



Palm OS® Resource Tools Guide

Palm OS® Developer Suite

Written by Mark Dugger, Brian Maas, and Jenny Green
Technical assistance from Kevin Cooper, Eric Cloninger, Richard Levenberg, John Dance.

Copyright © 2003–2004, PalmSource, Inc. and its affiliates. All rights reserved. This technical documentation contains confidential and proprietary information of PalmSource, Inc. ("PalmSource"), and is provided to the licensee ("you") under the terms of a Nondisclosure Agreement, Product Development Kit license, Software Development Kit license or similar agreement between you and PalmSource. You must use commercially reasonable efforts to maintain the confidentiality of this technical documentation. You may print and copy this technical documentation solely for the permitted uses specified in your agreement with PalmSource. In addition, you may make up to two (2) copies of this technical documentation for archival and backup purposes. All copies of this technical documentation remain the property of PalmSource, and you agree to return or destroy them at PalmSource's written request. Except for the foregoing or as authorized in your agreement with PalmSource, you may not copy or distribute any part of this technical documentation in any form or by any means without express written consent from PalmSource, Inc., and you may not modify this technical documentation or make any derivative work of it (such as a translation, localization, transformation or adaptation) without express written consent from PalmSource.

PalmSource, Inc. reserves the right to revise this technical documentation from time to time, and is not obligated to notify you of any revisions.

THIS TECHNICAL DOCUMENTATION IS PROVIDED ON AN "AS IS" BASIS. NEITHER PALMSOURCE NOR ITS SUPPLIERS MAKES, AND EACH OF THEM EXPRESSLY EXCLUDES AND DISCLAIMS TO THE FULL EXTENT ALLOWED BY APPLICABLE LAW, ANY REPRESENTATIONS OR WARRANTIES REGARDING THIS TECHNICAL DOCUMENTATION, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING WITHOUT LIMITATION ANY WARRANTIES IMPLIED BY ANY COURSE OF DEALING OR COURSE OF PERFORMANCE AND ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, ACCURACY, AND SATISFACTORY QUALITY. PALMSOURCE AND ITS SUPPLIERS MAKE NO REPRESENTATIONS OR WARRANTIES THAT THIS TECHNICAL DOCUMENTATION IS FREE OF ERRORS OR IS SUITABLE FOR YOUR USE. TO THE FULL EXTENT ALLOWED BY APPLICABLE LAW, PALMSOURCE, INC. ALSO EXCLUDES FOR ITSELF AND ITS SUPPLIERS ANY LIABILITY, WHETHER BASED IN CONTRACT OR TORT (INCLUDING NEGLIGENCE), FOR DIRECT, INCIDENTAL, CONSEQUENTIAL, INDIRECT, SPECIAL, EXEMPLARY OR PUNITIVE DAMAGES OF ANY KIND ARISING OUT OF OR IN ANY WAY RELATED TO THIS TECHNICAL DOCUMENTATION, INCLUDING WITHOUT LIMITATION DAMAGES FOR LOST REVENUE OR PROFITS, LOST BUSINESS, LOST GOODWILL, LOST INFORMATION OR DATA, BUSINESS INTERRUPTION, SERVICES STOPPAGE, IMPAIRMENT OF OTHER GOODS, COSTS OF PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, OR OTHER FINANCIAL LOSS, EVEN IF PALMSOURCE, INC. OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES OR IF SUCH DAMAGES COULD HAVE BEEN REASONABLY FORESEEN.

PalmSource, Palm OS, and certain other trademarks and logos are trademarks or registered trademarks of PalmSource, Inc. or its affiliates in the United States, France, Germany, Japan, the United Kingdom, and other countries. These marks may not be used in connection with any product or service that does not belong to PalmSource, Inc. (except as expressly permitted by a license with PalmSource, Inc.), in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits PalmSource, Inc., its licensor, its subsidiaries, or affiliates. All other product and brand names may be trademarks or registered trademarks of their respective owners.

IF THIS TECHNICAL DOCUMENTATION IS PROVIDED ON A COMPACT DISC, THE SOFTWARE AND OTHER DOCUMENTATION ON THE COMPACT DISC ARE SUBJECT TO THE LICENSE AGREEMENTS ACCOMPANYING THE SOFTWARE AND OTHER DOCUMENTATION.

Palm OS Resource Tools Guide
Document Number 3125-002
November 15, 2004
For the latest version of this document, visit
<http://www.palmos.com/dev/support/docs/>.

PalmSource, Inc.
1240 Crossman Avenue
Sunnyvale, CA 94089
USA
www.palmsource.com

Table of Contents

About This Book	v
How This Book Is Organized	v
Palm OS Developer Suite Documentation	vi
Additional Resources	viii
 1 Introducing Palm OS Resource Tools	 1
Defining Resource for Palm OS Applications	2
Understanding Palm OS Resource Tools	2
GenerateXRD: Resource Migration Tool	4
Palm OS Resource Editor: Resource Editing Tool	4
PalmRC and PRCMerge: Resource Building Tools	4
PRCCompare: Resource Utility Tool	4
hOverlay: Application Localization Tool.	5
PRCSign and PRCCert: Application Security Tools	5
 2 Converting Existing Resources into XML Resource Files	 7
Introducing GenerateXRD	8
When to Use GenerateXRD	8
How GenerateXRD Works.	8
Using GenerateXRD Wizard	9
Using GenerateXRD	10
Specifying Command Line Options.	10
 3 Compiling XML Resource Files	 21
Introducing PalmRC	22
Using the PalmRC Command Line Tool	23
Specifying Command Line Options.	23
 4 Building a Palm OS Application	 29
Introducing PRCMerge	30
Using the PRCMerge Command Line Tool	31
Specifying Command Line Options.	32
 5 Comparing Palm OS Resource Database Files	 35
Introducing PRCCompare	36

Using the PRCCCompare Command Line Tool	36
Specifying Command Line Options.	37
6 Localizing a Palm OS Application	41
Introducing hOverlay	42
Preparing to Use hOverlay.	42
Understanding the hOverlay Process	43
Using the hOverlay Command Line Tool	47
Specifying Command Line Options.	47
7 Securing a Palm OS Application	49
Introducing PRCCert	50
Using the PRCCert Command Line Tool	51
Specifying Key Generation Mode.	51
Specifying Certificate Generation Mode	51
Specifying PRCCert Command Line Options.	52
Introducing PRCSign	53
Using the PRCSign Command Line Tool	54
Specifying PRCSign Command Line Options	55

About This Book

This book describes a set of developer tools that you can use to create, edit, process, and compile Palm OS® resources--forms, menus, text strings, and controls--for Palm OS applications.

In this book, you learn how to use the following tools:

- `GenerateXRD` - This tool converts existing Palm OS PRC files and Macintosh RSRC files into an XML-based file format (called an XRD file). This book describes `GenerateXRD` version 1.3.
- `Palm OS Resource Editor` - With this tool, you can create and edit XML resource description (XRD) files.
- `PalmRC` - This tool compiles XML resource description (XRD) files. This book describes `PalmRC` version 1.3.
- `PRCMerge` - This tool links the output of `PalmRC` with your compiled source code to produce a Palm OS binary file (a PRC file). This book describes `PRCMerge` version 1.3.
- `PRCCompare` - This tool compares two Palm OS binary resource files and produces a report describing the differences between the files. This book describes `PRCCompare` version 1.3.
- `hOverlay` - This tool allows you to create localized versions of your Palm OS application. This book describes `hOverlay` version 1.3.
- `PRCSign` and `PRCCert` - These tools allow you to add security features to your Palm OS application. This book describes `PRCSign` and `PRCCert` version 1.0.

How This Book Is Organized

The first chapter of this book provides an overview of the resource tools.

- [Chapter 1, “Introducing Palm OS Resource Tools,”](#) on page 1 introduces you to a set of developer tools that you can use to create, edit, process, and compile Palm OS resources.

The remaining chapters each cover a tool in the resource toolset.

About This Book

Palm OS Developer Suite Documentation

- [Chapter 2, “Converting Existing Resources into XML Resource Files,”](#) on page 7 explains how you can use the GenerateXRD tool to convert your existing Macintosh RSRC files into XML-based resource description files (XRD files).
- [Chapter 3, “Compiling XML Resource Files,”](#) on page 21 tells you how to use PalmRC to compile XRD files.
- [Chapter 4, “Building a Palm OS Application,”](#) on page 29 covers information on linking your compiled XRD file with your compiled source code to produce a Palm OS application (a PRC file).
- [Chapter 5, “Comparing Palm OS Resource Database Files,”](#) on page 35 describes how to use PRCCompare to identify the differences between two Palm OS binary resource files.
- [Chapter 6, “Localizing a Palm OS Application,”](#) on page 41 discusses how to create overlay resources for Palm OS applications using the hOverlay tool.
- [Chapter 7, “Securing a Palm OS Application,”](#) on page 49 explains how to use PalmSource’s application security tools, PRCSign and PRCCert.

Palm OS Developer Suite Documentation

The following tools books are part of the Palm OS Developer Suite package:

Document	Description
<i>Introduction to Palm OS Developer Suite</i>	Provides an overview of all of the Palm OS development tools: <ul style="list-style-type: none">• Compiler Tools• Resource Tools• Testing and Debugging Tools
<i>Palm OS Protein C/C++ Compiler Tools Guide</i>	Describes the tools associated with the Palm OS Protein C/C++ Compiler.

