framework User’s Guide

  Refer to this documentation for detailed information about the services provided with Service Management Framework (SMF) and the SMF Bundle Developer tools.

Setting up the Application Tools environment

Before you use the Application Tools, verify that you have done the following:

- If you want to develop Web projects for the WebSphere Everyplace Custom Environment (WECE), verify that you installed the appropriate WECE runtime for your platform.

  Note: These features do not install by default.
- Before launching the WebSphere Studio tools, insure that the `JAVA_HOME` environment variable is not set. Setting this environment variable to a non-J9 JRE can cause problems when launching J9 runtimes.

Running a sample Web application

Refer to the following instructions for steps to run the Pizza Java Server Page Web application:

1. Select **File**->**New**->**Project**.

   The new project wizard displays.
2. Select **Examples**->**SMF** on the left panel and select **Pizza JSP Web Application** on the right panel. Then click **Next**.

   A description of the application displays.
3. Click **Finish** to create the PizzaJSP project.
4. Convert the PizzaJSP project into an Extension Services project by completing the steps in “Converting a Web project into an Extension Services project” on page 13.
5. Right-click on the project and select **Run on Server...**.

   The **Server Selection** dialog window displays.

6. Under the **Create a new server** dialog, click on the plus sign (+) next to the **Extension Services** server type, and select **Local Server**.
7. Click **Finish**.

   The Extension Services server launches, and the PizzaJSP project installs and runs.

   The Web browser displays the PizzaJSP application's page for ordering a pizza.

   If you want to stop the server, open the Server perspective. Right-click on the local server listed in the Servers view and select **Stop**.

Roadmap of major tasks

This roadmap provides a list of the major tasks you need to complete to create and run an Extension Services Web application.

Table 1. Roadmap of major tasks

Task	**Reference**
If you are using projects or servers created in a previous version of Application Tools for Extension Services, review possible migration actions.	"Migration"
Review the Web development features that Extension Services supports.	"Supported Web development features" on page 2
Review the concepts information to learn about the differences between a Web application and an Extension Services Web application.	Chapter 2, "Concepts," on page 9
Create a new Extension Services Web project, or convert an existing Web project into an Extension Services Web project.	"Creating an Extension Services Web project" on page 13 or "Converting a Web project into an Extension Services project" on page 13
Run an Extension Services project.	"Running and debugging Extension Services Web projects" on page 16

Migration

The following migration instructions cover all of the Extension Services for WebSphere Everyplace related products:

- SMF Bundle Developer Tools
- Extension Services for WebSphere Everyplace
- Application Tools for Extension Services

If you are not using a pre-existing workspace or projects created with a previous version of these tools, you may skip this section. If your usage was limited to the features provided by Application Tools for Extension Services, then you need only read the information in "Migrating Application Tools for Extension Services" on page 8. Otherwise, review the remaining information in this section.

The workspace is the major organization container for the WebSphere Studio suite of tools. Within each workspace, you can create projects to contain code, define servers to test code, create launch configurations to run programs, and many other tasks. For each of the artifacts that you create within a workspace, metadata is created within the project, or within the workspace.

With this release, the installed features include enhancements and changes. Because of the differences, it is strongly recommend that you create a new workspace for project development. This will simplify the number of migration steps that you will need to perform.

If you choose to use an existing workspace, then you will need to review and apply the information contained in “Migrating workspaces” on page 6, “Migrating SMF runtime directories” on page 7 and “Migrating workspace projects” on page 7.

If you choose to create a new workspace, then you may import projects from existing workspaces, or source control systems. You will also need to review and apply the information contained in “Migrating SMF runtime directories” on page 7 and “Migrating workspace projects” on page 7.

What’s new

SMF Runtime

- Externalized message files, resulting in two libraries
- `SMFCoreMsg_<locale>.jar` is added to the classpath
- `SMFBundleMessages` is a bundle providing a service to many other SMF bundles
- Updated bundle store for the SMF Runtime
- J9 2.2 specific command line options

SMF Bundle Developer

- Extension Services Bundle Projects provide for platform profile based project creation

Web Services

- Separation into two features for MIDP and Extension Services. As a result, library file names were changed

Extension Services

- SyncML/DM OSGi Agent replaces Device Agent 1.1
- SyncML 2.0 Framework updated to version 2.0
- DB2 Everyplace updated to version 8.1.4
- ISync database synchronization libraries for DB2 Everyplace and Cloudscape
- Platform Builder Wizard and Editor (upgraded from the Installable Image Build project)
- Calendar Sample
- Two provided Beta services
  - eSWT (includes 2 examples)
  - jsr169 JDBC 3.0 for J2ME

Application Tools for Extension Services

- Server startup options to preserve configuration settings
- Server definition options to define JVM and SMF arguments

What’s changed

The following libraries provided by the Mobile Web Services features have been updated in this release.

