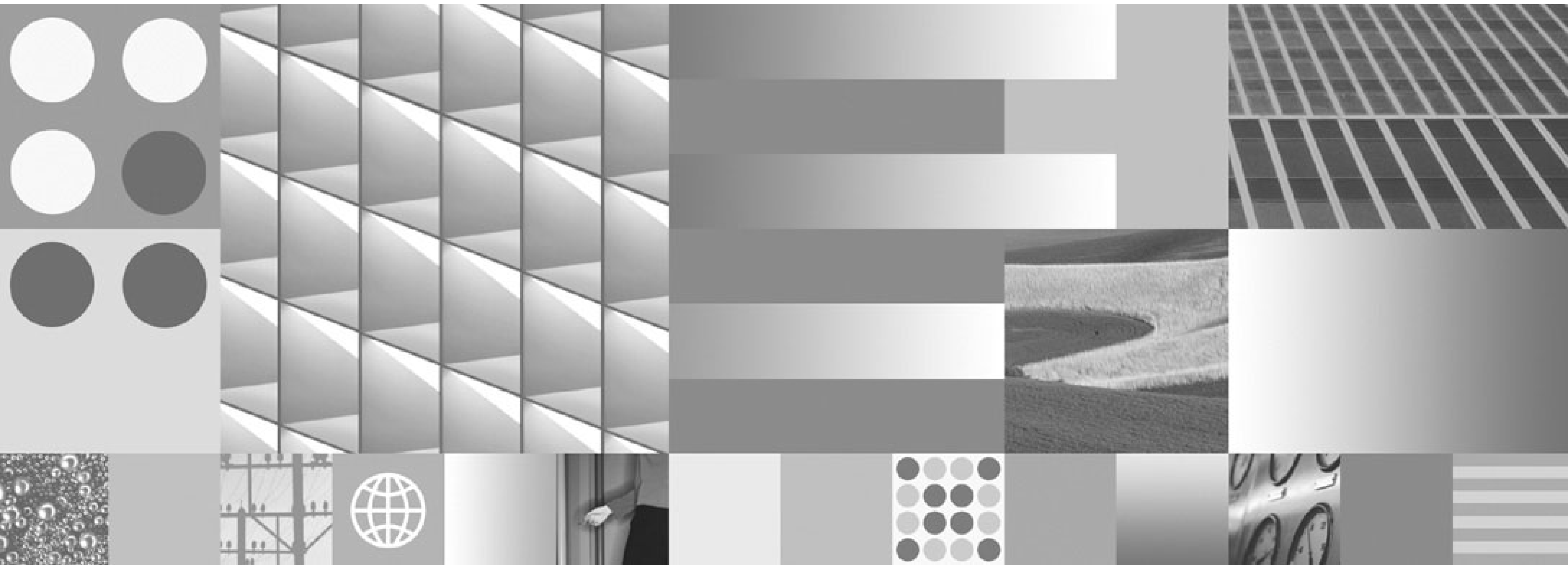
Workplace Micro Environment v5.7



J2ME Developer's Guide
For Sharp Zaurus Devices

First Edition (month 2000)

Notice here

Contents

Chapter 1. Getting Started: Linux Zaurus Targets

This section provides detailed information on:

- Preparing the target
- Setting up an installation package (IPK) for the target
- Installing and launching on a Linux Zaurus device

Currently, the only supported and fully tested models of the Linux Zaurus device are:

- SL-5600
- SL-C750/C760
- SL-6000

Note: If any of the required components are missing, these items are all available from the WebSphere® Studio Device Developer "Trials and Betas" at http://www.ibm.com/embedded.

Preparing the target

Specific files must be installed on the Linux Zaurus target, so you should know:

- How to install files
- How to uninstall files

Installing files to the target

Before an application can be run on the Linux Zaurus device, the J9 VM and class library files need to be installed as follows:

1. On your development machine, browse to the directory that contains the IPK file you need.

Profile	Directory Location
Foundation 1.0	C:\Program Files\IBM\DeviceDeveloper\wsdd5.0\ive-2.2\runtimes\zaurus\arm\foundation10\ipk
Personal Profile 1.0	C:\Program Files\IBM\DeviceDeveloper\wsdd5.0\ive-2.2\runtimes\zaurus\arm\ppro10\ipk