Document	Description
<i>Palm OS Protein C/C++ Compiler Language and Library Reference</i>	Provides reference information about the C language and runtime libraries used with the Palm OS Protein C/C++ Compiler.
<i>Palm OS Debugger Guide</i>	Describes how to use Palm OS Debugger.
<i>Palm OS Resource Editor Guide</i>	Describes how to use Palm OS Resource Editor to create XRD files.
<i>Palm OS Resource Tools Guide</i>	Describes how to use the Palm OS resource tools: <ul style="list-style-type: none">• GenerateXRD - migration tool• Palm OS Resource Editor - XRD editor• PalmRC - building tool• PRCMerge - building tool• PRCCompare - comparison tool• hoverlay - localization tool• PRCSign and PRCCert - code-signing tools
<i>Palm OS Resource File Formats</i>	Describes the XML formats used for XML resource definition (XRD) files. XRD files are used to define Palm OS resources, and are the input files for the Palm OS resource tools.
<i>Palm OS Cobalt Simulator Guide</i>	Describes how to use Palm OS Cobalt Simulator.
<i>Palm OS Virtual Phone Guide</i>	Describes how to use Virtual Phone.

Additional Resources

- Documentation

PalmSource publishes its latest versions of this and other documents for Palm OS developers at

<http://www.palmos.com/dev/support/docs/>

- Training

PalmSource and its partners host training classes for Palm OS developers. For topics and schedules, check

<http://www.palmos.com/dev/training>

- Knowledge Base

The Knowledge Base is a fast, web-based database of technical information. Search for frequently asked questions (FAQs), sample code, white papers, and the development documentation at

<http://www.palmos.com/dev/support/kb/>

Introducing Palm OS Resource Tools

This chapter describes the resource tools that you can use to build user interfaces for Palm OS® applications.

- [“Defining Resource for Palm OS Applications”](#) on page 2 provides an overview on how you use XML to create user interface resources for your Palm OS application.
- [“Understanding Palm OS Resource Tools”](#) provides an overview of each of the Palm OS resource tools:
 - [“GenerateXRD: Resource Migration Tool”](#) on page 4
 - [“Palm OS Resource Editor: Resource Editing Tool”](#) on page 4
 - [“PalmRC and PRCMerge: Resource Building Tools”](#) on page 4
 - [“PRCCompare: Resource Utility Tool”](#) on page 4
 - [“hOverlay: Application Localization Tool”](#) on page 5
 - [“PRCSign and PRCCert: Application Security Tools”](#) on page 5

Defining Resource for Palm OS Applications

The source code for application resources is stored in a platform independent text file format. This file format is called an XML resource description file, or *XRD file*. This XRD file is an XML implementation for defining application resources.

The Palm OS resource tools operate on this XRD file format rather than the Macintosh resource binary format (RSRC) that was used in earlier Palm OS versions.

To learn more about the XRD file format, see the book *Palm OS Resource File Formats*.

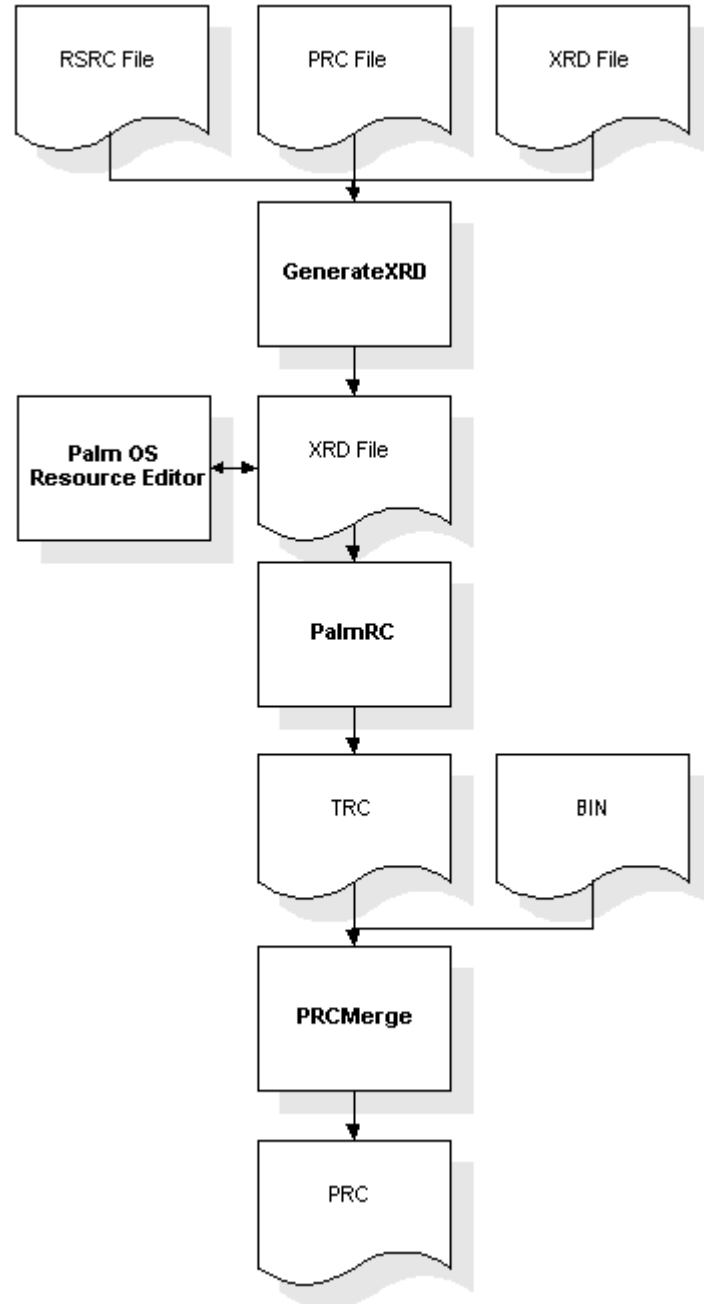
Understanding Palm OS Resource Tools

Palm OS resource tools fall into the following categories:

- Migration tools
- Editing tools
- Building tools
- Utility tools
- Localization tools
- Security tools

PalmSource's toolset provides resource tools in each of these categories, as described in the following overview.

Figure 1.1 Basic Resource Tools



GenerateXRD: Resource Migration Tool

Migration tools are a one-way conversion from existing resource files to the new resource file format. If you currently have resources in Macintosh RSRC files, you can convert them to the new XRD format without loss of information. You can also “de-compile” resources from PRC files into the XRD resource file format. The tool you use to migrate these existing formats to the XRD format is called GenerateXRD.

To learn more about GenerateXRD, see [Chapter 2, “Converting Existing Resources into XML Resource Files,”](#) on page 7.

Palm OS Resource Editor: Resource Editing Tool

Editing tools allow you to view, edit, and change Palm OS resource files (XRD files). The tool you can use to edit XRD files is called Palm OS Resource Editor. With Palm OS Resource Editor, you can edit XML tags directly, or you can use a graphical form editor, dragging and dropping user interface controls from a catalog of user interface elements.

To learn more about Palm OS Resource Editor, see *Palm OS Resource Editor Guide*.

PalmRC and PRCMerge: Resource Building Tools

Build tools are resource file compilers and linkers that build resources into Palm OS applications (PRC files). The resource file compiler is called PalmRC; the linker is called PRCMerge.

To learn more about PalmRC, see [Chapter 3, “Compiling XML Resource Files,”](#) on page 21.

To learn more about PRCMerge, see [Chapter 4, “Building a Palm OS Application,”](#) on page 29.

PRCCompare: Resource Utility Tool

Utility tools provide additional functions that can help you develop and debug Palm OS applications. PRCCompare is a utility tool that

shows you the difference between two Palm OS binary resource database files.

To learn more about PRCCompare, see [Chapter 5, “Comparing Palm OS Resource Database Files,”](#) on page 35.

hOverlay: Application Localization Tool

Localization tools allow you to create locale-specific versions of your Palm OS application. hOverlay is a tool that helps you create overlay PRCs with national language strings that work with your original base PRC.

To learn more about hOverlay, see [Chapter 6, “Localizing a Palm OS Application,”](#) on page 41.

PRCSign and PRCCert: Application Security Tools

Security tools provide a way for securing and authenticating a Palm OS application. You use PRCSign and PRCCert to create and embed a digital signature and associated certificates in your Palm OS application (PRC file).

To learn more about PRCSign and PRCCert, see [Chapter 7, “Securing a Palm OS Application,”](#) on page 49.

Introducing Palm OS Resource Tools

Understanding Palm OS Resource Tools

Converting Existing Resources into XML Resource Files

This chapter describes how to use `GenerateXRD` to convert 68K Palm OS® PRC files and Macintosh RSRC files into the XML format that can be compiled by `PalmRC`, the Palm OS XML resource compiler.

- “[Introducing GenerateXRD](#)” on page 8 describes the basic functions of the `GenerateXRD` tool.
- “[Using GenerateXRD](#)” on page 10 documents how to use `GenerateXRD` to produce XRD files.

Introducing GenerateXRD

GenerateXRD is a migration tool for resource files.

Prior to Palm OS Garnet, user interface resources were saved in Macintosh-format binary resource files (RSRC files). The Palm OS resource tools now support an XML-based file format called XRD files. To learn more about the XRD file format, see the book *Exploring Palm OS: Resource File Formats*.

GenerateXRD provides a one-way conversion from 68K resource files to the XRD file format. You can use XRD files in applications targeted for any Palm OS release.

When to Use GenerateXRD

Use GenerateXRD if:

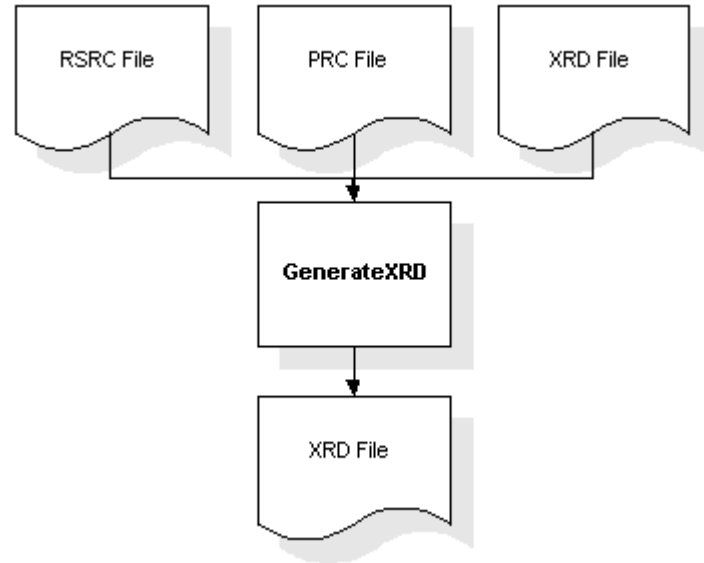
- you have user interface resources in RSRC file format (that is, if you have resource files created by Constructor for Palm OS) or in PRC files created with other resource tools,
AND
- you want to use those user interface resources to build applications with the resource tools described in this book.

