WebSphere® Studio Device Developer

Extension Services Developer's Guide

Second Edition (March 2004)

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

Contents

Chapter 1. Getting started

Before you use Extension Services

Before you use Extension Services, you should be familiar with the following:

- Java™ programming
- XML tagging
- Ant build tool
- Service Management Framework™ (SMF) and OSGi

  For more information about SMF, refer to the Service Management Framework documentation. For more information about OSGi, refer to the OSGi Alliance Web site located at `http://www.osgi.org`.

Some Extension Services bundles require the use of native system libraries. To successfully access these libraries, WebSphere Studio must be able to locate these libraries. Perform the following steps to set up the development environment for Extension Services:

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

   ```
   WSDD_HOME\wsdd5.0\technologies\smf\client\nativeLib;WSDD_HOME\wsdd5.0\technologies\eswe\files\
   ```

   For Linux, add the following entry to the end of your LD_LIBRARY_PATH environment variable:

   ```
   WSDD_HOME/wsdd5.0/technologies/smf/client/nativeLib:WSDD_HOME/wsdd5.0/technologies/eswe/files/
   ```

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

Prerequisites

Extension Services requires the following software:

- WebSphere® Studio Device Developer 5.7
- Service Management Framework Bundle Development Kit 5.7
- WebSphere Everyplace Custom Environment Max class library 5.7
- WebSphere Everyplace Custom Environment RM Class library 5.7

  Notes:

  1. You only need to install the WebSphere Everyplace Custom Environment Max or RM class libraries if you want to build Max or RM platforms for certain devices.

To take advantage of all the capabilities of Extension Services for WebSphere Everyplace® 5.7.0 Runtime, you can install the Application Tools for Extension Services. Refer to the Application Tools for Extension Services 5.7.0 Application Developer's Guide for information about the prerequisites.

Migration

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 SMF runtime directories" on page 4 and "Migrating workspace projects" on page 5.

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 workspaces" on page 3, "Migrating SMF runtime directories" on page 4 and "Migrating workspace projects" on page 5.

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 1.

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 2.

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 3.

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.

Contents

Extension Services for WebSphere Everyplace contains documentations, tools, and samples. This feature extends the capabilities delivered by the SMF Bundle Development Kit to provide additional runtime components for enterprise applications.

Runtime components

These runtime components deliver bundles that reside in the `WSDD_HOME\wsdd5.0\technologies\eswe\bundlefiles` directory, where `WSDD_HOME` is the location where WebSphere Studio Device Developer is installed. These bundles have also been added to the SMF Bundle Server repository.

- Database Enabler - Java Database Connectivity (JDBC) 2.0 interfaces
- DB2 Everyplace - Small footprint embedded relational database
- DB2 Everyplace Sync - Synchronization capability for DB2 Everyplace database with DB2 Everyplace Sync Server
- DB2 Cloudscape - 100% pure Java embedded relational database
- DB2 Cloudscape Sync - Synchronization capability for DB2 Cloudscape database with DB2 Everyplace Sync Server
- Enterprise Components - Libraries providing enterprise interfaces, including Java Message Service (JMS) 1.1
- MQ Everyplace - 100% pure Java messaging capabilities for clients
- MQ Telemetry Transport - Small footprint publish/subscribe capabilities for embedded devices (sensors, actuators, controllers, etc.)
- SyncML - OMA SyncML/DM and /DS protocol implementation
- SyncML/DM OSGi Device Agent - Management agent that interacts with the Tivoli Device Management Server component
- Web Services - Client and hosting capabilities for web services

The following runtime components are beta level and must be separately downloaded from Extension Services. When installed, these beta components also reside in the `WSDD_HOME\wsdd5.0\technologies\eswe\bundlefiles` directory, however, they are not automatically added to the SMF Bundle Server repository. See the Service Management Framework Bundle Developer Tools User's Guide for information on how to import bundles into the SMF Bundle Server.

- Embedded SWT (beta) - Small footprint subset of the SWT library
- JSR 169 (beta) - JDBC 3.0 interfaces for J2ME Foundation profile

For convenience, the Java classpath variable, `ESWE_BUNDLES`, has been defined to refer to this directory. If you are not using an Extension Services or Extension Services Web Application project, bundles present in this directory can be referred to by using the `ESWE_BUNDLES` variable.

Native system libraries are required for some of the bundles. These libraries have been copied to the `WSDD_HOME\wsdd5.0\technologies\eswe\files` directory, organized by bundle, operating system, and processor. (See "Before you use Extension Services" on page 1 for more information on setting up your system path).

Bundle file names, along with other reference information, for all of these components can be found in "Platform and bundle descriptions" on page 25.

Tools

This section describes the tools provided with Extension Services.

- **Platform profile**

  Extension Services for WebSphere Everyplace Runtime provides three Platform Profiles: one for the WebSphere Everyplace Micro Environment (WEME) Foundation and others for WebSphere Everyplace Custom Environment (WECE) Max and RM. The Platform Profile defines the application services available for each of the profiles. Platform Profiles also include the services provided by Service Management Framework (SMF) and the services included with the Extension Services for WebSphere Everyplace Runtime. Refer to the Application Tools for Extension Services Developer's Guide for detailed descriptions of Platform Profile and Application Services.
- **Target Platform**

  A target platform provides the base Extension Services for WebSphere Everyplace components and functions for specific devices. For example, a platform might include a Java Virtual Machine (JVM) and Service Management Framework (SMF) with a bundle management capability, or the platform might include a JVM, SMF with multiple pre-installed bundles, a management agent, and other custom files.

  After you create a target platform, you can copy the contents of the platform onto a device, which creates an environment in which Java applications and bundles can run.

Samples

This section describes the samples provided with Extension Services.

- **Calendar**

  The Calendar sample is a calendar/scheduling application that makes use of servlet and database technologies.

- **Order Entry**

  The Order Entry sample is a sample provided within Extension Services for WebSphere Everyplace in previous releases, but that is now provided as an additional installable feature. This sample application uses web application, database, and messaging technologies to construct an application that can run on a range of devices.
- **Embedded SWT Controls**

  The Embedded SWT Controls sample is an application that demonstrates the available Embedded SWT controls. It can be run in WebSphere Studio or on a Windows or Windows Mobile 2003 device. It is available when the Embedded SWT beta feature is installed.
- **Embedded SWT TextEditor**

  The Embedded SWT TextEditor sample is an SWT text editor application that can be run on a Windows Mobile 2003 device. It is available when the Embedded SWT beta feature is installed.