Note: The variable names (`WS_RUNTIME, ESWE_BUNDLES`, etc) listed in the tables below can be found via **Windows > Preferences > Java > Classpath variables**.

For information on MIDP library locations, refer to the following table.

Table 2.

Old Location	New Location
WS_RUNTIME\WebServicesWME.jar	WS_RUNTIME_MIDP\WebServicesMIDP.jar
WS_RUNTIME\WS-Security.jar	WS_RUNTIME_MIDP\WS-Security.jar

For information on Extension Services library locations, refer to the following table.

Table 3.

Old Location	New Location
WS_RUNTIME\WebServicesWME.jar	ESWE_BUNDLES\WebServicesME.jar
WS_RUNTIME\WebServicesWCE.jar	
WS_RUNTIME\WS-Security.jar	ESWE_FILES\webservices\WS-Security.jar

The following libraries provided by Extension Services features have been updated in this release.

Table 4.

Old Location	New Location
ESWE_BUNDLES\javasql.jar	ESWE_BUNDLES\database_enabler.jar
ESWE_BUNDLES\db2.jar	ESWE_BUNDLES\DB2e.jar

Note: Extension Services now provides only a single `DB2e.jar` that does not contain native libraries. The native system libraries for DB2e have been installed into the `ESWE_FILES\db2e` directory.

Migrating workspaces

If you attempt to access a workspace created with previous editions of the tools, there are migration steps that must be performed to enable the tools to work within the current workspace.

The first step in migration is to ensure that all projects are up-to-date. Objects known as builders associated with each project will verify that the projects are up to date, or will update the projects automatically. To cause the builders to perform this activity, you should select **Project + Rebuild All**.

Updating program launch descriptions

To update program launch descriptions, perform the following procedure:

1. Select **Run > Run...** from a workspace perspective to view launch configurations.
2. If you have created any instances of the SMF Bundle Server, delete these instances and create new instances if needed. Instances defined using previous editions will have incorrect working directories, VM arguments, and User classpath settings.
3. If you have created any instances of the SMF Runtime applications, the following changes will be required:

- User class path will be incorrect. You will need to add `SMF_CLIENT/SMFCoreMsg_en.jar` to the User Classpath.
- If the SMF Runtime is using a J9 JVM, the `-Xalwaysclassgc` parameter must be added as a VM argument.

4. If you have created Java applications that use libraries from SMF, ESWE, or Web Services, then you will need to update the variables, or the variable extensions to refer to the updated libraries in the classpath or source tabs of the application definitions.

Updating server configurations

Extension Services has provided new platform profiles for this release. These new profiles include additional components, as well as upgrades to existing components. Servers defined using platform profiles from previous editions will not start, and must be updated to current platform profiles. Attempting to start a server using the old platform profiles will result in the following message:

Problems encountered with the application profile. Referenced platform profile is undefined.

The Tasks window will identify server definitions that contain undefined profile references. The Java or SMF perspectives will also identify via the error marker those server configurations that need to be updated.

To update the Server configurations, switch to the SMF perspective, and go the Servers folder. For each of the folders (server instances) marked as an error, locate the <folder name>.es file, and open the file using the Extension Services editor. Go to the Profile tab, and select one of the available Platform Profiles.

Migrating SMF runtime directories

The format of the SMF runtime bundles directory has changed, and is not compatible with formats used in previous versions of the runtime.

If you attempt to start SMF using these existing directories, the SMF Runtime will fail with the message, *"Old SMF Framework version 2 detected in bundle directory. Migration of old SMF data is not supported. Unable to continue..."*

For any the SMF Runtime launch configurations that you have created, you will need to remove some or all of the contents of their working directories. The working directory for a specific SMF Runtime can be found on the arguments panel of the runtime configuration window.

If you have not changed the `smf.properties`, or do not have files located in that directory, then remove and recreate the directory.

If you need to preserve information in that directory, then remove the bundles directory from that directory.

Migrating workspace projects

Projects created with previous versions of the tools, whether existing in workspaces, or source control systems, may need to be updated to use the latest library locations and jars. If Existing Java or J2ME projects, with or without SMF Bundle Folders refer to any the libraries listed above, you will need to update the projects to refer to the new library locations and jars. The Tasks window will identify project classpath entries that can no longer be updated.

If you are using WebSphere Studio Application Developer or Site Developer, and the Application Tools for Extension Services for web application development, then you may need to take the following action. If any Extension Services Web Application projects exist in the workspace, and that use any previous versions of Extension Services platform profiles, you will need to select a new platform profile for the project. The Tasks window will identify projects that contain undefined profile references. Services marked as being used in the current profile, except for DB2 Everyplace usage, will be marked in the newly selected profile (if they exist).