How GenerateXRD Works

GenerateXRD produces a new XRD resource file by reading existing resource files:

- Macintosh binary resource files (RSRC files)
- Palm OS binary resource files (PRC files)
- Existing XRD files

Figure 2.1 GenerateXRD Overview



Resources may be selectively filtered from the source files depending on the GenerateXRD parameters specified. For example, you could specify to only extract menu and form resources, but not bitmap or code resources.

In addition, the data for some resources (such as bitmaps) may be optionally extracted into secondary data files (such as BMP files) that are then referenced from the resource descriptions in the XRD file.

GenerateXRD is integrated into Palm OS Developer Suite. When you use Developer Suite's function **File > Import** and select the Import Palm OS Resource File wizard, the wizard calls GenerateXRD to convert your existing Palm OS resource file into an XRD file.

GenerateXRD is available as a Windows command line tool. A subset of GenerateXRD's features are also available in the tool GenerateXRD Wizard.

Using GenerateXRD Wizard

GenerateXRD Wizard provides a simple interface to the GenerateXRD tool. GenerateXRD Wizard provides a subset of the

Converting Existing Resources into XML Resource Files

Using GenerateXRD

most commonly-used features of the GenerateXRD command line tool in a simple, step-by-step, Wizard interface.

Using GenerateXRD

GenerateXRD is a command line tool. The command syntax for GenerateXRD is:

```
GenerateXRD input_files -o output_file
[options]
```

input_files

One or more input resource files.

Palm OS PRC file format

The file name extensions recognized are PRC, OPRC, BPRC, and TRC.

Macintosh resource file containing PalmRez / Constructor for Palm OS resources

The file name extensions recognized are RSRC, RSR, and PLR.

XRD file format

The only file name extension recognized is XRD.

output_file

The -o option is required to specify where the XRD output file is written. You should use the file extension XRD for your filename. The file specified is overwritten automatically if it already exists.

options

Additional command line options, as described in the following section "[Specifying Command Line Options](#)."

Specifying Command Line Options

-@paramFile GenerateXRD reads parameters from a specified file.

-bitmaps *bitmapFmt*

When GenerateXRD generates bitmaps as external files, the default bitmap format is BMP (Microsoft Windows Bit Map

file format). BMP format is a commonly-used, cross-platform format.

`-bitmapFilePrefix prefixValue`

When GenerateXRD generates bitmaps as external files, you can specify a name prefix for the external bitmap filenames. For example, if then input file is `System.prc`, you might want to specify `"System_"` as the prefix so that the generated file names would be `System_Bitmap_1000-1.bmp` rather than just `Bitmap_1000-1.bmp`.

This option can help you organize files, especially if there are multiple XRD files in the same directory and you want to make the association of bitmap files with parent XRD file explicit in the filename.

For more information on how bitmap filenames are generated, see the `-externDataFiles` option.

`-d`

The attributes of the input PRC database header is stored as a `DATABASE_HEADER` element in the output XRD file. For this option, the input file specified must be exactly one PRC format file; otherwise, the results are undefined.

The `DATABASE_HEADER` element is primarily used for testing and debugging; it is not part of normal application development. Normally, when the resource tools build a PRC file, the database header of the output PRC is initialized with values specified to `PRCMerge`.

`-encoding utfvalue`

Controls the character encoding of the output XRD file. *utfvalue* must be one of the following:

`"UTF8"`

Specifies the Unicode UTF-8 encoding. This is the default.

`"UTF16"`

Specifies the Unicode UTF-16 encoding.

UTF-8 is the most compatible encoding with legacy text editors and text processing such as source control and comparison tools.

Almost all Unicode aware text editing and processing tools accept either UTF-8 or UTF-16 interchangeably.

Converting Existing Resources into XML Resource Files

Using GenerateXRD

`-externDataFiles`

When GenerateXRD creates an XRD file, all data is included in the XRD file by default. Some data, such as bitmaps and MIDI sounds, may be more easily edited and managed if the resources are stored as individual files with a standard format (such as BMP for a bitmap). The

`-externDataFiles` option tells GenerateXRD to place resource data in external data files when possible, which is then referenced by the XRD file.

Note: The file names for external data files are automatically generated based on the resource type, resource ID, and resource type specific information.

For example, for bitmap resources, the bitmap file names are generated as:

`Bitmap_resID-bitDepth.BMP`

where *resID* and *bitDepth* are filled in as decimal integers using the resource ID and logical bitmap element depth.

After you have created an XRD file using GenerateXRD, you can rename the output data files as long as you update the file reference in the XRD file to match.

`-externFilesDir directory`

When generating resource data as external data files, GenerateXRD creates the files in the same directory as the output XRD file by default. If there are a large number of external data files, or if there are multiple XRD files in the same directory, you might want to place the external data files in another directory. Use the option `-externFilesDir` to specify a particular directory where external files is created.

Generally, you should specify a subdirectory of the directory where the XRD is output, in which case the XRD file uses a relative file path for the external data files.

Note: The directory you specify for the `-externFilesDir` option must exist before calling GenerateXRD. GenerateXRD does not create the directory for you.

`-help`

GenerateXRD displays help information and ignores any other options.

Converting Existing Resources into XML Resource Files

Using GenerateXRD

`-helpTargets`

GenerateXRD displays information about supported targets and ignores any other options.

`-helpVersion`

GenerateXRD displays the tool's version information and ignores any other option.

`-inlineData`

For all resources in the input XRD files that specify data using external data files, GenerateXRD integrates the data into the output XRD file.

You can use this option to reverse the `-externDataFiles` option, if you decide you don't want external data files.

`-knownRawResType resType`

When GenerateXRD converts an unknown resource type, it outputs a message indicating that the unknown resource is being converted into a `RAW_RESOURCE` type. The binary data of the resource is preserved in the XRD file

If you have created a resource type that you want treated as a `RAW_RESOURCE` type but don't want the GenerateXRD message, then use the `-knownRawResType` option to suppress these messages.

The variable *resType* specifies a four-letter resource type code of the input resource type. You can specify the `-knownRawResType` option multiple times, one for each resource type.

`-locale "xxYY"`

Tags output resources with a `LOCALE` attribute. The `LOCALE` attribute is used by `PalmRC` to determine whether resources go into the base PRC (that is, the locale-independent resources) or into the overlay PRC (that is, the locale-specific resources).

The locale *xxYY* is either an empty string (""), to indicate forcing base locale, or a specific locale code. GenerateXRD recognizes the following specific locale codes:

`enUS`

English

`frFR`

French

Converting Existing Resources into XML Resource Files

Using GenerateXRD

itIT	Italian
deDE	German
esES	Spanish
jaJP	Japanese
zhCN	Simplified Chinese
zhTW	Traditional Chinese
koKR	Korean

If you are using GenerateXRD to input resources that may be used to build overlays, or if you plan to store multiple localized versions of a resource in a single XRD file, you should specify the `-locale` option with the appropriate locale code.

GenerateXRD requires that the locale value specified be in the form `xxYY`, where `xx` is a lowercase two-character alphabetic language code and `YY` is an uppercase two-character alphabetic country code.

When you specify a non-standard locale value (a locale value that is not defined by Palm OS), then you must specify the `-noLocaleCheck` option. See the description of `-noLocaleCheck` below for more information.

NOTE: When the input files are Macintosh resources, the resource name overlay flags (such as `strip`, `nostrip`, `ovlyAdd`) are used in conjunction with the locale and resource type to determine whether the output resource is marked with an empty (base) locale or with the localized locale and overlay attributes.

`-mapInvalidChars`

When you use this option, GenerateXRD maps any invalid character it finds to ' ? '.

`-noLocaleCheck`

When the `-locale` option is used, if the specified locale code is not one of the currently recognized codes, GenerateXRD issues a warning message. This warning guards against accidentally specifying an incorrectly locale code (for example, by mistyping a locale code). When you need to specify a non-standard locale code, you should also specify the `-noLocaleCheck` option to suppress the warning message.

`-noRsrcNameInfo`

Specifies that GenerateXRD should not input Macintosh resource names as resource comment fields when generating the XRD file. This is primarily useful when generating XRD files from Macintosh resource files and the corresponding PRC files for comparison.

When generating the XRD file from the PRC file, the comments are not available. In order to generate matching XRD from the Macintosh resources, you need to suppress the comment generation using this option so that the XRD files could be compared without the comments showing as differences.

`-noSort`

All resources are placed in the XRD file in the same order as they were read from the input files. If you do not specify this option, all resources are sorted according to XRD resource type, ID, and locale. Normally, resources are kept sorted in canonical order. This canonical order makes it easier to locate resources in the XRD file when you edit the file, and makes it easier to use source control and comparison tools to manage changes to the XRD file.

`-noUnknown`

Any resource types that are not recognized are not included in the output XRD file. A warning message is displayed for each unknown resource that is skipped.

This option should rarely be needed. In most cases, if an application is being converted from PRC or RSRC format to XRD, it is necessary to preserve any custom resources used

Converting Existing Resources into XML Resource Files

Using GenerateXRD

by that application as RAW_RESOURCE elements in the XRD file.

