

Using Palm Transcoder with WTP

Presented by developerWorks, your source for great tutorials

ibm.com/developerWorks

Table of Contents

If you're viewing this document online, you can click any of the topics below to link directly to that section.

1. About this tutorial	2
2. Overview	4
3. Installation	5
4. Palm.Net setup	7
5. Network configuration	9
6. WTP configuration	10
7. Sample PQA page	11
8. Accessing content	13
9. PalmRedirector plugin	14
10. Profile parameters	17
11. In conclusion	19

Section 1. About this tutorial

Who should take this tutorial?

If you're using WebSphere Transcoding Publisher 3.5x (WTP) or higher, and you want to bring intranet or Internet Web pages to Palm devices via Palm.Net services, this tutorial is for you. It will show you how to install and use the Palm Transcoder plugin for bringing Web pages to your Palm devices.

About the authors

This tutorial was written by L. Victor Marks, lvmarks@us.ibm.com and Ralph Case, caser@us.ibm.com.

L. Victor Marks is a Software Engineer at IBM who develops documentation for WebSphere Transcoding Publisher. He has experience with home wireless networking, and has a degree in Music from DePauw University.

Ralph Case is a senior programmer, who develops transcoding strategy and advanced technology for IBM WebSphere products. He also works on APPN architecture and chairs the APPN Implementers Workshop (AIW). Previously, he helped develop S/390 hardware, specializing in channel subsystems, ESCON, and total systems test. He has experience building automated tools and cooperative processing systems using communications protocols, knowledge-based systems, and object-oriented programming. He holds a B.S. in Electrical Engineering from Rensselaer Polytechnic Institute.

Introduction to the tutorial

This tutorial is an introduction to the Palm Transcoder, a tool distributed as an additional feature for customers of WebSphere Transcoding Publisher 3.5x. This tutorial's goal is to help get you up and transcoding pages over Palm.Net right away. You can download the service package for the Palm Transcoder from the IBM SecureWay Software Internet Service Delivery site at <http://www6.software.ibm.com/network/isd/home.html>.

Prerequisites

To begin this tutorial, you should have an installation of a working WTP3.5x or higher, and a subscription to Palm.Net services on a PalmVII or Mobile Internet Kit-enabled Palm device with PalmOS 3.5 or higher. Alternatively, you can use a Palm emulator if you don't have a Palm device that meets these requirements. Because you'll be building Web applications for a Palm device, you'll be a Palm Developer. You might want to become familiar with the resources that Palm has made available to Palm Developers at <http://palmos.com/dev/>. Using WebSphere Transcoding Publisher to adapt your existing content will save you a lot of work,

but you'll still need to get:

- * The Palm OS Emulator (<http://www.palmos.com/dev/tech/tools/emulator/>)
- * A ROM image (the software for the emulator) that supports Web Clipping Applications. You can transfer the ROM image from your Palm device, or be a Palm Alliance Member (<http://www.palmos.com/alliance/resources/>) to download an image.
- * The Web Clipping Application Builder (<http://www.palmos.com/dev/tech/webclipping/gettingstarted.html>) tool to compile your "starter" pages

Section 2. Overview

Definition

The Palm Transcoder is a plugin for WTP that transcodes Web content to the format needed for Palm.Net devices.

Purpose of the Palm Transcoder

The Palm Transcoder for WebSphere Transcoding Publisher (WTP) helps you to enable existing enterprise Web sites for mobile use.

Web sites can be displayed on a PalmOS device without requiring the user to install a Web browser and without rewriting the Web sites to conform to Palm's HTML rules. Existing Web sites are transcoded to Palm's Web Clipping Application format (<http://www.palmos.com/dev/tech/webclipping/>).

Scenario

The Palm.Net transcoder allows a Palm user to view Web sites without installing a Web browser on the Palm device. Instead, Web sites are transcoded using a WTP plugin and Palm.Net services to the Web Clipping Application format (<http://www.palmos.com/dev/tech/webclipping/>) that can be viewed directly using the built-in support in PalmOS 3.5 and later. Although this solution can be used for general purpose Web "surfing," the expected use is to make it easier to enable existing enterprise Web sites for mobile access. A customer who chooses WTP for mobile device support can then support PalmVII devices and others connected to the Internet with the Palm Mobile Internet Kit (MIK) (<http://www.palm.com/software/mik/>) with no additional work.