The Platform Builder project layout has changed from the previous Installable Image Builder project layout. In addition, files and file locations referenced in previous Installable Image Build project scripts have changed or no longer exist. If you have any Installable Image Builder projects, you will need to create a new project using the new Extension Services Platform Builder wizard.

Migrating Application Tools for Extension Services

To migrate Extension Services projects and Extension Services servers created with a previous version of Application Tools for Extension Services, perform the following tasks.

- If you are migrating a workspace created with a previous version of the tools, you should rebuild all of the projects in the workspace. Select **Project > Rebuild All**.
- If a project has the error *Referenced platform profile is undefined*, follow the steps in "Setting Extension Services project properties" on page 14 to select a new platform profile for the project.
- If a server definition has the error *Referenced platform profile is undefined*, follow the steps in "Editing a server definition" on page 21 to update the server's platform profile.

Chapter 2. Concepts

This chapter contains information about Extension Services concepts.

Service Management Framework

Service Management Framework (SMF) is the IBM implementation of the OSGi Alliance, an open standards platform that supports the development, deployment, and management of services to Internet-enabled devices. SMF provides the core framework for the Extension Services platform and supports a component-based framework that resides on top of Java.

Bundles are OSGi components that you can dynamically install, start, stop, update, and uninstall without restarting the SMF framework. You can manage the bundle life cycle over the Internet, which enables you to manage devices that run SMF over the Internet. For more information about SMF, refer to the Service Management Framework documentation. For more information about the OSGi Alliance, refer to the OSGi Web site at `http://www.osgi.org`.

Extension Services Web project

Extension Services Web projects are Web projects that contain the additional functionality needed to support the Extension Services platform. Because Extension Services projects are Web projects, you can use existing WebSphere Studio Web development tools to develop your application. However, because Extension Services platforms do not support the full J2EE runtime, you cannot use all of the features of WebSphere Studio. Refer to "Supported Web development features" on page 2 for more information.

Extension Services supports Extension Services Web projects as OSGi bundles. The Application Tools automatically handle the OSGi-specific details for Web applications that do not use the underlying OSGi framework APIs.

Note: To understand more about the OSGi Alliance and SMF, refer to the Service Management Framework documentation for more information about SMF and refer to the OSGi Web site at `http://www.osgi.org` for more information about OSGi.

Extension Services Web projects differ from standard Web projects because Extension Services Web projects:

- Contain an Extension Services Content folder.

  This folder contains Extension Services-specific project content, such as:

  - A Manifest file, `META-INF/MANIFEST.MF`, which contains the OSGi metadata for the project.
  - A default bundle activator class.
  - Servlet Java source files that result from the translation of project Java Server Page (JSP) files.
- Automatically update the Java Build Path.

Extension Services automatically updates the project's Java Build Path to reflect the project's Platform Profile and Application Services settings. Refer to "Platform Profile" and "Application Services" on page 11 for more information.

- Automatically update the Manifest file.

  Extension Services automatically updates the Manifest file in the Extension Services Content folder to contain appropriate OSGi metadata for the project. Extension Services manages or provides initial default values for the metadata fields by:

  - Setting `Bundle-Name` to the project name.
  - Setting `Bundle-Version` to `1.0.0`.
  - Setting `Bundle-Activator` to the default bundle activator provided by the Application Tools.
  - Including Web application libraries from the WEB-INF/lib directory in `Bundle-Classpath`.
  - Updating `Import-Packages` to reflect the packages imported by the project's classes.
  - Setting `Import-Services` to include all of the services from the Application Services selected for the project. Refer to "Platform Profile" for more information.
- Provide a default bundle activator.

  Extension Services Web applications require an OSGi bundle activator class to register the Web application with the Web container at runtime. The Application Tools include a default bundle activator class, ExtensionServicesBundleActivator, for this purpose. You can tailor the default bundle activator class by editing the source file for the class in the Extension Services Content folder.
- Incrementally translate JSP files.

  The Application Tools translate the JSP files in Extension Services Web applications to their underlying servlet classes every time you save changes to the JSP files. Translating the JSP files eliminates the need to have a Java compiler on the Extension Services platform.

  Note: The JSP Compilation Validator provided by WebSphere Studio is disabled for Extension Services Web projects. Instead, Extension Services validates the files using the Extension Services incremental JSP translator.

Platform Profile

Typical Java applications build and run against a static classpath. In WebSphere Studio, you can specify this classpath through the project's Java Build Path properties. Under Service Management Framework (SMF), the base class libraries of the Java Runtime Environment (JRE) and the SMF bundles currently running in the framework are the packages available to an application's classloader. In other words, the JRE and the set of running bundles define the application's available classpath.