2. Select the IPK file. (For more information, refer to Installable IPK Types.)
3. Copy the IPK file on to Zaurus device's internal or external storage. Use Zaurus File Transfer or NFS to copy the IPK file to internal storage, or use an external storage type such as a SD Card, CF Card or a network location.
4. On the **Settings** tab, select **Add/Remove Software** to run the Zaurus installer.
5. Click **Install packages** or **Install packages via networks** according to the selected setup environment.
6. Select one of the following IPK files:

Profile	File Name
Foundation 1.0	j9-foun10-zaurus_22_arm.ipk
Personal Profile 1.0	j9-ppro10-zaurus_22_arm.ipk

7. Select the desired destination and click **OK**.

   Note: The install process may take 1 to 2 minutes. A completion dialog displays when the uninstall has completed.
8. Select **OK**, and close the installer by selecting the **X** in the upper right hand corner of the installer window.
9. For Personal Profile, confirm the installation by launching the example ″HelloWorld″ application. The application shortcut (icon) is located in the **WCTME** tab of Qtopia desktop.

Uninstalling files from the target

To remove files installed on the target follow these steps:

1. On the Settings tab, select Add/Remove Software to run the Zaurus installer.
2. Select Uninstall packages.
3. Select one of the following packages to uninstall:

Profile	**File Name**
Foundation 1.0	j9-foun10-zaurus_22_arm.ipk
Personal Profile 1.0	j9-ppro10-zaurus_22_arm.ipk

4. Select YES to proceed.

   Note: The uninstall process takes approximately 10 seconds. A completion dialog displays when the uninstall is complete.
5. Select **OK**, and close the installer by selecting the **X** in the upper right hand corner of the installer window.

Setting up an installation package (IPK)

Create an IPK file and use the Zaurus installer to install applications to the Linux Zaurus target.

For general information on creating IPK files and using the Zaurus installer, refer to Zaurus documentation provided by the Zaurus Developer Site (http://www.zaurus.com/dev/).

The following sections contain:

- Run script example
- .desktop file example
- Control file example

Run script example

It is recommended that you use the J9 VM wrapper script (**startj9foun**, **startj9ppro**) included in the package. The following is an example run script for the Linux Zaurus target.

```
#!/bin/sh
. $QPEDIR/bin/installdir.sh
- export IVEHOME=$QPEDIR/j9/ppro10
$IVEHOME/bin/startj9ppro -XappName=$0 -cp
$INSTALLDIR/helloppro/helloworld.jar
com.ibm.ive.examples.jclppro.HelloWorld
```

Note: To display the application as a task in the ″qtopia taskbar″, the value given for the ″**-XappName**″ option should match the ″**Exec**″ field value in the .**desktop** file.

For information on running Applets and Xlets, refer to the section, **Running Applications, Applets and Xlets for the Linux Zaurus target**.

For information on J9 VM command line options, refer to **Running the J9 VM from a Command Line**. If you are viewing the Linux Zaurus PDF, you will not be able to access this link. To access this information you must view the online help or the main WSDD PDF.

.desktop file example

The following is an example of a .desktop file for the Linux Zaurus target:

```
[Desktop Entry]
Comment=HelloWorld for Zaurus
Exec=j9ppro-helloworld
Icon=j9.png
Type=Application
Name=HelloWorld
Display=640x480/144dpi,480x640/144dpi
HidePrivilege=1
```

Control file example

The following is an example control file for the Linux Zaurus target.

```
Package: helloworld
Installed-Size: 10k
Filename: ./helloworld_1.0_arm.ipk
Version: 1.0
Priority: optional
Section: java
Architecture: arm
Maintainer: myname <my@address>
Depends:j9-ppro10-zaurus.ipk
Description: HelloWorld Example
 This is an example description for the
 control file.
```

Note: If you have the *Depends:j9-ppro10-zaurus.ipk* field in the control file, you can make the dependency to the WCTME for Zaurus runtime prior to installation.

Installing and Launching on a Linux Zaurus device

After the runtime and class libraries are installed on the Linux Zaurus device, and an IPK package is set up for the application being installed, install using the Zaurus installer and launch by clicking on the appropriate application icon created on the device.

For information on debugging and/or profiling refer to, **Remotely Debugging/Profiling an Application on a Linux Zaurus Device**.