Section 3. Installation

Installing the Palm Transcoder

Follow these instructions to download the files:

1. Download the **PalmTranscoder.jar.download** and **PalmVII.prop** files to the WTP installation root folder.
 2. Rename the transcoder **PalmTranscoder.jar.download** to **PalmTranscoder.jar**.
 3. Register the transcoder with the Administration Console by clicking **Transcoder**.
 4. Register the **PalmVII.prop** file as a new Preference Profile with the Administration Console by clicking **Register- >Preference Profile**.
-

The Web Clipping Application

The instructions listed on the previous panel are all that is required for the transcoder. However, to use the Palm Transcoder, a Web Clipping Application (PQA) must be loaded on the device. Because you will have to build your own Web Clipping Applications, you should follow Palm's Web Clipping Developers Guide (<http://www.palmos.com/dev/tech/webclipping/gettingstarted.html>) for building the application. Of course, the choice is all yours as to how much of your application you want to customize for the Palm, and how much you want to include in the PQA. With the Palm Transcoder, you can even choose minimal customization. All the Clipping pages will be automatically converted to the right format by the transcoder when they are needed. **wtpdemo.htm** is a minimum sample of how to build a PQA for the PalmOS.

Building your PQA

Follow these instructions for building your application:

1. Copy the [HTML source code](#) on page 11 sample below to a file called **wtpdemo.htm**.
2. Edit the URL's line in **wtpdemo.htm** to point to the WebSphere Transcoding Publisher server.
3. Using the Web Clipping Application Builder (<http://www.palmos.com/dev/tech/webclipping/gettingstarted.html>) application from Palm.com, build **wtpdemo.PQA** from **wtpdemo.htm**. If you haven't used the Builder before, you can follow Palm's Basic Tutorial at http://www.palmos.com/dev/tech/webclipping/tutorials/tutorial_basic.html .
4. Optionally, if you intend to enter a URL on the PalmOS device, as opposed to including a specific URL in the PQA file, download the **PalmRedirector.jar.download** file to the WTP installation root folder.
5. Rename **PalmRedirector.jar.download** to **PalmRedirector.jar**.
6. Register **PalmRedirector.jar** with the Administration Console by clicking

Transcoder.

7. Edit the Form Action in **wtpdemo.htm** to point to the WebSphere Transcoding Publisher server where the **PalmInternetRedirector** plugin is installed.

All of these files that you are instructed to download can be downloaded from the IBM SecureWay Software Internet Service Delivery Web site, using the service key that comes with your purchase of WTP 3.5x.

Section 4. Palm.Net setup

Requirements

Palm.Net requires that Web pages be designed to certain criteria specified in the Palm Web Clipping Developers Guide (<http://www.palmos.com/dev/tech/docs/webclippingguide.zip>), available from Palm.com. Pages must use a defined subset of HTML with a few extra tags and attributes. This transcoder allows any HTML pages to be used, and ensures that the Palm guidelines are followed.

The Palm.Net transcoder allows a Palm user to view Web sites without installing a Web browser on the Palm device. Instead, Web sites are transcoded using a WTP plugin and Palm.Net services to the Web Clipping Application (<http://www.palmos.com/dev/tech/webclipping/>) format that can be viewed directly using the built-in support in PalmOS versions 3.5 and later. Although this solution can be used for general purpose Web "surfing," the expected use is to make it easier to enable existing enterprise Web sites for mobile access. A customer who chooses WTP for mobile device support can then support PalmVII devices and other devices connected to the Internet with the Palm Mobile Internet Kit (MIK) (<http://www.palm.com/software/mik/>) with no additional work.

Palm device type

The device must be a Palm VII with PalmOS 3.5 or higher, or a supported compatible device with the Palm Mobile Internet Kit installed.