The Platform Profile provides a method for you to define the JRE used by the runtime environment, the build-time environment for an Extension Services project, and the set of bundles that can run on the platform.

A Platform Profile defines a set of Application Services. An Application Service represents a logical service, such as an XML parser or logging service, that consists of one or more bundles. Application Services enable you to focus on the logical service requirements of the service instead of the requirements of the actual underlying bundles.

When you create an Extension Services project, you select a Platform Profile and a set of Application Services for the project. Extension Services updates the Java Build Path for the project to reflect the Java Runtime Environment (JRE) of the platform, based on the Platform Profile you selected. Extension Services places the bundles for the Application Services you selected in the Java Build Path of the project. "Application Profile" refers to the Platform Profile and the set of Application Services you selected for the project.

The Application Tools provide Platform Profiles that represent the bundles available in SMF, for WebSphere Everyplace Micro Environment Foundation, WebSphere Everyplace Custom Environment Resource Managed Gateway Plus, and WebSphere Everyplace Custom Environment Max. The Application Tools also provide an Eclipse Default JRE platform profile that uses the currently defined Eclipse Default JRE. You can make additional Platform Profiles available to the Application Tools through the `com.ibm.smf.tools.project.platformprofilerepository` extension point. For details on the services provided by these bundles, refer to the Service Management Framework documentation.

Application Profile

Application Profile refers to the Platform Profile and the set of Application Services that you selected for the project. The Platform Profile and Application Services you select define the runtime and build time environment for the project. Extension Services updates the Java Build Path for the project to reflect the Java Runtime Environment (JRE) of the platform, based on the Platform Profile you selected. Extension Services places the bundles for the Application Services you selected in the Java Build Path of the project.

The Application Tools use the project's Application Profile when the project runs on a server. If you test a project on a local server, the server runs on the JRE specified in the Application Profile, and installs and runs the project and the bundles associated with the Application Services you selected. As such, Extension Services tests the project against the runtime environment defined by the Application Profile for the project.

Application Services

An Application Service represents a logical service, such as an XML parser or logging service, that consists of one or more bundles. Application Services enable you to focus on the logical requirements of the service instead of the requirements of the actual underlying bundles. The Applications Services and the Platform Profile you selected are referred to as the Application Profile.

Web Application Bundle

Web Application Bundles are OSGi bundles that contain Web applications. Web Application Bundles are the Extension Services version of a Web Application Archive (WAR).

The Application Tools transform Extension Services projects into Web Application Bundle files before they are run on an Extension Services server. As such, the Service Management Framework manages Web applications as bundles.

Server

Extension Services supports local and remote servers. The Application Tools launch and control the local server on the local development machine. *Remote server* refers to an Extension Services platform that is already running locally on the development machine or on a remote machine. When you begin to develop an application, you might choose to use a local server for testing on the development machine, and then use a remote server to test on the actual target device.

The Platform Profile associated with a server definition defines the Java Runtime Environment (JRE) and a set of required runtime services for the server. The Platform Profile considers projects and servers to be compatible when the project's Platform Profile matches the profile of the server.

An Extension Services server consists of two components:

- Service Management Framework (SMF) Runtime

  A Service Management Framework runtime.
- SMF Bundle Server

  An application manager that maintains a repository of bundles and manages the deployment of these bundles to the runtime.

For more information about the SMF Runtime and the SMF Bundle Server, refer to the Service Management Framework documentation.

Chapter 3. Tasks

This chapter contains information for completing Extension Services tasks.

Working with Extension Services Web projects

This section instructions on working with Extension Services web projects.

Creating an Extension Services Web project

Complete the following steps to create a new Extension Services Web project:

1. Select **File**->**New**->**Project**.

   The new project wizard displays.
2. Select **Extension Services** on the left panel and **Extension Services Web Project** on the right panel. Then click **Next**.

   The **Create an Extension Services Web project** panel is displayed.
3. Specify a project name in the **Project name** field. You can also specify a name for the context root if you want the context root to be different from the project name.
4. Click **Next**.

   The second **Create an Extension Services Web project** panel is displayed.
5. Select a **Platform Profile** from the pull-down menu.

   A description of the platform profile you selected displays in the **Description** field.

   Note: The project creates the Java Build Path from the platform profile and application services you select.
6. Select the boxes next to the **Application Services** you want to use.

   Note: If the platform that you selected in step 5 requires specific services, the services are automatically selected in the **Application Services** field. You can select any additional services that the application uses.
7. Click **Finish**.

   Extension Services creates an Extension Services Web project.

Converting a Web project into an Extension Services project

Complete the following steps to convert a Web project into an Extension Services project:

1. Select **File**->**New**->**Other...**.

   The **New** project wizard displays.
2. Select **Extension Services** on the left panel and **Convert Web to Extension Services Project** on the right panel. Then click **Next**.