Chapter 2. Concepts

This section is divided into these parts:

- Prerequisites
- Installable IPK Types
- Linux Zaurus Runtime and Class Libraries

For information on installable IPK files and required sizes refer to, **Installable IPK Types**.

Prerequisites

The Linux Zaurus runtime and class libraries are provided in the Zaurus installer package form (IPK). The amount of temporary work space required is usually double the size of the installed package. Required free space on the target depends on the type of installation package selected.

Note: The only supported and fully tested models of the Zaurus device are:

- SL-5600
- SL-C750/C760
- SL-6000

Installable IPK Types

Two types of Zaurus installer packages (IPK) are provided for installing the required runtimes and class libraries to the target The following IPK files are available:

IPK filename	Description
j9-foun10-zaurus_22_arm.ipk	**J2ME Foundation Profile 1.0** (JSR-46) runtime package. Includes J9 VM runtime and the CDC/Foundation Profile class libraries. **Dependencies**: None. This is also used for **Debugging and Profiling**. For information on debugging/profiling refer to, **Remotely Debugging/Profiling an Application on a Linux Zaurus Device**.
j9-ppro10-zaurus_22_arm.ipk	**J2ME Personal Profile 1.0** (JSR-62) runtime package. Includes J9 VM runtime and the Personal Profile class libraries which also include the CDC/Foundation Profile class libraries. An example Personal Profile ″HelloWorld″ application is included and installed with this package. **Dependencies**: None This is also used for **Debugging and Profiling**. For information on debugging/profiling refer to, **Remotely Debugging/Profiling an Application on a Linux Zaurus Device**.

The installer packages can be found in the following locations:

Profile	Directory Location
Foundation 1.0	C:\Program Files\IBM\DeviceDeveloper\wsdd5.0\ive-2.2\runtimes\zaurus\arm\foundation10\ipk
Personal Profile 1.0	C:\Program Files\IBM\DeviceDeveloper\wsdd5.0\ive-2.2\runtimes\zaurus\arm\ppro10\ipk

Note: The two IPKs are mutually exclusive. The **Personal Profile 1.0** IPK also installs **Foundation 1.0**.

Note: An example procedure for installing an .ipk package is available in the **Installing files to the target** section.

Linux Zaurus Runtime and Class Libraries

A complete J9 runtime consists of a version of the J9 VM executable plus a collection of shared library files. The J9 virtual machine is based on the JDK 1.3 specification and is designed specifically for the execution of Java applications on embedded devices.

Two J9 class libraries are available for Linux Zaurus:

- **jclFoundation** for Linux Zaurus is an implementation of the **J2ME Foundation Profile 1.0 (JSR-46)**, based on the **Connected Device Configuration (JSR-036)**.
- **jclPPro** for Linux Zaurus is an implementation of the **J2ME Personal Profile 1.0 (JSR-62)**, based on the **Foundation Profile**.

On the desktop, runtimes files are located in the following directories:

Type	Location
Executables and shared libraries	C:\Program Files\IBM\DeviceDeveloper\wsdd5.0\ive-2.2\runtimes\zaurus\arm\<JCL NAME>\bin
Foundation 1.0 Class library	C:\Program Files\IBM\DeviceDeveloper\wsdd5.0\ive-2.2\runtimes\zaurus\arm\foun10\lib
Personal Profile 1.0 Class library	C:\Program Files\IBM\DeviceDeveloper\wsdd5.0\ive-2.2\runtimes\zaurus\arm\ppro10\lib

Note: IPK files are located in the IPK directory.

On the device, runtimes files are located in the following directories:

Type	Location
Executables and shared libraries	/home/QTPalmtop/j9/<JCL NAME>/bin
Class libraries	/home/QTPalmtop/j9/<JCL NAME>/lib

Chapter 3. Tasks:Working with Linux Zaurus Targets

This section provides information on working with Linux Zaurus Targets, including instructions and details on:

- Establishing a Network Connection with the Desktop
- Deploying Applications to the Zaurus Target
- Remotely Debugging/Profiling an Application on a Linux Zaurus Device

Establishing a network connection with the desktop

Before the application can be remotely debugged/profiled and deployed on the Linux Zaurus target, a network connection must be established between the development machine and the target.

For information on a network connection for the Linux Zaurus, refer to the user manual. Also the Zaurus Developer site's ″Howto″ page (http://docs.zaurus.com/) has related information on the subject.