Proxy setting

If you don't have a Palm VII or other PalmOS device with the Palm Mobile Internet Kit installed, you can use a PalmOS emulator. To use an emulator with Palm.Net, the proxy setting in the Wireless Preferences page is set to point to Palm.Net. To set the proxy address, open the `prefs` application and choose `wireless` from the drop-down. See the Palm Developer Knowledge Base (<http://oasis.palm.com/dev/kb/faq/1409.cfm>) for details. The proxy is currently 206.112.114.81. Check for proxy server address changes at the Palm Web Clipping Development (<http://www.palmos.com/dev/tech/webclipping/#gettingstarted>) site.

PQA application

You will need a PQA application installed on the PalmOS device. You should develop this application to meet your specific needs. An [Installing and building the PQA page](#) on page 11 is included, with guidelines on how to customize it. At a minimum, this application needs to be configured with the address of the WTP host (either server or proxy) that does the transcoding. It is used to access the starting page for Web browsing. The starting page could be indicated in several ways, but must be coded in the PQA. Several starting pages could be included, but all must resolve to somewhere WTP is installed.

Alternatively (for testing) a Palm Emulator could be used. This requires a 3.5 ROM, and must be installed on a TCP/IP network so it can communicate with the Palm.Net service. Continue to the next section, [Configuring the network](#) on page 9 , for more details.

Section 5. Network configuration

Configuring the network

The Palm Device must be configured as described in Section 4. The proxy address is the address of the Palm.Net service, not of the WTP transcoding server.

There is no way to specify a proxy server for Palm.Net, so WTP must be configured as a reverse proxy, or WebSphere Application Server (WAS) plugin (not a proxy). Of course, applications that access WTP function using the JavaBean interface can also be used.

The communications between the Palm device and Palm.Net do not use standard HTTP, but rather a highly optimized protocol that uses UDP port 5002. If there are any firewalls between the Palm (or Palm Emulator) and Palm.Net, they must be configured to allow this protocol to go through.

In addition, the content host must be accessible from the Internet, so that Palm.Net can forward the requests there. This connection uses standard HTTP protocols.

Section 6. WTP configuration

Begin the configuration

In WTP, the PalmTranscoder.jar plugin must be registered and enabled. If used in the reverse proxy configuration, the WTP admin console can be used to set appropriate configuration. A suitable device profile must be installed. The included PalmVII.prop file will work, or it can be customized.

Section 7. Sample PQA page

Installing and building the PQA page

A PQA is an application that is built using HTML files in accordance with the rules in the Palm Web Clipping Applications document. The HTML and image files are then compiled into an application (with file extension .pqa) using the Palm PQA Builder tool. A PQA application can be installed on a Palm or Palm emulator using the Palm Install Tool provided with the Palm Desktop.

Copy the HTML sample that follows to a file called **wtpdemo.htm**. You can then build the .PQA application from the source HTML using the Palm.com PQABuilder (<http://www.palmos.com/dev/tech/webclipping/gettingstarted.html>). The source HTML for this application is displayed on the following panel.

HTML source code

Here is the source for a sample **wtpdemo.pqa** application:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>

<TITLE>Palm.Net Transcoding Demo</TITLE>
<META http-equiv=Content-Type content="text/html; charset=windows-1252">
<META name="palmcomputingplatform" content="true">
<META name="historylisttext" content="web browser &time &date">
<META name="palmlauncherversion" content="1.0">
</HEAD>
<BODY>

<!-- The following link points to a server where WebSphere Transcoding Publisher is installed.
      Make sure to change this to point to your server.
      Since the host is WTP and there is no path, the default reverse proxy web page will be
>
<A href="http://127.0.0.1/">WebSphere Transcoding Publisher</a>
<br>