   The **Convert Web to Extension Services** panel displays.
3. Select the Web project that you want to convert.

   If you want to create a copy of the project before you convert it, select the **Copy before creating** box and provide a name for the new project. When the new project converts, the original project will remain unchanged.

Note: You cannot convert an Extension Services Web project back to a Web project. If you want to retain the Web version and the Extension Services Web version of the project, select the **Copy before creating** box.

4. Click **Next**.

   The second **Convert Web to Extension Services** window displays.
5. Select a **Platform Profile** from the pull-down menu.

   A description of the platform profile you selected displays in the **Description** field.

   Note: The project creates the Java Build Path from the platform profile and application services you select.
6. Select the boxes next to the **Application Services** you want to use.

   Note: If the platform that you selected in step 5 requires specific services, the services are automatically selected in the **Application Services** field. You can select any additional services that the application uses.
7. Click **Finish**.

   Extension Services converts the Web project into an Extension Services project.

Setting Extension Services project properties

You can modify the platform profile and application services for the Extension Services project on the properties page for the project. To modify the project properties, complete the following steps:

1. Right-click on your project in the Navigator view and select **Properties**.
2. Select **Extension Services**.
3. When the Extension Services properties page displays, select the **Application Profile** tab.
4. Select the platform profile you want to use from the pull-down menu.
5. Select the application services that you want to add and deselect the application services you want to remove.
6. Click **OK** to commit your changes.

   After the Java Build Path and Manifest file update to reflect your changes, your project rebuilds against the new Java Build Path.

You can modify which aspects of your project are automatically managed by the tools. It is suggested that the tools be allowed to manage the following items (the default settings) unless special circumstances dictate otherwise.

Bring up the Extension Services properties page, as described in steps 1 and 2 above. Select the **Options** tab.

You can select the following options:

- **Use Platform Profile to manage Java Build Path**

  When selected, the project's Java Build Path is automatically updated to reflect the platform profile and application services selected for the project. The Java Runtime Environment (JRE) of the platform profile is placed on the project's Java Build Path. The bundles associated with the selected application services are also placed on the Java Build Path. When deselected, the tools will not automatically update the project's Java Build Path.
- **Automatically manage Import-Package key in Bundle Manifest**

When selected, the `Import-Package` key in the Manifest file is automatically updated to reflect the packages imported by the Java code in your project. When deselected, the tools will not automatically update the `Import-Package` key in the Manifest file.

Restoring an Extension Services Web project environment

The Application Tools manage the Extension Services project environment automatically by:

- Updating the Java Build Path to reflect the project's platform profile and application services.
- Maintaining Extension Services metadata in the Manifest file.
- Generating a bundle activator class.

If you modify any of these settings so that they no longer function properly or if you want to restore any of these settings to their original state, complete the following steps:

1. Right-click on your project in the Navigator view and select **Extension Services**->**Restore Environment**.

   The **Restore Extension Services Environment** window is displayed.
2. Refer to the following descriptions to decide which environment settings you want to restore. Then select the box next to each setting that you want to restore.

   Restore Java Build Path
   Updates the Java Build Path based on the project's platform profile and application services.

   Restore Bundle Activator
   Creates a default bundle activator for the project and overwrites any changes you made to the default bundle activator.

   Restore Bundle Manifest
   Updates the Extension Services metadata for the Manifest file.
3. Click **OK** to restore the settings you selected.

Importing and exporting Web Application Bundles

Refer to "Importing a Web Application Bundle file" for instructions to import Web Application Bundles. Refer to "Exporting Web Application Bundle files" on page 16 for instructions to export Web Application Bundles.

Importing a Web Application Bundle file

You can import Web Application Bundle files that were exported with source using the Web Application Bundle Export Wizard. However, Extension Services does not support importing Web Application Bundle files that were created by other methods.

Note: If you choose to import the Web Application Bundle into an existing project, files in the Web Application Bundle will overwrite any files in the project that have the same name as those in the bundle. For example, the project's Web deployment descriptor is overwritten by the Web Application Bundle's deployment descriptor. By default, you are prompted to confirm which files will be overwritten, and you can select that specific files not be overwritten.

Complete the following steps to import a Web Application Bundle file:

1. Open the Import wizard by selecting **File**->**Import...**.

   The **Import** window displays.
2. Select **WAB file**, and click **Next**.

   The **Import Resources from a WAB file** window displays.
3. Select the WAB file that you want to import by selecting **Browse** or type the file name into the **WAB file** field.
4. Specify whether you want to import the file into a **New** or **Existing** Extension Services project.