`-omit type`

Indicates that you want to omit a particular resource or resources of a particular type. This option is followed by a resource type expression:

type

A resource type to indicate all resources of that type.

type (id)

A resource type followed by a resource ID to indicate a specific resource.

type (id:id)

A resource type followed by an ID range to indicate all matching resources.

The `-omit` option may be specified more than once; if you use multiple `-omit` options, the options are combined. For example, "`-omit Talt -omit tFRM`" indicates that the output XRD contains all resources except for alert resources and form resources.

`-only type`

Indicates that you want to input only a particular resource or resources of a particular type. This option is followed by a resource type expression:

type

A resource type to indicate all resources of that type.

type (id)

A resource type followed by a resource ID to indicate a specific resource.

type (id:id)

A resource type followed by an ID range to indicate all matching resources.

The `-only` option may be specified more than once; if you use multiple `-only` options, the options are combined. For example, "`-only Talt -only tFRM`" indicates that the output XRD contains both alert and form resources.

Converting Existing Resources into XML Resource Files

Using GenerateXRD

`-q /-quiet`

GenerateXRD suppresses the tool description (tool name and version number) and copyright information.

`-raw`

All resources that included in the output XRD file are created as RAW_RESOURCE elements that preserve the binary input data exactly. Normally, you want the logical XML element that corresponds to the input resource to be created. However, for debugging purposes or for compatibility purposes, you may want to output the raw data.

This option is intended for PRC input files. The output behavior is undefined for Macintosh resource or XRD input files.

`-remapChars`

GenerateXRD maps single-byte instances of special characters (the ellipsis character, ®, ©, and the bullet character) to their equivalent Unicode sequences, when they are found in a double-byte character stream.

`-resInfo rsrcFile`

rsrcFile specifies a Macintosh resource file containing PalmRez or Constructor for Palm OS resources.

GenerateXRD collects Macintosh resource names, but does not extract resource data from the input file.

You can use this option to specify the Macintosh resource file that GenerateXRD should scan in order to find comment information while decompiling a PRC file.

PRC files do not retain comment information; so when GenerateXRD decompiles a PRC file, it does not find any comment information to propagate into the output XRD file. However, if you have a comparable RSRC file, you can specify it so that GenerateXRD can extract comment information from the RSRC file.

For example, say you have an overlay PRC file (OPRC) that was built using Constructor for Palm OS and the 68K-based localization tool PRC2OVL. When you convert this OPRC file into an XRD file, you may want to retain the resource symbolic names as comments, which can be extracted from the original RSRC file.

Converting Existing Resources into XML Resource Files

Using *GenerateXRD*

-s

Any "segment" resources from the input files are output to the XRD file, usually as `RAW_RESOURCE` elements. Segment resources are executable code resources or support resources that are typically generated by the compiler and/or linker. PRC resource types that are considered segment resources are 'code', 'data', 'libr', and 'ovly'.

Generating segment resources in the XRD is primarily used for testing and debugging; it is not a part of normal application development. Normally, when you build a Palm OS executable file (a PRC file), the C/C++ compiler and linker tools generate the code or related resources, not *GenerateXRD*.

-target *palmOS*

Specifies the text encoding of the input resources. *palmOS* must be one of the following targets:

4.0

Indicates Palm Latin R2 text encoding. This is the default.

4.0J

Indicates Palm ShiftJIS (Japanese) text encoding. When working with Japanese resource files, it is important to specify the "-target 4.0J" option. Otherwise, the Japanese text is corrupted in the output file.

4.0CT

Indicates Traditional Chinese text encoding.

4.0KR

Indicates Korean text encoding.

For locales that are not explicitly supported, you can specify a Microsoft Windows code page.

If you do not specify a target setting that matches the input files text encoding, the text in the output XRD is garbage.

When the input file is a Macintosh resource file, the target setting is also used to specify the fonts that are used to calculate the width of menus and certain controls.

Converting Existing Resources into XML Resource Files

Using GenerateXRD

`-types customTypesFile`

GenerateXRD uses custom resource types that are defined in the XRT file specified by *customTypesFile*.

If you have a PRC file that includes custom resource types, the only way to decompile those custom resources is to specify the custom resource types description file (XRT file) that originally defined those custom resources.

To specify more than one XRT file, use multiple `-type` options (one for each XRT file).

`-xrdMatchLocale "xxYY"`

Allows filtering resources based on the locale attribute. Only resources with locale attributes that match the specified locale *xxYY* are included in the output XRD file. All other resource types are filtered silently.

xxYY specifies a four-character alphabetic locale value or an empty string (" ") to match the empty (base) locale.

You can use the `-xrdMatchLocale` option to split resource files that have combined base and localized resources in one XRD file into separate output XRD files for base and localized resource:

1. Run GenerateXRD once to produce the base XRD file:

```
GenerateXRD input_combined.xrd  
-xrdMatchLocale "" -o output_base.xrd
```

2. Run GenerateXRD for each locale-specific XRD file:

```
GenerateXRD input_combined.xrd  
-xrdMatchLocale "enUS" -o output_enUS.xrd
```

Converting Existing Resources into XML Resource Files

Using GenerateXRD

Compiling XML Resource Files

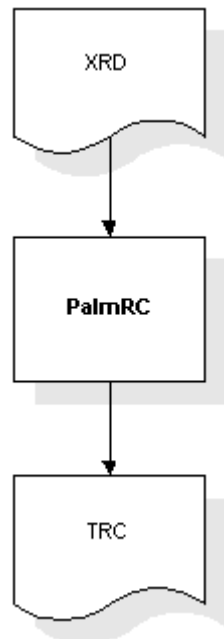
This chapter describes how to use `PalmRC` to compile XML resource files (XRD files).

- “[Introducing PalmRC](#)” on page 22 describes the basic functions of the `PalmRC` tool.
- “[Using the PalmRC Command Line Tool](#)” on page 23 describes how to specify the `PalmRC` command line arguments to compile an XRD file.

Introducing PalmRC

PalmRC compiles XRD resource files into Palm OS® resource binary format. The resource compilation typically does not include the code resources necessary to produce the final Palm OS application resource file.

Figure 3.1 PalmRC Overview



PalmRC is usually an intermediate step in the build process. PalmRC generates an output file with a TRC extension (for “temporary resource file”) rather than a PRC file. The only difference between a PRC file and a TRC file is that a TRC file does not contain the code resources and database header that are required in a PRC file.

NOTE: The TRC file is used in the build process to convey the compiled resource data. The TRC file is the output from PalmRC and the input to PRCMerge.

PalmRC is integrated into Palm OS Developer Suite. When you build a project using Developer Suite, PalmRC is called as part of the process of building the project.

PalmRC is available as a command line tool for Windows.

Using the PalmRC Command Line Tool

PalmRC is used to compile an XRD file (or files) into PRC resources. The command syntax for PalmRC is:

```
PalmRC inputFiles -o outputFile [options]
```

input_files

One or more XRD files.

output_file

The -o option is required to specify where the PalmRC output file is written. The output file is in PRC resource database format. Normally you should specify a file with a file name extension of TRC, but in some cases for testing, you might specify a file name extension of PRC, BPRC, or OPRC. The output file is overwritten automatically if it already exists.

options

Additional command line options as described in the following section "[Specifying Command Line Options](#)."

Specifying Command Line Options

-append

If the -append option is specified and the output file PRC file already exists, then PalmRC merges the compiled resources into the existing output file. If the output file does not exist, specifying this option has no effect, and the output file is created as usual.

-densityTarget *target*

Specifies target density configuration for bitmaps and fonts. The supported values of *target* are 1X, 1.5X, and 2X.

-help

PalmRC displays help information and ignores any other options.

Compiling XML Resource Files

Using the PalmRC Command Line Tool

`-helpTargets`

PalmRC displays supported targets.

`-helpVersion`

PalmRC displays version information and ignores any other options.

`-i dirpath`

Specifies a directory to be searched when header files are using during compilation.

Note: PalmRC does not currently support symbolic identifiers, so this option is ignored.

`-locale xxYY`

Filters resources using the `LOCALE` attribute. Resources with locale attributes that do not match the specified locale are omitted from the output file.

Locale-independent resources (resources with an empty locale attribute) are included or omitted depending on the value specified for the `-overlayFilter` option. By default, locale-independent resources are included in the output file.

PalmRC requires that the locale value specified be in the form `xxYY`, where `xx` is a lowercase two-character alphabetic language code and `YY` is an uppercase two-character alphabetic country code.

When you specify a non-standard locale value (a locale value that is not defined by Palm OS), then you must specify the `-noLocaleCheck` option. See the description of `-noLocaleCheck` below for more information.

The locale `xxYY` is either an empty string (`" "`), to indicate forcing base locale, or a specific locale code. PalmRC recognizes the following specific locale codes:

`enUS`

English

`frFR`

French

`itIT`

Italian

`deDE`

German

esES
Spanish
jaJP
Japanese
zhCN
Simplified Chinese
zhTW
Traditional Chinese
koKR
Korean

-makeDeps

Specifies that PalmRC should generate a list of all external data files (such as bitmap files) that the compilation depends on. The list of files is written to the specified output file in platform native text format, one file per line, with each file specified as a full path name. If the specified dependency already exists and the dependencies are unchanged, the file is not changed; otherwise it is overwritten.

-noLocaleCheck

When the `-locale` option is used, if the specified locale code is not one of the currently recognized codes, PalmRC issues a warning message. The purpose of this warning is only to guard against accidentally specifying an incorrectly locale code. When you need to specify a non-standard locale code, you should also specify this option to suppress any warning messages.

-noWarnSize

By default, PalmRC displays a warning if the size of the resource it creates is greater than 64K. But if you specify `-noWarnSize`, PalmRC does not display the warning.

-overlayFilter *filter*

The `-overlayFilter` option is used in conjunction with the `-locale` option to filter resources for overlay stripping. There are three options that may be specified for *filter*:

BASE

Only locale-independent resources (resources with an empty locale attribute) are placed in the output file.