<!-- The following form points to a server where WebSphere Transcoding Publisher is installed,
      and the PalmRedirector plugin is enabled to fetch the url that the user enters.
      Make sure to change this to point to your server. -->
<FORM name="redirect" action="http://127.0.0.1/PalmInternetRedirector" method="get">
  <Label for="target">Starting url:</Label>
  <INPUT type="text" id="target" name="target">
  <INPUT type="submit" value="Go!">
</FORM>
<BR>
</BODY>
</HTML>
```

Notes on wtpdemo.pqa. PQA code

Keep several things in mind regarding the code in the previous panel:

1. The `palmcomputingplatform` META tag must be present.
2. The "WTP" anchor creates a link to the WTP host. This sample uses the `BASE` tag above to define the address once for all links, but you could easily put a fully qualified URL here. WTP can serve as a Reverse Proxy or as a WebSphere Application Server Servlet plugin. In Reverse Proxy configuration, Transcoding Publisher acts as a proxy on behalf of the Web servers rather than the clients. Client browsers and wireless gateways are not configured to use the Transcoding Publisher server as a proxy. Instead, the hostname of the Transcoding Publisher is substituted for the hostname of the Web server for which Transcoding Publisher serves as a reverse proxy.
3. The form shows how to enter a URL on the Palm device. To use this technique, you must have installed and enabled the PalmRedirector WTP plugin, also available in this package. The user can enter a starting URL in the `target` input field, and tap the `Go!` button to send this URL to the WTP server. The action on this form must define the address of the WTP server, and the rest of the fields must be coordinated with the plugin sample, also provided.

Section 8. Accessing content

Accessing content through the WTP PalmTranscoder plugin

This graphic demonstrates one of two ways that you can access content through the WTP PalmTranscoder plugin.



Customize, build, and install the PQA

Next, follow these steps:

1. Customize the PQA source appropriately.
2. Build the PQA using the Palm PQA Builder.
3. Install the PQA on the device or emulator you will use to access transcoded content.

Section 9. PalmRedirector plugin

Sample PalmRedirector plugin

If you are using a form like the one on the [Installing and building the PQA page](#) on page 11 to enter a URL, you need a "Redirector" plugin like this one to receive the URL entered on the PalmOS device and redirect WTP to fetch the desired content.

Java code for the sample PalmRedirector

The following listing contains the Java code for the sample PalmRedirector:

```
/*
 * A Transcoding Publisher plugin to redirect a URL entered on a
 * form to the requested site.
 */

import java.io.IOException;
import java.util.Properties;

import com.ibm.wbi.protocol.http.*;
import com.ibm.wbi.protocol.http.beans.*;
import com.ibm.wbi.*;
import com.ibm.wbi.RequestEvent;

import com.ibm.transform.preferences.*;

public class PalmRedirector extends HttpPlugin {

    /**
     * The name of the Persistent Section that contains initialization information
     * (WBI Name, WBI Priority, WBI Condition).
     */
    private static final String SETUP_PROPERTIES = "plugins/myplugins/PalmRedirector";

    public void initialize() {

        /* Just create the MEG that does the work... */
        PalmRedirectorEditor re = new PalmRedirectorEditor();

        /* load the MEG's properties from the .prop file. */
        re.setup(SETUP_PROPERTIES);

        try {
            addMeg(re);
        } catch (PluginError pe) {
            pe.printStackTrace();
        }
    }
}

class PalmRedirectorEditor extends HttpRequestEditor {

    public void handleRequest( RequestEvent e )
        throws RequestRejectedException, IOException
    {
        HttpRequestHeader hr = ((DocumentInfo)e.getRequestInfo()).getHttpRequestHeader();

        /* Get the data entered in the "target" input field.  If there was nothing,
         * www.ibm.com/developerworks is the default. */
        Properties data = FormHelper.interpretFormData(e);
        String url = data.getProperty("target", "http://www.ibm.com/developerworks/index.html").trim();

        /* Set the URL in the HTTP request to be what we got from the form. */
        hr.setUrl(url);
    }
}
```

Check your work

- * The SETUP_PROPERTIES string must match where the .prop file will be installed.
 - * The target property name must match the name of the input field in the form in the PQA.
-

The properties file

Here is the PalmRedirector.prop file that controls when the plugin is invoked:

```
#Properties of translator sample plugin
Class=PalmRedirector
Description=Redirects to the url entered on a Palm form
DescriptiveName=Palm form url Redirector
Major=1
Minor=1
Name=Palm Redirector
Condition=(path=*/PalmInternetRedirector*)
Priority=100
```

- * The *Class* parameter must match the class of the plugin, defined in the Java code.
 - * The *Condition* parameter must match the action in the form in the PQA that is installed in the Palm device.
 - * The *Priority* parameter must be 100, so this request editor runs before the Text Engine Adjust Header Request Editor that is built in to WTP and runs at priority 99.
-

Building the plugin

To build the PalmRedirector plugin:

1. Customize the Java code, if necessary.
2. Compile the Java code, making sure that the classpath includes the required WTP libraries.
3. Customize the .prop file, if necessary.
4. Make the Plugin JAR file for the PalmRedirector, using something like:
5. Register PalmRedirector.jar with WTP using the Administration Console.

Section 10. Profile parameters

Sample device profile

Here is a sample device profile, followed by information on how the parameters affect transcoding.