   If you choose **New**, specify a name for the new project in the **New project name** field.
5. If you want the context root to be different from the project name, specify a name for the context root in the **Context Root** field.
6. Select the check box if you want to overwrite the existing project resources without warning.
7. Click **Finish** to import the Web Application Bundle.

Exporting Web Application Bundle files

Complete the following steps to export a Web Application Bundle file:

1. Open the Export wizard by selecting **File**->**Export...**.

   The **Export** wizard displays.
2. Select **WAB file**, and click **Next**.

   The **Export Resources to a WAB File** window displays.
3. Select the Extension Services project that you want to export from the drop-down list.
4. Specify a destination location.
5. Check the **Export source files** box if you want to include the Java source files from the project.

   Note: If you want to import the Web Application Bundle file into another project, you must include the source files.
6. Select the **Overwrite existing files without warning** box to overwrite existing files without warning.
7. Click **Finish**.

   Extension services exports the Extension Services project you selected as a WAB file to the destination you specified.

Running and debugging Extension Services Web projects

Extension Services enables you to run and debug Extension Services Web projects.

To debug a project, follow the instructions for running a project and select **Debug on Server...** instead of **Run on Server...** in the first step.

Use one of the following methods to run an Extension Services project and restart the server every time you run a new project:

- "Running a Web project on the default local server" on page 17 enables the Application Tools to create a default server for your project to run on. The Application Tools automatically define a local server based on the JRE and the application services in the project's application profile.

- “Running a Web project using an existing server definition” on page 18 enables you to select a server that has already been defined to run the project on. You might chose the default server definition that was automatically created by the Application Tools (refer to the previous option), or you can use a server definition that you created.

  To run a Web project on a local server, select a local server definition. To run a Web project on a remote server, select a remote server definition.

  Note: Refer to “Defining a server” on page 20 for information to define local and remote servers.
- “Running a Web project by defining a new server” on page 18 enables you to define the server at the same time you request to run the project on a server.

“Running multiple projects on the server” on page 19 describes how to run multiple projects without restarting the server.

“Reloading a project on the server” on page 19 describes how to reload a project on the server without restarting the server.

As described in “Server” on page 12, an Extension Services server consists of a Service Management Framework (SMF) runtime and an SMF bundle server. “Accessing the server through the SMF perspective” on page 19 describes how you can access the underlying components of the server through the SMF perspective.

The following constraints are unique to running an Extension Services Web project:
- You can only run Extension Services Web projects on an Extension Services server. Refer to “Server” on page 12 for more information about Extension Services servers.
- The Application Tools for Extension Services can only run one Extension Services server at a time.
- Extension Services does not support profiling.
- When you start a local server, the runtime’s bundle data repository resets. As a result, any bundle data that was stored the last time the server ran is erased, such as any Configuration Admin settings. If losing this data creates problems in your test environment, test with a remote server instead of a local server. When you use a remote server, you can control how and when the runtime starts.

Running a Web project on the default local server

Complete the following steps to run an Extension Services Web project on the default local server:

1. Right-click on the Extension Services project you want to run, and select Run on Server... to run your project on the server. Or, right-click on one of the resources within the project, such as a Java Server Page (JSP) or Hyper Text Markup Language (HTML) file, and select Run on Server... to run your project on the server.

   Note: If you select a resource, Extension Services sets the initial location of the Web browser to that resource.

   The Server Selection window displays.
2. Under the **Create a new server** dialog, click on the plus sign (+) next to the **Extension Services** server type, and select **Local Server**.
3. Click **Finish**.

After you click Finish, Extension Services launches the server, runs the project on the server, and opens a Web browser window.

If you chose a project, the Web browser references the context root of the project as the location. If you chose a resource instead of a project, the Web browser references that resource as the initial location.

Running a Web project using an existing server definition

Complete the following steps to run an Extension Services Web project using an existing server definition:

1. Right-click on the Extension Services project you want to run, and select **Run on Server...** to run your project on the server.

   Or, right-click on one of the resources within the project, such as a Java Server Page (JSP) or Hyper Text Markup Language (HTML) file, and select **Run on Server...** to run your project on the server.

   Note: If you select a resource, Extension Services sets the initial location of the Web browser to that resource.

   The **Server Selection** window displays.
2. Select **Use an existing server**.
3. Select a server definition from the list.

   Note: The list displays only Extension Services servers that have the same platform profile as your project.
4. Click **Finish**.

   After you click **Finish**, Extension Services launches the server, runs the project on the server, and opens a Web browser window.

   If you chose a project, the Web browser references the context root of the project as the location. If you chose a resource instead of a project, the Web browser references that resource as the initial location.

Running a Web project by defining a new server

Complete the following steps to run an Extension Services Web project by defining a new server:

1. Right-click on the Extension Services project you want to run, and select **Run on Server...** to run your project on the server.