Compiling XML Resource Files

Using the PalmRC Command Line Tool

OVERLAY

Only resources matching the specified locale are output; locale-independent resources are not output.

FULL

Both locale-independent resources and resources matching the specified locale are output.

-p *proc*

Specifies the target resource format. *proc* is one of the following values:

68K

To build resources for 68K applications (in the format used by Palm OS 4.1 and prior), specify the "-p 68K" option.

This is the option you should use for all 68K applications, even for 68K applications that are targeted for Palm OS releases after Palm OS 4.1.

PalmRC also provides the option "-p PalmOS4" as a synonym for the "-p 68K" option.

ARM

To build resources in the native format used by Palm OS Cobalt and later, specify the "-p ARM" option.

PalmRC also provides the option "-p PalmOS6" as a synonym for the "-p ARM" option.

-q /-quiet

PalmRC suppresses the tool description (tool name and version number) and copyright information.

-strictLocale

PalmRC issues an error messages if a resource locale setting in the XRD file is different from the target locale setting.

-strictTextEncoding

PalmRC issues an error messages if conversion to the output text encoding fails.

By default, PalmRC issues a warning message when an input character cannot be represented in the target text encoding.

With this option, PalmRC issues an error message and then stops processing.

`-target palmOS`

Specifies the text encoding of the output resources. *palmOS* must be one of the following targets:

`4.0`

Indicates Palm Latin R2 text encoding. This is the default.

`4.0J`

Indicates Palm ShiftJIS (Japanese) text encoding. When working with Japanese resource files, it is important to specify the "`-target 4.0J`" option. Otherwise, the Japanese text is corrupted.

`4.0CS`

Indicates PalmGB text encoding. This is the Simplified Chinese encoding format.

`4.0CT`

Indicates Traditional Chinese encoding.

`4.0KR`

Indicates Korean encoding.

For locales that are not explicitly supported, you can specify a Microsoft Windows code page.

If you do not specify the target that matches the input files text encoding, the text in the output XRD is garbage.

For Macintosh resource files, the target is also used to determine the fonts that are used to calculate the width of menus and certain controls.

Compiling XML Resource Files

Using the PalmRC Command Line Tool

Building a Palm OS Application

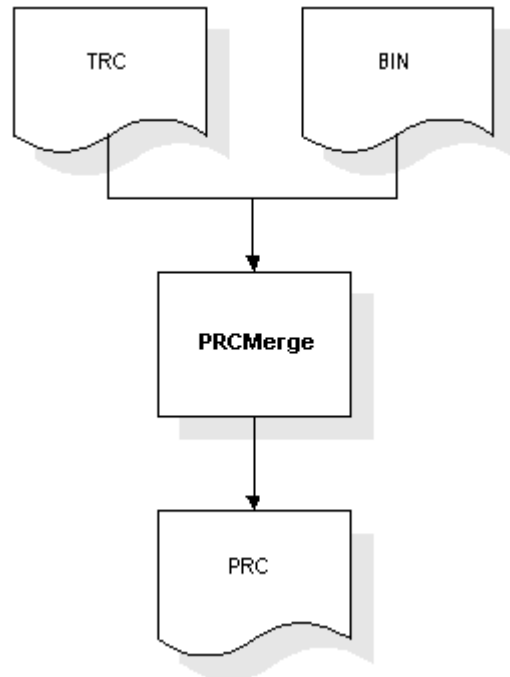
This chapter describes how to use PRCMerge to combine a resource TRC or PRC with a code PRC to make a Palm OS® application.

- “[Introducing PRCMerge](#)” describes the basic functions of the PRCMerge tool.
- “[Using the PRCMerge Command Line Tool](#)” on page 31 describes how to specify the PRCMerge command line arguments to make a Palm OS application.

Introducing PRCMerge

PRCMerge builds a Palm OS binary resource file (PRC), typically for a Palm OS application or library. It takes as input a set of Palm OS binary resources specified as one or more input files.

Figure 4.1 PRCMerge Overview



Each input file must be one of the following formats:

- TRC/PRC file format. The resources from the input file are merged to the output.
- RSRC (Macintosh binary resource file). The resource data from Macintosh binary resource files is not processed in any form, but only merged into the output, possibly with resource type remapping (for example, mapping CODE to code).

You use RSRC files when you are building 68K applications with CodeWarrior. The CodeWarrior 68K compiler produces a Macintosh resource file with the CODE resources which are then merged into the Palm OS application file.

- BIN file. A BIN file specifies the binary data for a single resource. The name of the file must be of the form `<resource-type><resource-ID>.bin`, where `<resource-type>` is the 4-letter character resource type and `<resource-ID>` is a 4-character hexadecimal resource ID. For example "tFRM03E8.BIN" would contain the data for 'tFRM' 1000 (0x3E8 hexadecimal = 1000 decimal).

PRCMerge also takes as input the data used to build the PRC file database header, such as database name, version, type code, creator code. These values must be specified appropriately for the Palm OS application or library being built.

PRCMerge is integrated into Palm OS Developer Suite. When you build a project using Developer Suite, PRCMerge is called as part of the process of building the project.

PRCMerge is available as a command line tool for Windows.

Using the PRCMerge Command Line Tool

PRCMerge is used to merge PRC resources into a Palm OS PRC resource database file, such as a Palm OS application. PRCMerge takes as input one or more PRC format files (PRC, TRC, OPRC, and BPRC), Macintosh resource files, or BIN files.

The command syntax for PRCMerge is:

```
PRCMerge inputFiles -o outputFile [options]
```

input_files

One or more files of the following type:

TRC

PRC

Macintosh resource

BIN format

output_file

The -o option is required to specify where the PRCMerge output file is written. The file is in PRC resource database format. Normally you should specify a file with a file name extension of PRC. The output file is overwritten automatically if it already exists.

Building a Palm OS Application

Using the PRCMerge Command Line Tool

options

Additional command line options as described in the following section “[Specifying Command Line Options](#).”

Specifying Command Line Options

`-append`

If the `-append` option is specified and the output file PRC file already exists, then PRCMerge merges the compiled resources into the existing output file. If the output file does not exist, specifying this option has no effect, and the output file is be created as usual.

`-b | -dbFlagBackup`

Sets the backup flag stored in the PRC database header. Applications should normally specify the `-dbFlagBackup` option.

`-c creatorstr | -dbCreator creatorstr`

Specifies the database creator code stored in the PRC database header. This is normally the unique creator code registered for the application.

`-dbFlagBundle`

Sets the bundle flag stored in the PRC database header.

`-dbFlagExecutable`

Sets the executable flag stored in the PRC database header.

`-dbFlagExtendedDB`

Sets the extended database flag to indicate that the database is an ARM-based non-schema database.

`-h | -dbFlagHidden`

Sets the hidden flag stored in the PRC database header.

`-help`

PRCMerge displays help information and ignores any other options.

`-helpVersion`

PRCMerge displays the tool’s version information.

`-n namestr | -dbName namestr`

Specifies the database name stored in the PRC database header. This is normally the name of the application.

-only type

Indicates that you want to input only a particular resource or resources of a particular type. This option is followed by a resource type expression:

type

A resource type to indicate all resources of that type.

type (id)

A resource type followed by a resource ID to indicate a specific resource.

type (id:id)

A resource type followed by an ID range to indicate all matching resources.

This option **-type** may be specified multiple times, in which case the options are combined. For example, "**-only Talt -only tFRM**" indicates that the output PRC contains both alert and form resources.

-omit type

Indicates that you want to omit a particular resource or resources of a particular type. This option is followed by a resource type expression:

type

A resource type to indicate all resources of that type.

type (id)

A resource type followed by a resource ID to indicate a specific resource.

type (id:id)

A resource type followed by an ID range to indicate all matching resources.

This option **-omit** may be specified multiple times in which case the options are combined. For example, "**-omit Talt -omit tFRM**" indicates that the output PRC contains all resources except for alert resources and form resources.

-p | -dbFlagProtect

Sets the protect flag stored in the PRC database header.

-q | -quiet

PRCMerge suppresses the tool description (tool name and version number) and copyright information.

Building a Palm OS Application

Using the PRCMerge Command Line Tool

- r | -dbFlagReset
Sets the reset flag stored in the PRC database header.
- t *typestr* | -dbType *typestr*
Specifies the database type code stored in the PRC database header. This is normally 'appl' for Palm OS applications.
- v *versionstr* | -dbVersion *versionstr*
Specifies the database version number stored in the PRC database header. Most applications start with database version 1 and increment as necessary when new versions of the application are developed.

Comparing Palm OS Resource Database Files

This chapter describes how to use PRCCompare to find differences between two Palm OS resource database files (files of type PRC, BPRC, OPRC, and TRC).

- “[Introducing PRCCompare](#)” on page 36 describes the basic functions of the PRCCompare tool.
- “[Using the PRCCompare Command Line Tool](#)” on page 36 documents how you specify the PRCCompare command line arguments to compare Palm OS resource database files.

Introducing PRCCompare

PRCCompare compares two Palm OS binary resource database files and produces a text file report detailing the results of the comparison. PRCCompare reports resources that are in one file but not the other, resources that have different binary data between the two files, and changes to header information such as database names and internal flags.

PRCCompare can be used as a general purpose debugging tool. For example, if you are testing with a debug ROM and you receive the error message “Resource not found,” then you can use PRCCompare to compare the old and new versions of the application’s PRC files and identify the resource that is missing.

PRCCompare can also be used when doing localization work. For example, say you have an older version of an application which was translated and you only want to translate the resources that have been added in the new version of the application. You can use PRCCompare to identify the specific resources that are new or changed.

PRCCompare is available as a command line tool for Windows.

Using the PRCCompare Command Line Tool

PRCCompare is used to compare two Palm OS binary resource database files.