Supported platforms

Refer to the following list for the operating system and processor combinations for which you can build a target platform:

Table 4. Platform operating systems and processors

Device	Operating system	Processor
PC	Windows® 2000	x86
PC	Windows XP	x86
PC	Red Hat Linux Pro 8	x86
PocketPC	Windows Mobile 2003	StrongARM/XScale
Zaurus	Embedded Linux	StrongARM/XScale

Next steps

Developing Extension Services applications

For information on developing Extension Services applications, refer to the Getting Started section of the **SMF Bundle Developer Tools User's Guide**, Getting Started section of the **SMF Runtime User's Guide**, and the Samples > Calendar section of the **Extension Services Developer's Guide**.

Creating an Extension Services platform

For information on creating an Extension Services platform, refer to the Tasks section of the **Extension Services Developer's Guide**, the **Creating a Plaform Builder** project section, and the Calendar section of the **Samples** section.

Using the Order Entry example

Order Entry is delivered as a separate installable WebSphere Studio Device Developer (WSDD) feature, Extension Services Samples, which requires WSAD or WSSD 5.1.x and the Application Tools for Extension Services feature. Once installed, refer to the **Extension Services Developer's Guide > Samples > Order Entry** section.

Chapter 2. Concepts

What is Extension Services

Extension Services enables you to develop applications for a wide range of platforms/devices. It extends the capabilities provided by the SMF Bundle Development Kit to provide additional enterprise capabilities.

Applications built on Extension Services are constructed of JAR files providing components -- packages and services -- running within the Service Management Framework environment, which is an implementation of the OSGi standards. The packages, services, and framework all execute within a single JVM instance on a platform. The Service Management Framework is management ready, providing access to management agents to install, update, and remove components within the framework, and to start and stop applications as required.

Application Considerations

Extension Services applications are designed in much the same way as standard enterprise applications. However, the environment in which these applications will execute has different capabilities than one might expect of applications running on a single node in an enterprise system. Considerations unique to applications enabled by Extension Services include the following:

- **Topology**

  The Extension Services platform is intended to enable the construction of distributed applications. Each node (device) in the application can have unique characteristics. These characteristics include display resolution and input capabilities, available resources (processor speed, available memory, limited/removable storage), and network capabilities (always, occasionally, or rarely connected, reliable or unreliable connections, transfer rates, cost of the connections, etc.)

  As an example, consider the typical web page. While suitable for an 800 x 600 display, it may not be suitable for a 320 x 200 display. Keyboards may be acceptable input devices on laptops or desktops, but pointing devices or voice capabilities may be better suited for devices with tiny, or pointing device activated keyboards. Large image download may be acceptable for Wi-Fi or fixed LAN connectivity over broadband devices since it is generally a reliable connection with a monthly flat pricing model, but if one is paying per byte over a cell phone modem, it may be very costly and time consuming to make that transfer.
- **Business logic**

  When building distributed applications, one must consider where business logic is distributed. If always connected over a reliable network, a browser accessing a web application may be sufficient. However, if the user may not always be able to connect to the server, whether traveling via airplane, or in areas without acceptable wireless coverage, being able to locate business logic on the client is crucial. The amount of business logic that is distributed must be sufficient to perform all the work that is required, but needs to be balanced with the requirements of how often it must be updated.

Applications may provide multiple levels of capability, reserving some capability for when a reliable connection exists, and disabling that capability if the server is unavailable.

- **Persistence**

  Most applications manipulate data. This can take the form of read-only access of databases to retrieve catalog items, or of database update for creation of orders that need to be processed. Data can take the form of files distributed on disk, or relational database capability. When dealing with databases, one can choose to use databases only as a local data repository, or as a repository that actively synchronizes with another node in the topology. In either case, if data needs to be distributed to a database, considerations of how much data needs to be distributed and when (once only at initialization, one way from one node to another only on an infrequent basis, frequent exchange between nodes), need to be balanced with storage capabilities at each node in question, and the networking requirements that would permit the exchange to take place. In addition, if synchronization is used, applications should consider database organization, filtering, and conflict resolution policies. Synchronization is useful for exchanging the current state of data between nodes, where transaction boundaries or the order of state changes are not important.
- **Messaging**

  Messaging can take various forms, whether a plain socket-based application, web services, or a more sophisticated store and forward capability that allows for connected as well as disconnected usage. Messaging provides a convenient mechanism for defining, or identifying, transaction boundaries when performing such actions as creating or updating orders, particularly if the transaction requires updated across multiple resources. (Certain nodes in the end-to-end system may not be able to manage or commit transactions since they may not have transaction coordination, and they do not have the master copy of all of the data).
- **Management**

  Management covers a wide range of activities, from initial device provisioning, to application management. When considering management, one must also consider network capabilities, and whether one should download a full new copy of the application, or whether the application has been componentized sufficiently to only distribute a few components. Also important for consideration is how data present on any given node is affected -- if I update the application, will it continue to process existing data, do I need to update data formats, or database schemas to conform to the new application level.
- **Serviceability**

  Distributed applications pose additional issues of serviceability as compared to applications running on a single node. Logging and problem resolution may be difficult if the application is running on one node, and node is only occasionally connects to a central logging repository. How does one transfer logging information from the node in question to the node hosting the central logging repository, to track user usage, or help in problem resolution?
- **Interaction**

  When constructing an application, one needs to decide what type of user interface, if any, will be used. Is there an existing web application that needs to be duplicated on another node because of connectivity? If so, then moving the same web-based application to that node can reduce training and maintenance costs. Other considerations that may affect user interface are the device characteristics, many of which were discussed in the section on Topology.

Additionally, Extension Services applications can be used in embedded environments, such as controllers, where no user interface is required.

This is not necessarily an exhaustive list, and comprehensive of all possible decision points, but it provides some items that may need to be considered when developing client applications.

Capabilities

Extension Services provides enterprise components that can be used in solutions for the following situations:

- **Platform**

  The foundation of an Extension Services based application is the JVM. When building an application, you should consider the target operating systems, as well as the resource constraints, and the classes required to build an application.

  IBM's J9 JVM uses common JVM runtime technologies, but provides a selection of class library capabilities for applications to use. The J9 class libraries are suitable for a wide range of devices, from a small footprint embedded device to laptop and desktop equivalent devices. The OSGi Minimum Execution Environment (OSGi MEE) is a definition of the smallest configuration that is sufficient for an Extension Services application. The OSGi MEE defines a core set of Java class library capabilities, such as Java language capabilities, networking, and I/O. The Resource Management (RM) class library is the implementation of the OSGi MEE, and is a separately installable feature from the update site. The Max class library is another custom class library configuration that is larger than the OSGi MEE, but smaller than a J2SE class library. It extends the OSGi MEE by providing additional capabilities within the Java class libraries, such as additional collection classes, and GUI capabilities.

  The J2ME Foundation profile is suitable for a wide range of capabilities that are more typically associated with server-hosted applications. The Personal Profile includes beans and AWT classes that are typically associated with client GUI applications. Applications developed against the RM class library will run without changes on a J2ME Foundation profile.