   Or, right-click on one of the resources within the project, such as a Java Server Page (JSP) or Hyper Text Markup Language (HTML) file, and select **Run on Server...** to run your project on the server.

   Note: If you select a resource, Extension Services sets the initial location of the Web browser to that resource.

   The **Server Selection** window displays.
2. Select **Create a new server**.
3. Select the folder where you want your new definition to reside.
4. In the Server type view, expand Extension Services, and select **Local** or **Remote** server.
5. Click **Next**.

   The window for configuring a server displays.
6. Refer to Table 5 for information to complete the server configuration fields.

After you click **Finish**, Extension Services launches the server, runs the project on the server, and opens a Web browser window.

If you chose a project, the Web browser references the context root of the project as the location. If you chose a resource instead of a project, the Web browser references that resource as the initial location.

Running multiple projects on the server

You can run a project on a server by using the project's **Run on Server...** operation from the project's context menu. However, using the **Run on Server...** operation restarts the server for every project that you run.

To avoid restarting the server for every project, refer to the following steps to add projects to a server that is already running. You can also complete the following steps to add projects to a server that is not running. If the server is not running, the projects run when the server starts.

To add projects to a server:

1. Open the Server perspective.
2. In the Server Configuration view, locate the server you are running in one of the server folders.
3. Right-click on the server you are running and click **Add and Remove Projects**.

   A list of the available Extension Services projects displays.
4. Select the project that you want to add, and click the **Add** button. The project is added as one of the server's configured projects.
5. Click **Finish**. The project is run on the server.

Note: To view the projects that are already associated with this server, expand the server by clicking on the plus sign (+) next to the server. The projects listed below the server automatically run every time the server starts. To remove a project from this list, right-click on the project you want to remove and select **Remove**.

Reloading a project on the server

Typical project testing consists of running the project on the server, observing any problems, correcting the problems, and running the project again to verify that you corrected the problems. If the project is already running on a server, you can reload the project without restarting the server by using the **Run on server...** operation.

Note: The **Run on Server...** operation reloads all of the projects associated with the server.

Accessing the server through the SMF perspective

You can access any Extension Services servers that you start through the Application Tools in the SMF perspective. You can access the underlying server components in the SMF Bundle Servers view and the SMF Runtime view. For more information about the tasks you can complete in the SMF perspective, refer to the SMF Bundle Developer information in the Service Management Framework User's Guide.

Managing servers

You can complete the following tasks to manage servers:

- "Defining a server"
- "Editing a server definition" on page 21
- "Starting and stopping a server" on page 21

Defining a server

Complete the following tasks to define a server:

1. Select **File**->**New**->**Other...**.
2. Select **Server** on the left panel and select **Server and Server Configuration** on the right panel.
3. Click **Next**.

   The **Create a New Server and Server Configuration** window is displayed.
4. Specify a name for the server and select the folder where you want the server definition to reside.
5. In the Server type view, expand Extension Services, and select **Local** or **Remote** server.
6. Click **Next**.

   The second **Create a New Server and Server Configuration** window displays.
7. Refer to the following table for descriptions of the server settings you can modify. In most situations, you can use the default values. However, if you connect to a remote runtime or bundle server, you might need to modify some of the settings on the **Server Options** window. When you use a remote runtime, it is likely that the only value you need to change is the Hostname.

Table 5. Server configuration setting descriptions

Setting	Definition
Runtime options	
Hostname	Represents the hostname or IP address of the machine on which the runtime is running. For a local server, this value is set to localhost.
Port	Represents the port that the runtime and the Application Tools use to communicate. For a remote server, this value must match the -ide port option for the remote runtime.
Clear runtime bundles and data before launching	Select this option to clear the runtime of bundles and data from previous launches.
Bundle server options	
Use a remote Bundle Server	If you are using a remote bundle server , select this option. For remote servers, you can use either a local or remote bundle server.
Hostname	Represents the hostname or IP address of the machine on which the bundle server is running. For a local bundle server, this value is set to localhost.
Port	Represents the port to use for bundle server requests.
URI	Represents the Uniform Resource Identifier (URI) to be used for bundle server requests.

Table 5. Server configuration setting descriptions (continued)

Setting	Definition
UserID	Represents the user ID for bundle server requests. The bundle server can assign different capabilities to different users. The default user ID is `Admin`, which provides full access.
Password	Represents the password associated with the user ID. The password for the default `Admin` user ID is `Admin`.
Debug options	
Connection type	Represents the type of debugging connection to use. Specify `J9` for J9 Java Virtual Machines (JVMs) or specify `Standard` for standards JVMs.
Port	Represents the debugging port.

8. Click **Next**.

   The third **Create a New Server and Server Configuration** window displays.
9. Select a platform profile to associate with this server and any additional application services you want to run automatically when the server starts.