The command syntax for PRCCompare is:

```
PRCCompare inputFile1 inputFile2 [options]  
inputFile1, inputFile2
```

One or more files of the following type:

TRC

PRC

BPRC

OPRC

options

Additional command line options as described in the following section "[Specifying Command Line Options](#)."

By default, PRCCompare compares the database headers, application information and sort information blocks, and all of the resources, and reports all differences between the two files.

PRCCompare does not report differences in the creation, modification, and backup date and timestamp information found in the database headers, because these are normally different for different builds of the same PRC file.

Specifying Command Line Options

`-help`

PRCCompare displays help information and ignores any other options.

`-helpVersion`

PRCCompare displays the tool's version information.

`-ignoreHeader`

PRCCompare does not compare the database headers.

`-ignoreInfo`

PRCCompare does not compare the application information and sort information.

`-ignoreOvlyTimes`

PRCCompare ignores the date and timestamps in overlay resources.

Since the date and timestamp in the 'ovly' resource is updated each time a PRC overlay is built, this is an expected difference. Therefore, the binary data of these resources may be different, but the logical data is the same.

`-ignoreRes`

PRCCompare does not compare the resources.

`-ignoreResData`

PRCCompare does not compare the resource data. PRCCompare reports if a resource is in one file but not the other, but not if the resource exists in both files but has different data.

Comparing Palm OS Resource Database Files

Using the PRCCompare Command Line Tool

`-ignoreStrResPad`

PRCCompare ignores the pad bytes after string resource types.

Some resources use a string constant with a 0 byte to mark the end of data. Some versions of Palm OS development tools additionally align the resource data to an even boundary by storing an additional 0 byte past the end of the data.

The binary data of these resources may be different (depending on the development tools used), but the logical data is the same.

`-noList`

PRCCompare does not list the individual resource differences. Instead, it outputs either "No differences found" or the number of differences detected.

`-only type`

Indicates that you want to input only a particular resource or resources of a particular type. This option is followed by a resource type expression:

type

A resource type to indicate all resources of that type.

type (id)

A resource type followed by a resource ID to indicate a specific resource.

type (id:id)

A resource type followed by an ID range to indicate all matching resources.

This option may be specified multiple times, in which case the options are additive. For example, "`-only tSTR -only tSTL`" indicates that you want to compare only the string and string list resources.

`-omit type`

Indicates that you want to omit a particular resource or resources of a particular type. This option is followed by a resource type expression:

type

A resource type to indicate all resources of that type.

Comparing Palm OS Resource Database Files

Using the PRCCCompare Command Line Tool

type (*id*)

A resource type followed by a resource ID to indicate a specific resource.

type (*id:id*)

A resource type followed by an ID range to indicate all matching resources.

This option `-omit` may be specified multiple times in which case the options are combined. For example, "`-omit Talt -omit tFRM`" indicates that you want to compare all resources except for alert resources and form resources.

`-q` | `-quiet`

PRCCCompare suppresses the tool description (tool name and version number) and copyright information.

Comparing Palm OS Resource Database Files

Using the PRCCompare Command Line Tool

Localizing a Palm OS Application

This chapter describes how to use hOverlay to create localized versions of a Palm OS application (PRC file).

- [“Introducing hOverlay”](#) on page 42
- [“Preparing to Use hOverlay”](#) on page 42
- [“Understanding the hOverlay Process”](#) on page 43
- [“Using the hOverlay Command Line Tool”](#) on page 47

Introducing hOverlay

The `hOverlay` command line tool (`hOverlay.exe`) simplifies the process of localizing a Palm OS application (PRC file). It lets you create a base PRC that is not locale-specific, and then combine the base PRC with an “overlay” PRC that contains locale-specific information.

When localizing an application for a new locale, you need only create a new overlay PRC. The base PRC remains unchanged.

NOTE: Before reading this chapter, you need to know how to create a PRC. Specifically, you need to be familiar with XRD files, `PalmRC`, and `PRCMerge`. For information on `PalmRC`, see [Chapter 3, “Compiling XML Resource Files,”](#) on page 21. For information on `PRCMerge`, see [Chapter 4, “Building a Palm OS Application,”](#) on page 29. For information on XRD files, see the book *Exploring Palm OS: Resource File Formats*.

Preparing to Use hOverlay

Before using `hOverlay`, you need to split your PRC into multiple, separate PRC files:

- a PRC that is not locale-specific (this is the *non-localized PRC*)
- a locale-specific PRC for each desired locale (these are the *localized PRC files*)

Steps to Split your PRC

1. Organize your application resources into multiple XRD files rather than just one:
 - The first XRD file should contain non-locale-specific resources such as the application icon and the version number; name this file something like `MyApp_nonlocalized.xrd`.
 - The second XRD file should contain locale-specific resources such as forms for one locale. For example, if this XRD file contain United States English resources, you might name it `MyApp_enUS.xrd`.

- The third XRD should contain locale-specific resources for another locale. For example, if this XRD contains resources for France, you might name it `MyApp_frFR.xrd`.
 - If you wish, create more locale-specific XRD files (such as, for example, `MyApp_jaJP.xrd` for Japan, `MyApp_deDE.xrd` for Germany, and other locales).
2. Use `PalmRC` to compile each XRD file, creating `MyApp_nonlocalized.trc`, `MyApp_enUS.trc`, `MyApp_frFR.trc`, and other locales.
 3. Use `PRCMerge` to link the TRC files with the appropriate code resources, creating PRC files named `MyApp_nonlocalized.prc`, `MyApp_enUS.prc`, `MyApp_frFR.prc`, and other locales.

You now have a non-localized PRC and multiple localized PRC files which `hOverlay` can use.

An alternative way to create a non-localized PRC and localized PRC files is to create a single XRD file for all your application resources, using the `LOCALE` attribute to indicate resources that pertain to a specific locale. You can then use the `PalmRC overlayFilter` parameter to create multiple, separate TRC files (a non-localized TRC and one or more overlay TRC files). Then use `PRCMerge` to link each TRC file with the appropriate code resources.

For detailed information on the `LOCALE` attribute, see *Exploring Palm OS: Resource File Formats*.

Understanding the hOverlay Process

`hOverlay` takes as input the non-localized PRC and localized PRC files that you created, and uses them to create a base PRC (BPRC) and overlay PRC files (OPRC). You can then install the base PRC (containing non-locale-specific information) and the appropriate overlay PRC (containing locale-specific information) on your device, and they work together like a single PRC.

The Palm OS Launcher application knows to treat the base PRC and overlay PRC files as a single application because they share the same database type and creator, and have the database *bundle* flag set. `hOverlay` uses the base PRC file's database header values

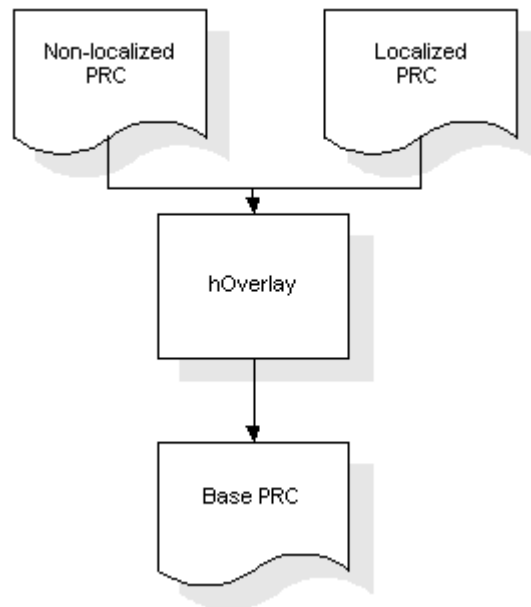
Localizing a Palm OS Application

Understanding the hOverlay Process

(type, creator, version, and flags) to set the database header values in the output files. When hOverlay builds an OPRC file, it creates the database name automatically by appending the locale value to the base PRC file's database name. Also, hOverlay automatically sets the *bundle* flag for all the overlay database headers.

hOverlay can perform three different operations, which correspond to three different modes, described in the sections that follow.

Figure 6.1 hOverlay Mode 1



Mode 1: Creating a Base PRC

The first operation mode constructs the base PRC (BPRC). As input, it takes one non-localized PRC and one localized PRC (for information on creating these inputs, see [“Preparing to Use hOverlay”](#) on page 42).

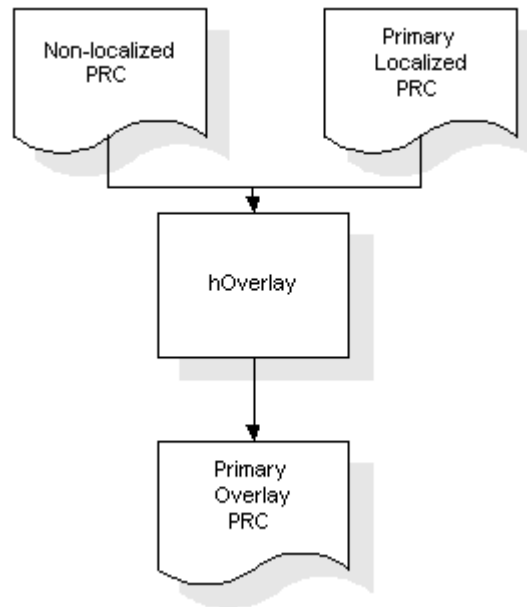
The locale of the localized PRC that you choose to use in mode 1 becomes the locale of the *primary overlay PRC* in Mode 2.

For example, you might choose to create your base PRC using the enUS (U.S. English) locale as the primary locale for this base PRC:

```
hOverlay -mode1 MyApp_nonlocalized.prc  
MyApp_enUS.prc enUS MyApp.bprc
```

This example creates a base PRC using *enUS* as the primary locale.

Figure 6.2 hOverlay Mode 2



Mode 2: Creating a Primary Overlay PRC

The second operation mode constructs an overlay PRC (OPRC) for the same primary locale used with Mode 1. The parameters are the same, except that the output file name is the same as the name of the localized PRC for the primary locale.