  J2SE can also be used as a basis for Extension Services applications, but is typically used in environments that have faster processors and larger memory resources.

  When developing an Extension Services application, the best practice is to identify the smallest possible class library, and to try to restrict application development to only those classes provided. This will ensure usage of the application across a range of class libraries, from smallest to largest. If one starts developing against the largest class library, and then learns that the target device may only provide a smaller class library, the application may require work to retarget to alternative classes or methods to accomplish the same tasks.

  Service Management Framework (SMF), the IBM implementation of the OSGi standard framework, provides the componentization and lifecycle management capabilities of bundles providing packages and services.

- **Platform management**

  Extension Services provides an OMA SyncML/DM based OSGi Agent that interacts with the Device Management Server provided in multiple IBM products, such as WebSphere Everyplace Device Manager (WEDM) and WebSphere Everyplace Access (WEA). By deploying the agent built into the platform, the agent can check for jobs to execute, such as inventory control, configuration, software distribution, and others. The agent provides the base capabilities, but it also provides service interfaces that permit application developers to interact with the agent to initiate jobs.

- **Data**

  Extension Services provides two relational databases accessible via JDBC interfaces. DB2 Everyplace is an extremely small footprint relational database. It is especially suitable for embedded devices, where large databases, and sophisticated queries are not normally required, but can also be used on larger devices. DB2 Everyplace provides transaction support covering updates to multiple tables within a single transaction, encrypted tables, and zero client administration. DB2 Cloudscape is a 100% pure Java relational database, providing SQL-92, partial SQL-99, and SQLJ support, indexes, triggers, transactions, encryption, and the standard features that one expects of a relational database. Since DB2 Cloudscape contains a larger number of features, it may not be suitable for the smaller, resource-constrained devices.

  Databases can be created by code running on the node in question, or the initial database schemas can be distributed along with applications from some other node. Both databases can be used in a read-only mode, so that databases distributed upon read-only media can be used as well. Another option for database creation, and for continual update, is to use database synchronization facilities. DB2 Everyplace and DB2 Cloudscape are both capable of synchronizing with the DB2 Everyplace Sync Server, using IBM's ISync technology provided with Extension Services. The initial synchronization activity will create the local database schema, as well as populate the initial set of data. As data is updated on the node, synchronization will transfer that data to other nodes as configured to receive it. Updates made to the data on the remote nodes will be synchronized back to the local node. Database administrators set up the necessary subscriptions for synchronization, and can set up filtering of data to limit the amount of data distributed between nodes.

  Database synchronization is based on row-level updates synchronized to and from the remote nodes. Database synchronization captures only the current state of data and synchronizes that data between nodes. If intermediate data updates or data ordering is required, application developers may need to develop their own history model, or use other technologies such as messaging to capture intermediate data.

  As part of the SMF capabilities, XML parsing services are provided. At present, two parsers are provided -- MicroXML, a small non-validating parser, and XML4J, a full-featured validating parser. Either or both of the parsers can be installed, and the application developer, when requesting a parser service, can request a specific type of parser service. See the Service Management Framework Runtime Documentation for more information on parsing capabilities.

  Extension Services also includes an implementation of the OMA SyncML/DM and /DS protocols as part of the SyncML4J bundle. This library provides the core SyncML protocol, allowing application developers to create sync adapters on both local and remote nodes to synchronize data objects between them.

- **Messaging**

  Extension Services provides support for the Java Message Service (JMS). MQ Everyplace is the messaging provider for JMS, and includes support for point-to-point messaging. MQ Everyplace is suitable for small devices, and can provide the assured, once-only delivery capabilities consistent with the WebSphere MQ family. MQ Everyplace operates in many topologies, from peer-to-peer, to client to server via the MQ Everyplace gateway technology. MQ Everyplace supports features such as synchronous and asynchronous messaging, encryption, Internet tunneling, connection aware policies to support low-bandwidth and fragile networking. JMS usage of MQ Everyplace will need to be bootstrapped using the supplied connection factory, but from then on, full JMS APIs can be used. MQ Everyplace APIs can also be used directly (without JMS).

Messaging is useful for transactional updates, or where intermediate data updates or data ordering is required. Messages containing the complete update can be sent to a server, where transaction managers can coordinate the update of multiple resources. Messaging can be paired with synchronization technology, such that transactions are sent via messages, and the resulting database updates distributed back to the client via synchronization. Messaging can also be used effectively in a disconnected environment, setting up a local queue manager to contain messages until a connection to the server infrastructure is available.

WebSphere MQ Telemetry Transport provides for publish and subscribe messaging capabilities in an extremely tiny footprint. This technology was designed for usage in the extreme embedded space of sensors, actuators, and controllers but can be used anywhere. It operates in a fully connected environment and is useful for sending data updates from a sensor to a controller, or distributing commands to various actuators.

Web Services provides an alternative request-response messaging in a fully connected environment. The web services implementation provides for both client connectivity to server hosted web services and the hosting of local web services. The implementation is based up on the Web Services for J2ME specifications, and provides support for document literal encoded streams exchanging well-typed data objects. In addition, the Web Services implementation provides a web services hosting environment. Application developers can develop OSGi services, and during registration of the service, indicate that it should also be available as a web service.

- **Interaction Services**

  SMF provides Web Container capabilities that allow application developers to create new applications, or potentially reuse existing applications, to run on a range of devices from PDAs to desktops. Application developers can make use of familiar Servlet or Java Server Pages (JSP) technologies to construct web applications. The web applications can contain all of the user interface and business logic necessary for the application, or exploit the OSGi environment to provide componentized model view controller components.

  In addition to the Web Container capabilities, the update site contains a beta implementation of the Embedded SWT technology. Embedded SWT provides a subset of the SWT technology provided by eclipse.org. Application developers can use the embedded SWT technology to construct client GUIs using familiar widgets such as buttons, text fields, lists, and more.

Chapter 3. Tasks

Chapter 4. Samples

Chapter 5. Reference

Cloudscape

DB2 Everyplace

For more information on DB2 Everyplace, refer to the *DB2 Everyplace documentation* contained in the WebSphere Studio Help.

MQ Everyplace

MQ Telemetry Transport

Web Services

For more information on Web Services, refer to the *Web Services Application Developer's Guide* contained in the WebSphere Studio Help.

Chapter 6. Javadoc

Appendix A. Additional Information

Refer to the following additional information:

- Some of the components listed in “Contents” on page 5 deliver bundles and other supporting files to the `wsdd5.0/technologies/eswe/bundlefiles` and `wsdd5.0/technologies/eswe/files` directories, respectively. During installation, Extension Services imports the bundles into the Service Management Framework (SMF) Bundle Server in WebSphere Studio Device Developer. Additionally, two Java classpath variables are defined that point to the directories containing the bundles (`ESWE_BUNDLES`) and the supporting files (`ESWE_FILES`).
- WebSphere Everyplace Micro Environment and WebSphere Everyplace Custom Environment provide an implementation of J9 for each operating system and processor combinations in the preceding table. The Extension Services allow you to add other Java Virtual Machine (JVM) implementations.