Deploying applications to the Zaurus target

For information on deploying applications to the Zaurus Target, refer to the following sections:

- Establishing a Network Connection with the Desktop
- Setting up an Installation Package
- Installing and Launching on a Linux Zaurus Device

In the following sections, you will find information on:

- Running Applications, Applets and Xlets
- Examples
- Qtopia Tasks

Running Applications, Applets and Xlets

This section describes how to do the following on a Linux Zaurus target:

- Run Applications
- Run Applets
- Run Xlets

You can use this information for writing run scripts to be included in IPK files.

Running Applications

To run Java Personal Profile applications, use ″**-classpath**″ option to specify the classpath to the jar or directory containing the main class and give the main class name as a parameter. For example,

$IVEHOME/bin/startj9ppro -classpath /path/to/SomeClass.jar SomeClass, where $IVEHOME is the location of the j9 VM on the Zaurus target. (e.g. /home/QtPalmtop/j9/ppro10)

Use the *"-jar"* option to run a jar file with manifest information about the Main class. For example,

```
$IVEHOME/bin/startj9ppro -jar /path/to/SomeClassWithManifest.jar.
```

Similar to Personal Profile applications, use the "**startj9foun**" script to run Foundation applications.

Running Applets (for Personal Profile runtime)

To run Java Applets, run the appletviewer (**com.ibm.oti.appletviewer.AppletViewer**) with the URL to the applet html file as its parameter. For example,

```
$IVEHOME/bin/startj9ppro com.ibm.oti.appletviewer.AppletViewer
file:/path/to/SomeApplet.html
```

Or, for HTTP retrieval:

```
$IVEHOME/bin/startj9ppro com.ibm.oti.appletviewer.AppletViewer
http://url/to/SomeApplet.html .
```

Also, the "**-appletviewer**" option can be used for simplicity:

```
$IVEHOME/bin/startj9ppro -appletviewer http://url/to/SomeApplet.html
```

Running Xlets (for Personal Profile runtime)

To run Xlets, run the **XletApplicationManager** (**com.ibm.oti.xlet.XletApplicationManager**) with the Xlet name (Class name) and the classpath. For example:

```
$IVEHOME/bin/startj9ppro com.ibm.oti.xlet.XletApplicationManager
-name:SomeXletName -path:/path/to/XletClass.jar
```

Also, the "**-xlet**" option can be used for simplicity:

```
$IVEHOME/bin/startj9ppro -xlet -name:SomeXletName
-path:/path/to/XletClass.jar
```

The **XletApplicationManager** can be run without any parameters. This will launch the **XletApplicationManager** where Xlets could be loaded interactively.

```
$IVEHOME/bin/startj9ppro -xlet
```

Type the following for more help:

```
$IVEHOME/bin/startj9ppro -xlet -h
```

Examples

Running the GraphLayout demo applet (C750 preinstalled)

```
$IVEHOME/bin/startj9ppro -appletviewer
file:/home/QtPalmtop/java/GraphLayout/graphlayout.html
```

Running an Xlet Application named *"XletTest"*

```
$IVEHOME/bin/startj9ppro -xlet -name:XletTest
-path:/home/zaurus/xletbasic.jar
```

Qtopia tasks

In order for applications to be shown as tasks on the Qtopia taskbar, a **.desktop** file must be installed for the application, and the name of the application executable (specified in the ″Exec″ field in .desktop) must be given to the **startj9ppro** wrapper script with the ″-**XappName**″ option.

A desktop entry example:

```
[Desktop Entry]
Comment=HelloWorld for Zaurus
Exec=j9ppro-helloWorld
con=j9.png
Type=Application
Name=HelloWorld
Display=640x480/144dpi,480x640/144dpi
HidePrivilege=1
```

Note: Verify that you have the icon (in this case ″*j9.png*″) installed in the `/home/QtPalmtop/pics` directory.

Note: The executable ″**j9ppro**″ in the **.desktop** file ″Exec″ field should be written as follows:

```
#!/bin/sh
. $QPEDIR/bin/installdir.sh
export IVEHOME=$QPEDIR/ive
exec IVEHOME/bin/startj9ppro -XappName=j9ppro -cp
IVEHOME/bin/startj9ppro -XappName=j9ppro -cp
$INSTALLDIR/helloppro/helloworld.jar
com.ibm.ive.examples.jclppro.HelloWorld
```

The value given for ″-XappName″, matches ″Exec″ in the .desktop file.