For example:

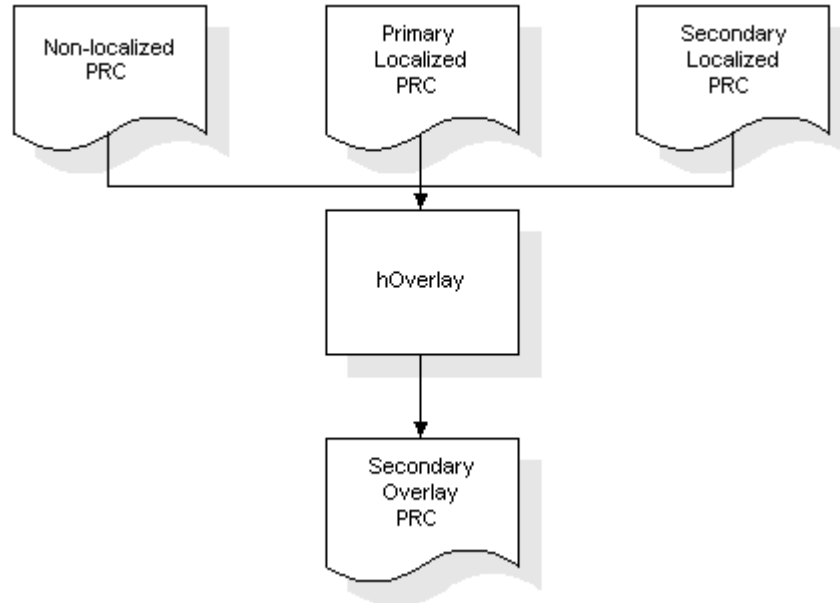
```
hOverlay -mode2 MyApp_nonlocalized.prc  
MyApp_enUS.prc enUS MyApp_enUS.oprc
```

This example creates an overlay PRC for the primary locale *enUS*.

Localizing a Palm OS Application

Understanding the hOverlay Process

Figure 6.3 hOverlay Mode 3



Mode 3: Creating Overlays for Other Locales

The third mode is used to construct overlay PRC files (OPRC) for locales other than the primary one. You must supply the following parameters as input:

- the same non-localized PRC file that you used in modes 1 and 2
- the same primary localized PRC file that you used in modes 1 and 2 (for example, if your primary locale is U.S. English, specify `enUS` here)
- the localized PRC file for the specific locale (for example, if you are creating an overlay PRC for Japanese, specify `jaJP` here)
- the name of the output overlay PRC file; this name is the same as the name of the localized PRC file for the specific locale. For example, if the localized PRC file that you used as input is `MyApp_frFR.prc`, the name of the output overlay PRC is `MyApp_frFR.oprc`.

Example

```
hOverlay -mode3 MyApp_nonlocalized.prc  
MyApp_enUS.prc MyApp_frFR.prc frFR MyApp_frFR.oprc
```

This example creates an overlay PRC for the *frFR* locale.

Using the hOverlay Command Line Tool

hOverlay is a Windows command line tool. hOverlay can be called from a batch file when building a PRC.

The command line syntax is as follows (in this example, the primary locale is *enUS* but you can use a different locale as your primary locale, if you wish):

```
hOverlay -mode1 nonlocalized.prc  
primary_locale.prc llCC  
output.bprc
```

```
hOverlay -mode2 nonlocalized.prc  
primary_locale.prc llCC  
output_llCC.oprc
```

```
hOverlay -mode3 nonlocalized.prc  
primary_locale.prc secondary_locale.prc xxYY  
output_xxYY.oprc
```

Specifying Command Line Options

-mode1

This mode, **-mode1**, generates only the base PRC, which contains only the non-locale-specific resources.

-mode2

This mode, **-mode2**, generates only the primary locale's overlay PRC.

-mode3

This mode, **-mode3**, generates a specified locale's overlay PRC.

nonlocalized.prc

A PRC file containing only the non-locale-specific resources.

Localizing a Palm OS Application

Using the hOverlay Command Line Tool

primary_locale.prc

A PRC file containing only the localized resources for the primary locale (in this example, the primary locale is enUS, but it could be any locale).

secondary_locale.prc

A PRC file containing localized resources for some locale other than the primary locale.

l1CC

The primary locale associated with the *primary_locale.prc* file. In most of the examples, this primary locale is enUS for United States English.

xxYY

The secondary locale associated with the *secondary_locale.prc* file.

output.bprc

The filename of the base PRC to be generated by hOverlay. For example, this could be *myprc.bprc*.

output_l1CC.oprc

The filename of the primary locale's overlay PRC, to be generated by hOverlay. The output name should always include the locale. For example, this could be *myprc_enUS.oprc*.

output_xxYYY.oprc

The file name of another locale's overlay PRC, to be generated by hOverlay. The output file name should always include the locale. For example, *myprc_frFR.oprc* for French, or *myprc_jaJP.oprc* for Japanese.

-noOvlyWarn

Turns off warning messages when overlay resources already exist in the source BPRC and OPRC files. This flag works in mode3 only.

Securing a Palm OS Application

This chapter describes how to use PRCCert and PRCSign to create and embed a digital signature and associated certificates in a Palm OS application (PRC file).

- [“Introducing PRCCert”](#) on page 50
- [“Using the PRCCert Command Line Tool”](#) on page 51
- [“Introducing PRCSign”](#) on page 53
- [“Using the PRCSign Command Line Tool”](#) on page 54

For a general description of the security and encryption features of Palm OS Cobalt, see *Exploring Palm OS: Security and Cryptography*.

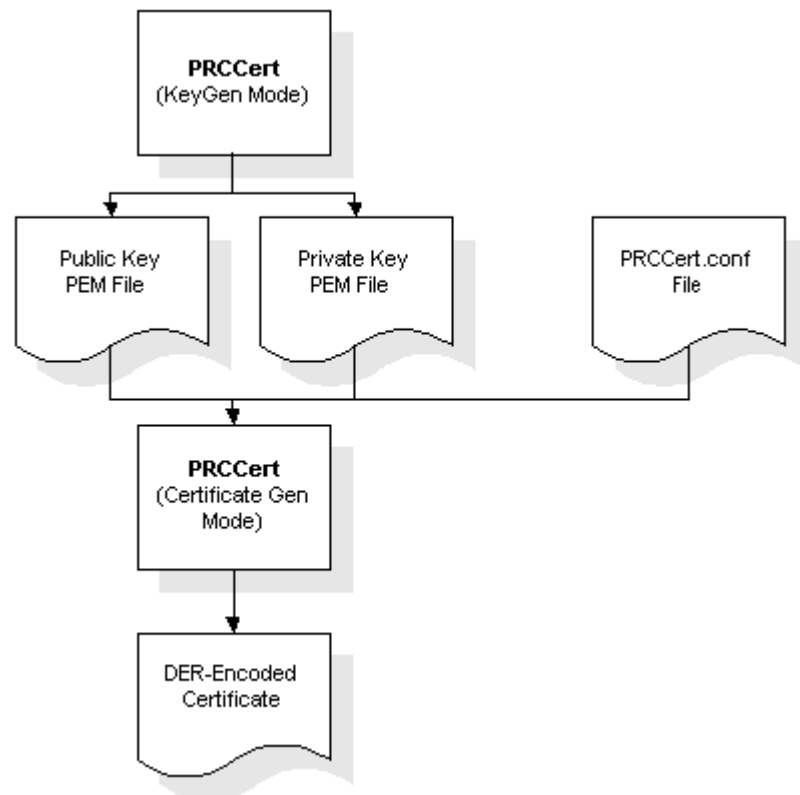
Introducing PRCCert

PRCCert is a command line tool that you can use to create your own RSA key pairs and digital certificates. You may create self-signed certificates for testing, or certificates that are signed by other private keys.

PRCCert creates RSA public/private key pairs at 1024-bit length in PEM format. PRCCert also generates a public certificate file in DER format.

The output files from PRCCert are used as input files to PRCSign, as described in [“Introducing PRCSign”](#) on page 53.

Figure 7.1 PRCCert Overview



Using the PRCCert Command Line Tool

There are two modes for PRCCert:

- Key Generation
- Certificate Generation

Specifying Key Generation Mode

The command syntax for PRCCert's key generation mode is:

```
PRCCert [options] -keygen -passphrase password
```

-keygen
Tells PRCCert to process in key generation mode.

-passphrase *password*
Specifies the pass phrase that was used to encrypt the private keyfile.

options
Additional command line options as described in the following section "[Specifying PRCCert Command Line Options](#)."

Specifying Certificate Generation Mode

The command syntax for PRCCert's certificate generation mode is:

```
PrcCert [options] -certgen  
-configfile filepath  
-keyfile filepath  
-passphrase password
```

-certgen
Tells PRCCert to process in certificate generation mode.

-configfile *filepath*
filepath specifies the full path and filename for the configuration file. The configuration file specifies the information that is embedded in the certificate (owner, date, location, and other information). See the provided sample file `PRCCert.conf` for more information (included in the developer tools package).

Securing a Palm OS Application

Using the PRCCert Command Line Tool

- keyfile *filepath*
filepath specifies the full path and filename for the private key file (a PEM format key file).
- passphrase *password*
Specifies the pass phrase that was used to encrypt the private keyfile.
- options*
Additional command line options as described in the following section "[Specifying PRCCert Command Line Options](#)."

Specifying PRCCert Command Line Options

- basename *name*
Specifies the base name for the output files.
- help
Tells PRCCert to display usage description without doing any processing.
- outputdir *path*
Specifies the directory for the output files.
- quiet
Suppresses the tool description (tool name and version number) and copyright information.

Examples

Creating a Key Pair

To create a key pair, you run PRCCert with the following command line options:

```
PRCCert -keygen -passphrase "myPassPhrase"
```

The output is two files: a public key PEM file and a private key PEM file.