  Note: The additional implementations you add might not have been tested in the Extension Services environment.
- J9 is IBM’s virtual machine that provides the core of WebSphere Everyplace Micro Environment and WebSphere Everyplace Custom Environment. The J9 virtual machine implements a configurable and compact architectural layer that provides a common interface for application programs regardless of the underlying device hardware or operating system.
- The Extension Services tools define and make use of several custom Ant tasks. These Ant tasks begin with the prefix ″eswe″. Since these tasks are for internal Extension Services tools, their behavior may change in future releases and are not recommended or supported for end user applications.
- When creating target platforms for Sharp Zaurus and using the IPK package format, the default target root may not be used as there may not be enough space in that location to install the platform. A custom target root should be specified that has enough space to contain the target platform. On a typical Zaurus device, `/home/zaurus` is one location where enough space may be found.
- On Linux devices (including Red Hat Linux and Sharp Zaurus), non-root users typically do not have access to create servers on privileged ports (with numbers less than 1024). If you wish to run the OSGi HTTP Service on your target platform on a Linux device, you may need to run the startup script (StartSMF) as root or modify the port on which the HTTP Service will register.

Appendix B. Determining which bundles to use

Platform and bundle descriptions

The following tables provide the names, descriptions, and sizes of platform and bundle components. If you have a limited amount of space on your device, you might use the information in these tables to determine which platform and bundle components to exclude in your image.

Table 5. Platform component descriptions and sizes

Name	Filename	Supported class libraries	JXE size in KB	JAR size in KB
SMF Framework	smf.jar	Max Foundation RM	266	435
SMF Framework Messages	SMFCoreMsg_xx.jar[1]	Max Foundation RM	21	10
SMF Bundle Developer Support	smfbd.jar	Max Foundation RM	90	158
SMF Console	smfconsole.jar	Max Foundation RM	41	48
Resource Management Framework	resmanimpl.jar	RM	40	52
Resource Management Console	resmanconsole.jar	RM	14	18

Footnotes:

1. *xx* is the supported locale of the bundle (e.g. en, ja, pt_BR)

Table 6. Bundle component descriptions and sizes

Name	Filename	Supported Class Libraries	JXE size (KB)	JAR size (KB)
Cloudscape Client	db2j.jar	Max[2]	2825	2030
Cloudscape ISync Client	db2jisync.jar	Max[2]	247	186
DB2 Everyplace Client	DB2e.jar	Max[2, 3] Foundation[3]	87	45
DB2 Everyplace ISync Client	ISync.jar	Max[2, 3] Foundation[3]	47	31

Table 6. Bundle component descriptions and sizes (continued)

Name	**Filename**	**Supported Class Libraries**	**JXE size (KB)**	**JAR size (KB)**
Embedded SWT for Win32 (beta) [6]	eswt-win32.jar	Max[4] Foundation[4] RM[4]	396	252
Embedded SWT for Windows Mobile 2003(beta) [6]	eswt-wm2003.jar	Max[5] Foundation[5] RM[5]	396	252
File Administrator (sample)	fileadmin.jar	Max Foundation RM	35	38
Java Servlet 2.1 API	servlet.jar	Max Foundation RM	29	38
Java Servlet 2.3 API	servlet23.jar	Max Foundation RM	154	180
JDBC 2.0 for jclMax	database_enabler.jar	Max	47	26
JDBC 3.0 for jclFoundation (beta) [6]	jdbc.jar	Max Foundation	44	24
JMS 1.1 API	jms11.jar	Max Foundation RM	28	26
Micro XML Parser	MicroXML.jar	Max Foundation RM	104	154
MQ Everyplace	MQeBundle.jar	Max Foundation RM	522	348
MQ Everyplace JMS Support	MQeJMS.jar	Max Foundation RM	177	109
MQ Telemetry Transport	wmqtt.jar	Max Foundation RM	61	52

Table 6. Bundle component descriptions and sizes (continued)

Name	Filename	Supported Class Libraries	JXE size (KB)	JAR size (KB)
OSGi Configuration Admin Service	cm.jar	Max Foundation RM	58	79
OSGi Device Access	deviceaccess.jar	Max Foundation RM	28	140
OSGi HTTP Service	httpservice.jar	Max Foundation RM	98	40
OSGi HTTP Service for Web Container	webhttpservice.jar	Max Foundation RM	52	72
OSGi Log Service	logservice.jar	Max Foundation RM	27	33
OSGi Metatype	metatype.jar	Max Foundation RM	49	87
OSGi Preferences Service	prefs.jar	Max Foundation RM	25	35
OSGi Service Interfaces and Classes	osgi-services.jar	Max Foundation RM	30	40
OSGi Service Tracker	tracker.jar	Max Foundation RM	10	12
OSGi User Admin Service	useradmin.jar	Max Foundation RM	41	66
OSGi Utility Classes	osgi-util.jar	Max Foundation RM	15	17

Table 6. Bundle component descriptions and sizes (continued)

Name	Filename	Supported Class Libraries	JXE size (KB)	JAR size (KB)
Persistence Manager	persistence.jar	Max Foundation RM	32	42
Remote VM Admin Utility	vmadmin.jar	Max Foundation RM	43	46
SMF Administrator (sample)	smfadmin.jar	Max Foundation RM	121	157
SMF Bundle Messages	SMFBundleMsg_xx.jar[1]	Max Foundation RM	21	13
SyncML/DM OSGi Agent	osgiagent.jar	Max Foundation RM	359	389
SyncML/DM OSGi Agent Configuration	osgiConfiguration.jar	Max Foundation RM	19	27
SyncML/DM OSGi Agent Extension (sample)	SampleAgentExt.jar	Max Foundation RM	18	22
SyncML/DM OSGi Agent Servlet (sample)	osgiagentservlet.jar	Max Foundation RM	170	174
SyncML4J Framework	syncml4j.jar	Max Foundation RM	150	115
Web Application	webapplication.jar	Max Foundation RM	20	23
Web Container	webcontainer.jar	Max Foundation RM	553	841

Table 6. Bundle component descriptions and sizes (continued)

Name	Filename	Supported Class Libraries	JXE size (KB)	JAR size (KB)
Web Services	WebServices.jar	Max Foundation RM	141	124
Web Services Proxy	wsosgi.jar	Max Foundation RM	238	208
Web Services Gateway Utility (sample)	wsosgi-ui.jar	Max Foundation RM	37	34
XML Parser APIs	xmlParserAPIs.jar	Max Foundation RM	110	103
XML4J Parser	xercesImpl.jar	Max	2598	1101

Footnotes:

1. *xx* is the supported locale of the bundle (e.g. en, ja, pt_BR)
2. Supported with JDBC 2.0 for jclMax
3. Supported with JDBC 3.0
4. Supported on Windows 2000 or Windows XP
5. Supported on Windows Mobile 2003 operating system
6. Bundles available as BETA versions. These bundles are only available by downloading separate BETA features from the WSDD update site.