   When you run the server, the runtime on the server uses the Java Runtime Environment (JRE) that is specified in the platform profile and automatically loads any required runtime services specified in the platform profile and the application services you selected.
10. Click **Finish**.

    Extension Services saves the server definition.

Editing a server definition

Complete the following tasks to edit a server definition:

1. Double-click on the server you want to edit in the Server Configuration view of the Server perspective.

   The server options display in the editor window. You can select the **Profile** tab to display and edit the Platform Profile settings. You can select the **Server** tab to display the server settings again so that you can edit them.

   For a local server, you can select the **Environment** tab to add additional launch arguments to the Java VM and SMF.

   Note that the tools already handle setting the appropriate Java and SMF options to launch the Extension Services Server based on the server configuration. One use of the environment settings is to set application specific properties through the Java -D option.
2. Modify the server options, environment settings, and the platform profile settings that you want to change.
3. Save the updated server definition.

Starting and stopping a server

Servers usually start in response to running a project. However, you can choose to start and stop a server without running a project. Refer to the following for instructions:

1. Open the Server perspective.
2. Right-click on the server you want to start or stop in the Servers view.
3. Select **Start** to start the server.

Select **Stop** to stop a server that is already running.

Select **Restart** to stop and restart a server that is already running.

Chapter 4. Reference

This chapter contains reference and problem determination information specific to Application Tools for Extension Services.

Platform profiles

For information on how to create a platform profile and contribute it to the tools, refer to Appendix B of the **SMF Bundle Developer Tools User's Guide**.

Problem determination

Consult the following table for problem determination information.

Table 6. Problem determination

Symptom	Solution
The **Convert Web to Extension Services Project** wizard does not list a web project as selectable for conversion.	Check that the web project is a J2EE 1.3 project. This can be done through the web project properties page. J2EE 1.2 projects are not supported for converting to Extension Services projects. You must first convert the project to a J2EE 1.3 project using the web tools. Refer to the **Migrating from J2EE 1.2 to 1.3** section under Support Information in the online help.
The following error is reported in the Tasks view when saving a JSP file: *JavaCompile: Code of method _jspService is exceeding the 65535 bytes limit* The following error is reported in the web browser when running an application: *Unable to compile class for JSP*	A JSP file in the application is too large to be processed. It is possible for a large JSP file, especially one with many custom tags, to generate a method that exceeds the Java VM 65535 byte code length limit. To avoid this error, restructure the large JSP file to reduce its size.
The following errors are listed in the Tasks view: • *JavaCompile: The import java.beans cannot be resolved* • *JavaCompile: The import java.sql cannot be resolved* • *JavaCompile: Beans cannot be resolved*	These errors are caused by the web JSP Compilation Validator being enabled on an Extension Services project. This validator is normally replaced with an Extension Services validator when the project is created, but there are ways in which it can become enabled. Follow these steps to disable this validator: 1. Right-click on your project in the Navigator view and select **Properties**. 2. Select **Validation**. 3. Select **Override validation preferences** and deselect **JSP Compilation Validator**. 4. Click **OK**.

Table 6. Problem determination (continued)

Symptom	Solution
Error parsing web.xml error received when running a project or exporting the project as a WAB.	Check the DOCTYPE statement in the project's web.xml file. A DOCTYPE that specifies DTD Web Application 2.2, will cause this error. This indicates a J2EE 1.2 application, and Extension Services only supports J2EE 1.3 applications. You can convert the web.xml file to a J2EE 1.3 compliant file by using the J2EE Migration Wizard described in section **Migrating from J2EE 1.2 to 1.3** under Support Information in the online help. After running the J2EE Migration Wizard, you must restore the Extension Services project's Java Build Path as follows: 1. Right-click on your project in the Navigator view and select **Extension Services > Restore Environment**. 2. Select **Reset Java Build Path**. 3. Click **OK**.
Error tags remain in the JSP editor after fixing the problem in your Java code.	JSP errors that are caused by problems in the project's underlying Java code are not automatically reset after fixing the problem in the Java code. To clear these errors, either re-save the JSP file in the editor or perform a Rebuild Project operation.
A project or server definition has a *Referenced platform profile is undefined* error.	The project or server definition was created with a previous version of the tools, and its platform profile is no longer defined in the current version of the tools. For a project, follow the steps in "Setting Extension Services project properties" on page 14 to select a new platform profile for the project. For a server definition, follow the steps in "Editing a server definition" on page 21 to update the server's platform profile.

Appendix. Notices

This information was developed for products and services offered in the U.S.A. IBM might not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM might have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the information. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this information at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Department LZKS
11400 Burnet Road
Austin, TX 78758
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, or other countries, or both:

IBM

Other company, product or service names may be trademarks or service marks of others.