Full Screen mode

Personal Profile for Zaurus supports ″**Full Screen**″ mode. To use it, run with the `″-Dcom.ibm.oti.awt.FullScreenWindowBehavior=true″` option:

```
$IVEHOME/bin/startj9ppro -Dcom.ibm.oti.awt.FullScreenWindowBehavior=true
-classpath /path/to/SomeClass.jar SomeClass
```

Remotely Debugging/Profiling an Application

The J9 VM supports the JDWP debugging interface and the JVMPI based profiling interface.

This section provides instructions on:

- Remotely Debugging an Application on Linux Zaurus Device
- Remotely Profiling with the Linux Zaurus Device

Remotely Debugging an Application

The J9 VM supports the JDWP debugging interface. The J9 VM supports a large subset of the JDWP protocol but does not support a few requests which are not feasibly implemented without JNI.

To remotely debug an application on the target, follow these steps:

1. Set up a network connection between the Linux Zaurus device and the development workstation. For more information, refer to **Establishing a Network Connection with the Desktop**.
2. Install the files needed for J9 debugger/profiler using the appropriate IPK file and Zaurus Installer on to the target.
3. After installing files onto the device, deploy the Application to be debugged with the following changes made.
   - Change the application icon name (″Name″ field in the .desktop file) to something noticeable such as ″Appname_debug″
   - Change the J9 VM wrapper script to ″startj9{JCL}_debug″ (where {JCL} is either ″foun″ for Foundation Profile or ″ppro″ for Personal Profile)

   For example, a setup for a ″HelloWorld″ Personal Profile application may be one of the following:

   ```
   [Desktop Entry]
   Comment=HelloWorld for Zaurus
   Exec=j9ppro-helloworld
   Icon=j9.png
   Type=Application
   Name=HelloWorld
   Display=640x480/144dpi,480x640/144dpi
   HidePrivilege=1
   ```

   j9ppro run script (normal run) example:

   ```
   #!/bin/sh
   $QPEDIR/bin/installdir.sh
   export IVEHOME=$QPEDIR/ive
   IVEHOME/bin/startj9ppro -XappName=j9ppro -cp
   $INSTALLDIR/helloppro/Helloworld.jar
   com.ibm.ive.examples.jclppro.HelloWorld
   ```

   For debugging this ″HelloWorld″ application, the .desktop file and run script should be edited like one of the following examples:

   .desktop file (debugging) example:

   ```
   [Desktop Entry]
   Comment=HelloWorld for Zaurus
   Exec=j9ppro_debug
   Icon=j9.png
   Type=Application
   Name=HelloWorld debug
   Display=640x480/144dpi,480x640/144dpi
   HidePrivilege=1
   ```

   j9ppro_debug run script (debugging) example:

   ```
   #!/bin/sh
   . $QPEDIR/bin/installdir.sh
   export IVEHOME=$QPEDIR/ive
   exec $IVEHOME/bin/startj9ppro_debug -XappName=j9ppro_debug -cp
   $INSTALLDIR/myapps/HelloWorld.jar
   com.ibm.ive.examples.jclppro.HelloWorld
   ```
4. Create an IPK file for the application to be debugged, and install to device using the Zaurus installer. For more information, refer to **Deploying Applications to the Zaurus Target**.
5. Create the remote debug configuration for the application. This step is completed on the WSDD development machine and not on the device. In WSDD select **Run > Debug**. Select **Remote Java Application** then select **New**.

   For default setup, set:

   - the project that will be debugged on the target.
   - the **Connection Type** to **Standard (Socket Attach)** in the **Connect** tab.

- the **Connection Properties-Host** to the **IP address of the target device** in the **Connect** tab.
- the **Connection Properties-Port** to *8000*.
- the appropriate class library source from the **Source** tab (″JCL Personal Profile 1.0[j2me/2.2.0/ppro10]″ for Personal Profile.)

For customized setup of the target, refer to -**debug** option description in the **J9 VM Commands** section.

For more information on remotely debugging, refer to the **Remote debugging** section.