Some of the bundles provide equivalent packages and/or services. Depending on your environment, you might prefer to use one and not the other. The following table lists bundles that provide equivalent packages and/or services and the differences between the bundles.

Table 7. Bundles with equivalent services and their differences

Bundles	Differences
Java Servlet 2.1 API Java Servlet 2.3 API	Each bundle provides different specification levels. The Java Servlet 2.3 API bundle consumes more system resources than the Java Servlet 2.1 API bundle. These bundles cannot be used together, as they both export some of the same packages.
JDBC 2.0 JDBC 3.0 for jclFoundation	Each bundle provides different specification levels. These bundles cannot be used together, as the both export some of the same packages.

Table 7. Bundles with equivalent services and their differences (continued)

Bundles	Differences
Micro XML Parser XML4J Parser	The XML4J Parser bundle provides more services than the Micro XML Parser bundle (such as validating parsers), however, it consumes more system resources. If simple, non-validating parsers are required, use the Micro XML Parser. If both bundles are installed, properties must be specified during the service request to differentiate between the parser services that are returned.
OSGi HTTP Service OSGi HTTP Service for Web Container	These bundles provide the same service. OSGi HTTP Service for Web Container requires the Web Container bundle to be installed. If you install both of these bundles, configure them to use different ports so that they function properly. Note: In most cases, only one of these bundles is needed.
Web Services Web Services Proxy	These bundles provide instances of the Web Services runtime. The Web Services bundle provides static stubs. The Web Services Proxy bundle provides the ability for dynamic creation of stubs. These bundles cannot be used together, as the both export some of the same packages.

Bundle dependencies

The following table lists the dependencies for each bundle. Refer to this table for information to help you decide whether or not to include a bundle in your image. If a bundle is not listed in this table, the bundle has no dependencies.

Table 8. Bundle dependencies

Bundle	Depends on
Cloudscape Client	JDBC 2.0
Cloudscape Client Sync	JDBC 2.0
DB2 Everyplace Client	JDBC 2.0 or JDBC 3.0 for jclFoundation
DB2 Everyplace ISync Client	DB2 Everyplace Client JDBC 2.0 or JDBC 3.0 for jclFoundation
Embedded SWT for Win32	-
Embedded SWT for Windows Mobile 2003	-
File Administrator	Java Servlet 2.1 API or Java Servlet 2.3 API OSGi HTTP Service or OSGi HTTP Service for Web Container
Java Servlet 2.1 API	-
Java Servlet 2.3 API	-
JDBC 2.0	-
JDBC 3.0 for jclFoundation	-
JMS 1.1 API	-

Table 8. Bundle dependencies (continued)

Bundle	Depends on
Micro XML Parser	OSGi Service Interfaces and Classes XML Parser APIs
MQ Everyplace	-
MQ Everyplace JMS Support	JMS 1.1 API MQ Everyplace
MQ Telemetry Transport	-
OSGi Configuration Admin Service	OSGi Log Service OSGi Service Interfaces and Classes OSGi Service Tracker Persistence Manager SMF Bundle Messages
OSGi Device Access	OSGi Log Service OSGi Service Interfaces and Classes OSGi Service Tracker SMF Bundle Messages
OSGi HTTP Service	Java Servlet 2.1 API or Java Servlet 2.3 API OSGi Log Service OSGi Service Interfaces and Classes OSGi Service Tracker SMF Bundle Messages
OSGi HTTP Service for Web Container	Java Servlet 2.3 API OSGi Service Interfaces and Classes OSGi Service Tracker Web Container
OSGi Log Service	OSGi Service Interfaces and Classes SMF Bundle Messages
OSGi Metatype	Micro XML Parser or XML4J Parser OSGi Service Tracker SMF Bundle Messages XML Parser APIs

Table 8. Bundle dependencies (continued)

Bundle	**Depends on**
OSGi Preferences Service	OSGi Log Service OSGi Service Interfaces and Classes OSGi Service Tracker Persistence Manager SMF Bundle Messages
OSGi Service Interfaces and Classes	-
OSGi Service Tracker	-
OSGi User Admin Service	OSGi Log Service OSGi Service Tracker SMF Bundle Messages
OSGi Utility Classes	-
Persistence Manager	-
Remote VM Admin Utility	Java Servlet 2.1 API or Java Servlet 2.3 API OSGi HTTP Service or OSGi HTTP Service for Web Container OSGi Service Interfaces and Classes SMF Bundle Messages
SMF Administrator	Java Servlet 2.1 API or Java Servlet 2.3 API OSGi Configuration Admin Service OSGi HTTP Service or OSGi HTTP Service for Web Container OSGi Log Service OSGi Metatype OSGi Service Interfaces and Classes OSGi Service Tracker OSGi User Admin Service
SMF Bundle Messages	-
SyncML/DM OSGi Agent	OSGi Service Interfaces and Classes
SyncML/DM OSGi Agent Extension	OSGi Service Interfaces and Classes OSGi Service Tracker
SyncML/DM OSGi Agent Servlet	Java Servlet 2.1 API or Java Servlet 2.3 API OSGi Service Interfaces and Classes OSGi Service Tracker
SyncML4J Framework	-
Web Application	Java Servlet 2.3 API

Table 8. Bundle dependencies (continued)

Bundle	Depends on
Web Container	Java Servlet 2.3 API Micro XML Parser or XML4J Parser OSGi Service Interfaces and Classes OSGi Service Tracker SMF Bundle Messages Web Application XML Parser APIs
Web Services	-
Web Services Gateway	OSGi Service Interfaces and Classes OSGi Service Tracker
Web Services Gateway Utility	-
XML Parser APIs	-
XML4J Parser	XML Parser APIs

Detailed bundle information

The following tables provide more detailed information about the bundles. Listed below are all of the packages and services that are imported and exported by each bundle.