```
#Preference Profile for PalmVII
#Wed Mar 14 14:18:28 EST 2001
desiredContentTypes=[text/html]
ObjectsSupported=false
imageScaleFactor=0.25
screenCapability=intermediate
propagateFirstTableRowData=false
disposeImages=false
textLinksPreferredToImages=false
supportedImages=[gif, jpg]
deviceType=PalmVII
fixedImageScale=false
NonConfigurableProperties=clientSideImageMapsSupported{bool}\ ObjectsSupported{bool}\
compressSource{bool}\ desiredContentTypes{text}\ createHTML{bool}\ screenCapability{text}\
colorSupported{bool}\ supportedImages{text}
imageSizeQualityTradeoff=favorHighQuality
screenPixels=153x144
deviceRule=User_Agent=*Elaine*
convertTablesToUnorderedLists=false
compressSource=true
createHTML=true
transcodeImages=true
ConfigurableProperties=imageScaleFactor{itext}\ fixedImageScale{bool}\
imageSizeQualityTradeoff{comboBox(compromise,favorHighQuality,favorSmallSize)}\
textLinksPreferredToImages{bool}\ disposeImages{bool}\ propagateFirstTableRowData{bool}\
convertTablesToUnorderedLists{bool}\ transcodeImages{bool}\ HTMLMaximumPageSize{itext}\
wantFragmenting{bool}
HTMLMaximumPageSize=6000
wantFragmenting=true
colorSupported=false
colorDepth=4
clientSideImageMapsSupported=false
```

Checking the code

Most of these properties are explained in the Profile Builder documentation (<http://www-4.ibm.com/software/webservers/transcoding/devdnlds.html>). The following properties can be of particular use to you while using the Palm Transcoder:

- * *deviceRule* must scan the User_Agent for "Elaine" to identify Palm.Net. Palm has documentation at <http://oasis.palm.com/dev/kb/faq/1460.cfm> that clarifies this.
- * *imageScaleFactor* controls the image scaling
- * *propagateFirstTableRowData* controls the table mutator
- * *createCHTML* must be true
- * *fixedImageScale* can be set to true
- * *transcodeImages* can be set to true
- * *supportedImages* must be gif or jpeg
- * *screenPixels* defines the size of the Palm screen
- * *convertTablesToUnorderedLists* -- if false, tables are preserved
- * *HTMLMaximumPageSize* Pages larger than this size are fragmented
- * *wantFragmenting* -- if false, pages are never fragmented
- * *colorSupported* -- At the time of this writing, there are no color PalmVILs, but when there are, this should allow color images
- * *colorDepth* Current PalmVII devices use a 2-bit gray scale

Tables cannot be split by fragmenting. If **wantFragmenting** is true, either make sure that your tables are small enough to fit in a fragment, or select **convert tables to lists within lists**. If you do neither of these, the page with your table on it will fragment only after the table is completely displayed. If you convert tables to lists within lists, the lists will fragment as expected.

Section 11. In conclusion

Transcoding is the way to go

Transcoding is a great way to bring both enterprise Web pages and Internet Web pages to wireless devices. The Palm Transcoder adds flexibility by enabling users with the Palm.Net services for their Palm devices to access the same Web pages and applications as users of other wireless devices.

Your feedback

We look forward to getting your feedback on this tutorial. Additionally, you are welcome to contact the authors if you have any questions or comments on this tutorial, at lvmarks@us.ibm.com for L. Victor Marks, and caser@us.ibm.com for Ralph Case.

Colophon

This tutorial was written entirely in XML, using the developerWorks Toot-O-Matic tutorial generator. The Toot-O-Matic tool is a short Java program that uses XSLT stylesheets to convert the XML source into a number of HTML pages, a zip file, JPEG heading graphics, and two PDF files. Our ability to generate multiple text and binary formats from a single source file illustrates the power and flexibility of XML.