6. Start the application (set for debugging) on the device by clicking the application icon. In the ″Hello World″ example, this is the application with the icon labeled ″**HelloWorld_debug**″.
7. Execute the debug configuration you just created to start the debugger.

Remotely Profiling an Application

To remotely profile an application on the device, use the MicroAnalyzer. Refer to the documentation for information on how to use the MicroAnalzyer to perform profiling of an application run on device.

Note: For simplicity, a **Host-First Connection type** profiling is recommended. For more information refer to the **Establishing a Host-First Connection** section of **Establishing a Host-Target Connection**. The following procedure is written based on this connection type.

To remotely profile applications for the Linux Zaurus Target, follow these steps:

1. Set up a network connection between the Linux Zaurus device and the development workstation. For more information, refer to **Establishing a Network Connection with the Desktop**.
2. Deploy the application to be profiled. It is recommended to change the application icon name (″**Name**″ field in the **.desktop** file) to something noticeable such as ″**Appname_profile**″.

   For example, a setup for a ″HelloWorld″ Personal Profile application might look like this:

   .desktop file (normal run) example:

   ```
   [Desktop Entry]
   Comment=HelloWorld for Zaurus
   Exec=j9ppro-helloworld
   Icon=j9.png
   Type=Application
   Name=HelloWorld
   Display=640x480/144dpi,480x640/144dpi
   HidePrivilege=1
   ```

   j9ppro run script (normal run) example:

   ```
   #!/bin/sh
   $QPEDIR/bin/installdir.sh
   export IVEHOME=$QPEDIR/ive
   exec $IVEHOME/bin/startj9ppro -XappName=j9ppro -cp
   $INSTALLDIR/helloppro/Helloworld.jar
   com.ibm.ive.examples.jclppro.HelloWorld
   ```

   For profiling this ″*HelloWorld*″ application, the *.desktop* file should look like this:

   ```
   [Desktop Entry]
   Comment=HelloWorld for Zaurus
   Exec=j9ppro_profile
   Icon=j9.png
   ```

```
Type=Application
Name=HelloWorld_profile
Display=640x480/144dpi,480x640/144dpi
HidePrivilege=1
```

3. Create an IPK file for the application with a run script set for profiling. The following example shows a **Host-First Connection** profiling setup where the MicroAnaylzer host's IP address is **192.168.0.21**.

   run_j9ppro_profile run script (profiling) example

   ```
   #!/bin/sh
   . $QPEDIR/bin/installdir.sh
   export IVEHOME=$QPEDIR/ive
   exec $IVEHOME/bin/startj9ppro_profile -XappName=j9ppro_profile -
   analyze:ia=192.168.0.21,st=false -cp
   $INSTALLDIR/myapps/HelloWorld.jar
   com.ibm.ive.examples.jclppro.HelloWorld
   ```
4. Install the IPK file for the application set for profiling to the target using the Zaurus installer. For more information, refer to **Deploying Applications to the Zaurus Target.**
5. Create the **Remote MicroAnalyzer Configuration**. For more information, refer to the **Establishing a Host-First Connection** section of **Establishing a Host-Target Connection** .

   Note: In the ″Hello World″ example, this is the application with the icon labeled ″**HelloWorld_profile**″.
6. Verify that you have:
   - Checked the **Wait for target to connect** if using a **Host-First Connection** profiling.
   - Set the local port number to wait for connections (default 4821).
   - Set the Target name/address accordingly (Not needed for *Host-First Connection* profiling)
7. Execute the **Remote MicroAnalyzer Configuration** created:
   a. Launch the application (set for profiling) on the device by clicking the application icon
   b. Verify that you have launched in the appropriate order (**Host-then-Target** or **Target-then-Host**).

      Note: The order of launch depends on the connection type. For **Host-First Connections**, the order would be ″**Host-then-Target**″.

   Result: The MicroAnalyzer starts profiling.

Appendix. Additional information

Further Information

If you need more information or have questions about our product:

In the United States and Canada, call 1-800-IBM-CALL (1-800-426-2255)

In all other countries, you can submit your questions on the web at:http://www.ibm.com/software/pervasive/support/questions.shtml

You might find helpful information at the following websites or newsgroup:

- http://www.ibm.com/software/wireless/wme/
- http://www.ibm.com/software/wireless/wce/
- http://www.ibm.com/embedded
- newsgroup: ibm.software.websphere.studio.device-developer

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.