Table 9. Imported services by bundle

Bundles	Imported Services
Cloudscape Client	-
Cloudscape Client Sync	-
DB2 Everyplace Client	-
DB2 Everyplace ISync Client	-
Embedded SWT for Win32	-
Embedded SWT for Windows Mobile 2003	-
File Administrator	org.osgi.service.http.HttpService
Java Servlet 2.1 API	-
Java Servlet 2.3 API	-
JDBC 2.0	-
JDBC 3.0 for jclFoundation	-
JMS 1.1 API	-
Micro XML Parser	-
MQ Everyplace	-
MQ Everyplace JMS Support	-
MQ Telemetry Transport	-

Table 9. Imported services by bundle (continued)

Bundles	Imported Services
OSGi Configuration Admin Service	org.osgi.service.log.LogService com.ibm.pvc.smf.nls.LanguagePackService
OSGi Device Access	org.osgi.service.log.LogService com.ibm.pvc.smf.nls.LanguagePackService
OSGi HTTP Service	org.osgi.service.log.LogService com.ibm.pvc.smf.nls.LanguagePackService
OSGi HTTP Service for Web Container	com.ibm.osg.webcontainer.WebContainer
OSGi Log Service	com.ibm.pvc.smf.nls.LanguagePackService
OSGi Metatype	javax.xml.parsers.SAXParserFactory com.ibm.pvc.smf.nls.LanguagePackService
OSGi Preferences Service	org.osgi.service.log.LogService com.ibm.pvc.smf.nls.LanguagePackService
OSGi Service Interfaces and Classes	-
OSGi Service Tracker	-
OSGi User Admin Service	org.osgi.service.log.LogService com.ibm.pvc.smf.nls.LanguagePackService
OSGi Utility Classes	-
Persistence Manager	-
Remote VM Admin Utility	org.osgi.service.http.HttpService com.ibm.pvc.smf.nls.LanguagePackService
SMF Administrator	com.ibm.osg.service.metatype.MetaTypeService org.osgi.service.cm.ConfigurationAdmin org.osgi.service.http.HttpService org.osgi.service.log.LogReaderService org.osgi.service.log.LogService org.osgi.service.packageadmin.PackageAdmin org.osgi.service.permissionadmin.PermissionAdmin org.osgi.service.useradmin.UserAdmin org.osgi.service.startlevel.StartLevel
SMF Bundle Messages	-
SyncML/DM OSGi Agent	org.osgi.service.log.LogService
SyncML/DM OSGi Agent Extension	org.osgi.service.log.LogService
SyncML/DM OSGi Agent Servlet	org.osgi.service.http.HttpService org.osgi.service.log.LogService
SyncML4J Framework	-
Web Application	-

Table 9. Imported services by bundle (continued)

Bundles	Imported Services
Web Container	javax.xml.parsers.SAXParserFactory com.ibm.pvc.smf.nls.LanguagePackService
Web Services	-
Web Services Gateway	-
Web Services Proxy Gateway Utility	org.osgi.service.http.HttpService
XML Parser APIs	-
XML4J Parser	-

Table 10. Exported services by bundle

Bundles	Exported Services
Cloudscape Client	-
Cloudscape Client Sync	-
DB2 Everyplace Client	-
DB2 Everyplace ISync Client	-
Embedded SWT for Win32	-
Embedded SWT for Windows Mobile 2003	-
File Administrator	-
Java Servlet 2.1 API	-
Java Servlet 2.3 API	-
JDBC 2.0	-
JDBC 3.0 for jclFoundation	-
JMS 1.1 API	-
Micro XML Parser	javax.xml.parsers.SAXParserFactory javax.xml.parsers.DocumentBuilderFactory
MQ Everyplace	-
MQ Everyplace JMS Support	-
MQ Telemetry Transport	-
OSGi Configuration Admin Service	org.osgi.service.cm.ConfigurationAdmin
OSGi Device Access	-
OSGi HTTP Service	org.osgi.service.http.HttpService
OSGi HTTP Service for Web Container	org.osgi.service.http.HttpService
OSGi Log Service	org.osgi.service.log.LogReaderService org.osgi.service.log.LogService
OSGi Metatype	com.ibm.osg.service.metatype.MetaTypeService
OSGi Preferences Service	org.osgi.service.prefs.PreferencesService
OSGi Service Interfaces and Classes	-
OSGi Service Tracker	-
OSGi User Admin Service	org.osgi.service.useradmin.UserAdmin

Table 10. Exported services by bundle (continued)

Bundles	Exported Services
OSGi Utility Classes	-
Persistence Manager	-
Remote VM Admin Utility	-
SMF Administrator	-
SMF Bundle Messages	com.ibm.pvc.smf.nls.LanguagePackService
SyncML/DM OSGi Agent	-
SyncML/DM OSGi Agent Extension	-
SyncML/DM OSGi Agent Servlet	-
SyncML4J Framework	-
Web Application	-
Web Container	com.ibm.osg.webcontainer.WebContainer
Web Services	-
Web Services Proxy Gateway	-
Web Services Proxy Gateway Utility	-
XML Parser APIs	-
XML4J Parser	javax.xml.parsers.SAXParserFactory javax.xml.parsers.DocumentBuilderFactory

Table 11. Imported packages by bundle

Bundles	Imported Packages	Version
Cloudscape Client	java.sql	-
Cloudscape Client Sync	java.sql	-
DB2 Everyplace Client	java.sql org.osgi.framework	-
DB2 Everyplace ISync Client	java.sql com.ibm.db2e.jdbc org.osgi.framework	-
Embedded SWT for Win32	-	-
Embedded SWT for Windows Mobile 2003	-	-
File Administrator	javax.servlet javax.servlet.http org.osgi.framework org.osgi.service.http	2.1 2.1 1.0 1.1
Java Servlet 2.1 API	-	-
Java Servlet 2.3 API	com.ibm.pvc.msg	1.0
JDBC 2.0	-	-
JDBC 3.0 for jclFoundation	-	-

Table 11. Imported packages by bundle (continued)

Bundles	**Imported Packages**	**Version**
JMS 1.1 API	-	-
Micro XML Parser	org.osgi.framework javax.xml.parsers org.w3c.dom org.xml.sax org.xml.sax.helpers	1.1 1.1 2.0 2.0 2.0
MQ Everyplace	-	-
MQ Everyplace JMS Support	com.ibm.mqe com.ibm.mqe.administration com.ibm.mqe.event com.ibm.mqe.mqemqmessage com.ibm.mqe.registry javax.jms	-
MQ Telemetry Transport	-	-
OSGi Configuration Admin Service	org.osgi.service.cm org.osgi.service.log com.ibm.pvc.persistence com.ibm.pvc.resman org.osgi.util.tracker org.osgi.framework com.ibm.osg.service.cm.nls	1.1 1.2 1.0 1.0.0 1.2 1.2
OSGi Device Access	org.osgi.framework org.osgi.service.device org.osgi.service.log org.osgi.util.tracker com.ibm.osg.service.device.nls	1.1 1.1 1.0 1.1 -

Table 11. Imported packages by bundle (continued)

Bundles	**Imported Packages**	**Version**
OSGi HTTP Service	org.osgi.service.http	1.1
	org.osgi.service.log	1.0
	org.osgi.util.tracker	1.1
	org.osgi.framework	1.0
	com.ibm.osg.socket	1.0
	com.ibm.osg.socket.https	-
	com.ibm.pvc.resman	1.0.0
	javax.servlet	2.1
	javax.servlet.http	2.1
	org.osgi.service.cm	1.0
	com.ibm.osg.service.http.nls	-
OSGi HTTP Service for Web Container	com.ibm.osg.webapp	1.0
	com.ibm.osg.webcontainer.listeners	1.0
	javax.servlet	2.3
	javax.servlet.http	2.3
	org.osgi.framework	1.1
	org.osgi.service.http	1.1
	org.osgi.service.log	1.1
	org.osgi.util.tracker	1.1
	com.ibm.pvc.resman	1.0
OSGi Log Service	com.ibm.pvc.eventmgr	-
	com.ibm.pvc.resman	1.0.0
	org.osgi.framework	1.1
	org.osgi.service.cm	1.0
	org.osgi.service.log	1.2
	com.ibm.osg.service.log.nls	-

Table 11. Imported packages by bundle (continued)

Bundles	Imported Packages	Version
OSGi Metatype	javax.xml.parsers	1.1
	org.osgi.framework	1.1
	org.osgi.service.metatype	1.0
	org.osgi.util.tracker	1.1
	org.xml.sax	2.0
	org.xml.sax.helpers	2.0
	com.ibm.pvc.resman	1.0
	com.ibm.pvc.msg	1.0
	com.ibm.osg.service.metatypeimpl.nls	-
OSGi Preferences Service	com.ibm.pvc.persistence	1.0
	org.osgi.framework	1.1
	org.osgi.service.log	1.0
	org.osgi.service.prefs	1.0
	org.osgi.util.tracker	1.1
	com.ibm.osg.service.prefs.nls	-
OSGi Service Interfaces and Classes	org.osgi.framework	1.2
OSGi Service Tracker	org.osgi.framework	1.1
OSGi User Admin Service	com.ibm.pvc.eventmgr	-
	com.ibm.pvc.persistence	1.0
	com.ibm.pvc.resman	1.0.0
	org.osgi.framework	1.1
	org.osgi.service.log	1.0
	org.osgi.service.useradmin	1.0
	org.osgi.util.tracker	1.1
	com.ibm.osg.service.useradmin.nls	-
OSGi Utility Classes	-	-
Persistence Manager	com.ibm.pvc.reliablefile	1.0
Remote VM Admin Utility	javax.servlet	2.1
	javax.servlet.http	2.1
	org.osgi.framework	1.1
	org.osgi.service.http	1.1
	org.osgi.service.log	1.0
	com.ibm.osg.vmadmin.nls	1.0

Table 11. Imported packages by bundle (continued)

Bundles	Imported Packages	Version
SMF Administrator	com.ibm.osg.service.metatype	-
	com.ibm.pvc.msg	1.0
	javax.servlet	2.1
	javax.servlet.http	2.1
	org.osgi.framework	1.2
	org.osgi.service.cm	1.1
	org.osgi.service.http	1.1
	org.osgi.service.log	1.2
	org.osgi.service.metatype	1.0
	org.osgi.service.packageadmin	1.1
	org.osgi.service.permissionadmin	1.1
	org.osgi.service.useradmin	1.0
	org.osgi.service.startlevel	1.0
	org.osgi.service.wireadmin	1.0
	org.osgi.util.tracker	-
SMF Bundle Messages	-	-
SyncML/DM OSGi Agent	org.osgi.framework	-
	org.osgi.service.packageadmin	
	org.osgi.util.tracker	
	org.osgi.service.log	
	org.osgi.service.cm	
SyncML/DM OSGi Agent Extension	org.osgi.framework	1.2
	org.osgi.service.packageadmin	1.1
	org.osgi.util.tracker	1.2
	org.osgi.service.log	-
	com.ibm.osg.service.osgiagent	-
SyncML/DM OSGi Agent Servlet	com.ibm.osg.service.osgiagent	-
	javax.servlet	2.1
	javax.servlet.http	2.1
	org.osgi.framework	1.2
	org.osgi.service.http	1.1
	org.osgi.service.log	1.2
	org.osgi.util.tracker	1.2

Table 11. Imported packages by bundle (continued)

Bundles	Imported Packages	Version
SyncML4J Framework	-	-
Web Application	javax.servlet	2.3
	javax.servlet.http	2.3
	javax.servlet.jsp	1.2
	javax.servlet.jsp.tagext	1.2
	org.osgi.framework	1.1
Web Container	com.ibm.osg.webapp	1.0
	com.ibm.osg.webcontainer.listeners	1.0
	com.ibm.pvc.msg	1.0
	javax.servlet	2.3
	javax.servlet.http	2.3
	javax.servlet.jsp	1.2
	javax.servlet.jsp.resources	1.2
	javax.servlet.jsp.tagext	1.2
	javax.servlet.resources	2.3
	javax.xml.parsers	1.1
	org.osgi.framework	1.1
	org.osgi.service.cm	1.0
	org.osgi.service.log	1.1
	org.osgi.service.useradmin	1.0
	org.osgi.util.tracker	1.1
	org.xml.sax	2.0
	com.ibm.pvc.resman	1.0
Web Services	-	-
Web Services Proxy Gateway	org.osgi.framework org.osgi.service.log org.osgi.service.packageadmin org.osgi.util.tracker	-

Table 11. Imported packages by bundle (continued)

Bundles	Imported Packages	Version
Web Services Proxy Gateway Utility	com.ibm.osgi.util.logtracker	-
	com.ibm.pvcws.osgi.proxy	-
	com.ibm.pvcws.proxy	-
	com.ibm.pvcws.wsdlgleaner	-
	javax.servlet	2.1
	javax.servlet.http	2.1
	javax.xml.namespace	-
	org.osgi.framework	1.2
	org.osgi.service.http	1.1
	org.osgi.util.tracker	1.2
XML Parser APIs	org.osgi.framework	1.1
XML4J Parser	javax.xml.parsers	1.1
	org.w3c.dom	2.0
	org.w3c.dom.events	2.0
	org.w3c.dom.html	2.0
	org.w3c.dom.ranges	2.0
	org.w3c.dom.traversal	2.0
	org.xml.sax	2.0
	org.xml.sax.ext	1.0
	org.xml.sax.helpers	2.0
	org.osgi.framework	1.1

Table 12. Exported packages by bundle

Bundles	Exported Packages	Version
Cloudscape Client	com.ibm.db2j.authentication	-
	com.ibm.db2j.catalog	
	com.ibm.db2j.database	
	com.ibm.db2j.diag	
	com.ibm.db2j.jdbc	
	com.ibm.db2j.tools	
	com.ibm.db2j.types	
	com.ibm.db2j.util	
	com.ibm.db2j.vti	

Table 12. Exported packages by bundle (continued)

Bundles	**Exported Packages**	**Version**
Cloudscape Client Sync	com.ibm.mobileservices.isync com.ibm.mobileservices.isync.db2j com.ibm.mobileservices.isync.debug com.ibm.mobileservices.isync.event	-
DB2 Everyplace Client	com.ibm.db2e.jdbc	-
DB2 Everyplace ISync Client	com.ibm.mobileservices.isync	-
Embedded SWT for Win32	org.eclipse.swt org.eclipse.swt.events org.eclipse.swt.graphics org.eclipse.swt.layout org.eclipse.swt.program org.eclipse.swt.widgets	-
Embedded SWT for Windows Mobile 2003	org.eclipse.swt org.eclipse.swt.events org.eclipse.swt.graphics org.eclipse.swt.layout org.eclipse.swt.program org.eclipse.swt.widget	-
File Administrator	-	-
Java Servlet 2.1 API	javax.servlet javax.servlet.http org.osgi.service.http	2.1 2.1 1.1
Java Servlet 2.3 API	javax.servlet javax.servlet.http javax.servlet.resources javax.servlet.jsp javax.servlet.jsp.tagext javax.servlet.jsp.resources org.osgi.service.http	2.3 2.3 2.3 1.2 1.2 1.2 1.1
JDBC 2.0	java.sql	-
JDBC 3.0 for jclFoundation	java.sql javax.sql	-
JMS 1.1 API	javax.jms	1.1
Micro XML Parser	-	-

Table 12. Exported packages by bundle (continued)

Bundles	Exported Packages	Version
MQ Everyplace	com.ibm.mqe com.ibm.mqe.adapters com.ibm.mqe.administration com.ibm.mqe.attributes com.ibm.mqe.communications com.ibm.mqe.event com.ibm.mqe.messagestore com.ibm.mqe.mqemqmessage com.ibm.mqe.registry com.ibm.mqe.trace	-
MQ Everyplace JMS Support	com.ibm.mqe.jms com.ibm.mqe.transaction com.ibm.mqe.jms.admin	-
MQ Telemetry Transport	com.ibm.MQIsdp	-
OSGi Configuration Admin Service	-	-
OSGi Device Access	-	-
OSGi HTTP Service	com.ibm.osg.socket com.ibm.osg.socket.https	1.0 -
OSGi HTTP Service for Web Container	-	-
OSGi Log Service	-	-
OSGi Metatype	com.ibm.osg.service.metatype	-
OSGi Preferences Service	-	-
OSGi Service Interfaces and Classes	org.osgi.service.cm org.osgi.service.device org.osgi.service.log org.osgi.service.metatype org.osgi.service.prefs org.osgi.service.useradmin org.osgi.service.wireadmin	1.1 1.1 1.2 1.0 1.0 1.0 1.0
OSGi Service Tracker	org.osgi.util.tracker	1.2
OSGi User Admin Service	-	-
OSGi Utility Classes	org.osgi.util.measurement org.osgi.util.position	1.0 1.0
Persistence Manager	com.ibm.pvc.persistence	1.0

Table 12. Exported packages by bundle (continued)

Bundles	Exported Packages	Version
Remote VM Admin Utility	-	-
SMF Administrator	-	-
SMF Bundle Messages	com.ibm.osg.service.cm.nls com.ibm.osg.service.device.nls com.ibm.osg.service.http.nls com.ibm.osg.service.log.nls com.ibm.osg.service.metatypeimpl.nls com.ibm.osg.service.prefs.nls com.ibm.osg.service.useradmin.nls com.ibm.osg.vmadmin.nls com.ibm.osg.webcontainer.nls	1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
SyncML/DM OSGi Agent	com.ibm.osg.service.osgiagent	-
SyncML/DM OSGi Agent Extension	com.ibm.tivoli.agentext	-
SyncML/DM OSGi Agent Servlet	-	-
SyncML4J Framework	com.ibm.syncml4j com.ibm.syncml4j.authentication com.ibm.syncml4j.dm com.ibm.syncml4j.ds com.ibm.syncml4j.util	-
Web Application	com.ibm.osg.webapp com.ibm.osg.webapp.jsp com.ibm.osg.webcontainer.listeners	1.0 1.0 1.0
Web Container	-	-
Web Services	com.ibm.pvcws.jaxp.util javax.microedition.xml.rpc javax.xml.namespace javax.xml.rpc java.rmi	-

Table 12. Exported packages by bundle (continued)

Bundles	Exported Packages	Version
Web Services Proxy Gateway	com.ibm.osgi.util.logtracker com.ibm.pvcws.jaxp.util com.ibm.pvcws.jaxrpc.rpc com.ibm.pvcws.osgi.proxy com.ibm.pvcws.proxy com.ibm.pvcws.proxy.wsj2me com.ibm.pvcws.wsdlgleaner java.rmi javax.microedition.xml.rpc javax.xml.namespace javax.xml.rpc	-
Web Services Proxy Gateway Utility	com.ibm.wsosgi.proxy.test com.ibm.wsosgi.ui.util	-
XML Parser APIs	javax.xml.parsers org.w3c.dom org.w3c.dom.events org.w3c.dom.html org.w3c.dom.ranges org.w3c.dom.traversal org.xml.sax org.xml.sax.ext org.xml.sax.helpers org.osgi.util.xml	1.1 2.0 2.0 2.0 2.0 2.0 2.0 1.0 2.0 1.0
XML4J Parser	-	-

Appendix C. 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 illustrates 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.

No warranty

SUBJECT TO ANY STATUTORY WARRANTIES WHICH CAN NOT BE EXCLUDED, IBM MAKES NO WARRANTIES OR CONDITIONS EITHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION, THE WARRANTY OF NON-INFRINGEMENT AND THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, REGARDING THE PROGRAM OR TECHNICAL SUPPORT, IF ANY. IBM MAKES NO WARRANTY REGARDING THE CAPABILITY OF THE PROGRAM TO CORRECTLY PROCESS, PROVIDE AND/OR RECEIVE DATE DATA WITHIN AND BETWEEN THE 20TH AND 21ST CENTURIES. The exclusion also applies to any of IBM's subcontractors, suppliers, or program developers (collectively called "Suppliers").

Limitation of Liability

NEITHER IBM NOR ITS SUPPLIERS WILL BE LIABLE FOR ANY DIRECT OR INDIRECT DAMAGES, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST SAVINGS, OR ANY INCIDENTAL, SPECIAL, OR OTHER ECONOMIC CONSEQUENTIAL DAMAGES, EVEN IF IBM IS INFORMED OF THEIR POSSIBILITY. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE EXCLUSION OR LIMITATION MAY NOT APPLY TO YOU.

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:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. *2002* All rights reserved.

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.