Creating a Self-Signed Certificate

To create a self-signed certificate, you first need to do the following:

- Create a key pair (a public key PEM file and a private key PEM file), as described in the example above.

- Edit the configuration file, `PRCCert.conf` (provided in the developer tools kit), updating the security file information, and changing the company name and location as appropriate.
- Run `PRCCert` with the following command line options:

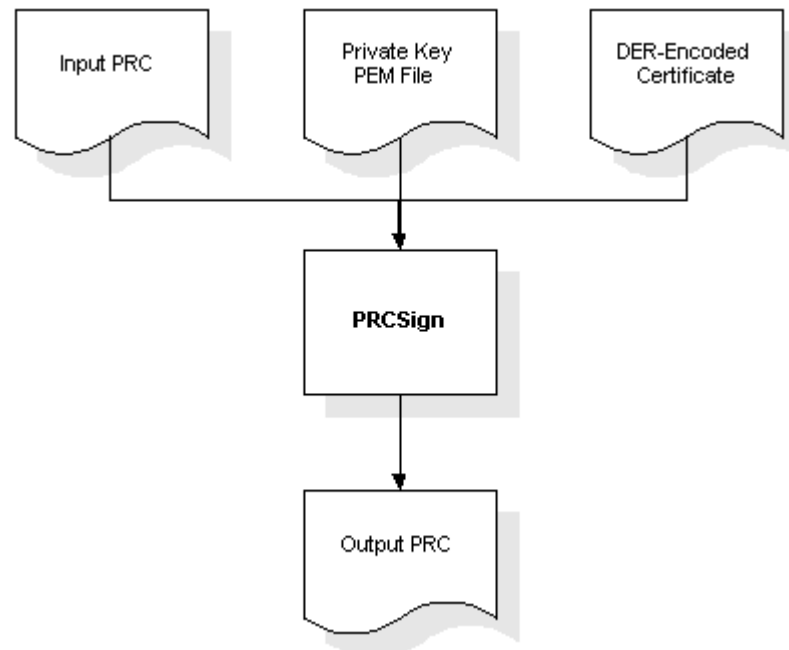
```
PRCCert -certgen  
-passphrase "myPassPhrase"  
-configfile c:\DirName\EditedPRCCert.conf  
-keyfile private_key_filename
```

The output is a DER-encoded x509 certificate that can be used with `PRCSign`.

Introducing PRCSign

`PRCSign` is a command line tool that you use to digitally sign your applications or to embed digital signature certificates in your applications.

Figure 7.2 PRCSign Overview



Securing a Palm OS Application

Using the PRCSign Command Line Tool

PRCSign creates a digital signature for a particular PRC using an asymmetric key cipher, storing the signature into the PRC as a resource of type 'sign'. The signature can be verified as authentic by using your public key to decipher the signature resource. Each application has at most one 'sign' resource with a resource ID of 1000.

PRCSign takes your private key and signs a SHA1 hash of all of the static (unchanging) resources in the PRC along the signature attributes. PRCSign then adds the resulting output as the 'sign' resource to your application PRC file.

PRCSign also takes a digital signature and adds it to the PRC as a 'cert' resource in such a way that the Palm OS Security Manager can retrieve it for application certification.

PRCSign takes your certificate and adds it to the PRC as a 'cert' resource with the certificate ID being a digest (SHA1) of the certificate itself. The certificates need to be in X.509 format and they need to be ASN.1 DER-encoded files. These certificates can be created by any certificate creation tool that generates this format. (For example, OpenSSL has such a tool).

Using the PRCSign Command Line Tool

The command syntax for PRCSign is:

```
PRCSign inputFile1 [options] [opt_file]
```

inputFile

One PRC input file must be specified as input to PRCSign, of the following type:

TRC

PRC

BPRC

OPRC

options

Additional command line options as described in the following section "[Specifying PRCSign Command Line Options](#)."

opt_file

You can place the options in a text file rather than specifying them on the command line.

Specifying PRCSign Command Line Options

Informational Options

-help

Tells PRCSign to display usage description without doing any processing.

-helpVersion

Tells PRCSign to display tool version information without doing any processing.

-quiet

Suppresses the tool description (tool name and version number) and copyright information.

Required Options

-keyfile *private_key_file*

Specifies the private key file.

private_key_file

The full path and filename to the PEM private key file. *private_key_file* is your private key file. The keyfile must be in PEM-encoded format (the format that PRCCert generates).

-passphrase *pass_phrase*

Specifies the pass phrase for the private key file.

pass_phrase

The pass phrase for the private key file. PEM files are encrypted with a pass phrase, so you must specify it so that PRCSign can access the private key. This is the same pass phrase you passed into PRCCert when the key file was generated.

-cert *certificate_file*

This option is used both when you sign a PRC and when you add a certificate to a PRC. When you are signing a PRC, the certificate is used to verify the signature. When you are

Securing a Palm OS Application

Using the PRCSign Command Line Tool

adding a certificate to a PRC, *certificate_file* specifies the data that you want embedded in the 'cert' resource.

certificate_file

The full path and filename to the public key certificate file (a DER-encoded digital certificate file).

Other Options

`-addcert`

Specifies that PRCSign should add the certificate (specified with the `-cert` option) to the PRC. You may only add one certificate at a time. If you need to embed multiple certificates, you need to run the tool once for each certificate.

`-o prcFile`

Specifies the output PRC filename. The output filename can be the same as the input filename.

`-ocert certificate_file`

For overlay certification, tells PRCSign to add an ID for the certificate specified by *certificate_file*. (You must have specified a certificate to add to this list with the `-cert` option.) Certificate IDs are used to validate signed overlay resources.

To add multiple certificate IDs, specify this parameter once for each *certificate_file*. You must set the list of multiple certificate IDs at the time of signing; this list may not be modified after the first signature is applied to the application.

`-omit resSpec`

Tells PRCSign to exclude the resource specified by *resSpec* from the signature. This option can be useful if a resource is expected to change at runtime.

`-remove_sign`

Removes the 'sign' resource from the input PRC file.

`-remove_cert cert_id`

Removes the 'cert' resource with the ID *cert_id* from the input PRC file.

`-scert certificate_file`

For shared library verification, tells PRCSign to add an ID for the certificate specified by *certificate_file*. (You

must have specified a certificate to add to this list with the `-cert` option.)

To add multiple certificate IDs, specify this parameter once for each *certificate_file*. You must set the list of multiple certificate IDs at the time of signing; this list may not be modified after the first signature is applied to the application.

`-smartcard provider opts`

Specify a *provider* to use with a smart card and additional options (specified by *opts*) that may be necessary for the provider. The only smart card provider supported is the WinCrypt with GemPlus reader.

Note: If you use this option, you do not need the `-key` or the `-pass` parameters.

`-verbose`

Outputs extensive diagnostic information about what data is being processed.

Securing a Palm OS Application

Using the PRCSign Command Line Tool

Index

Symbols

@paramFile option, GenerateXRD 10

A

append option, PalmRC 23, 32

B

b option, PRCMerge 32

BIN file 31

bitmapFilePrefix option, GenerateXRD 11

bitmaps option, GenerateXRD 10

BPRC file 10

build tools 4

building Palm OS applications 29

C

c option, PRCMerge 32

comparing base PRCs and overlay PRCs 35

comparing Palm OS resource database files 35

converting PRC files 7

converting RSRC files 7

D

d option, GenerateXRD 11

dbCreator option, PRCMerge 32

dbFlagBackup option, PRCMerge 32

dbFlagBundle option, PRCMerge 32

dbFlagExtendedDB option, PRCMerge 32

dbFlagHidden option, PRCMerge 32

dbFlagProtect option, PRCMerge 33

dbFlagReset option, PRCMerge 34

dbName option, PRCMerge 32

dbType option, PRCMerge 34

dbVersion option, PRCMerge 34

densityTarget option, PalmRC 23

developer tools vi

E

encoding option, GenerateXRD 11

excluding resource types 33, 38

externDataFiles option, GenerateXRD 12

externFilesDir option, GenerateXRD 12

G

GenerateXRD

command line syntax 10

functions 8

GenerateXRD command line options

@paramFile 10

bitmapFilePrefix 11

bitmaps 10

d 11

encoding 11

externDataFiles 12

externFilesDir 12

help 12

helpTargets 13

locale 13

mapInvalidChars 15

noLocaleCheck 15

noRsrcNameInfo 15

noSort 15

noUnknown 15

omit 16

only 16

q 17

quiet 17

raw 17

remapChars 17

s 18

target 18

H

h option, PRCMerge 32

help option, GenerateXRD 12

help option, PalmRC 23

help option, PRCCompare 37

help option, PRCMerge 32

helpTargets option, GenerateXRD 13

helpTargets option, PalmRC 24

helpversion option, PalmRC 24

helpversion option, PRCCompare 13, 37

helpversion option, PRCMerge 32

I

i option, PalmRC 24

ignoreheader option, PRCCompare 37

ignoreinfo option, PRCCompare 37
ignoreovlytimes option, PRCCompare 37
ignoreres option, PRCCompare 37
ignoreresdata option, PRCCompare 37
ignorestrrespad option, PRCCompare 38
including resource types 33, 38

L

locale option, GenerateXRD 13
locale option, PalmRC 24

M

Macintosh binary resource files 8
makeDeps option, PalmRC 25
mapInvalidChars, GenerateXRD 15
merging PRCs and TRCs 29
migration tool 4

N

n option, PRCMerge 32
nolist option, PRCCompare 38
noLocaleCheck option, PalmRC 25
noLocaleCheck, GenerateXRD 15
noRsrcNameInfo option, GenerateXRD 15
noSort option, GenerateXRD 15
noUnknown option, GenerateXRD 15
noWarnSize option, PalmRC 25

O

omit option, GenerateXRD 16
omit option, PRCCompare 38
omit option, PRCMerge 33
only option, GenerateXRD 16
only option, PRCCompare 38
only option, PRCMerge 33
OPRC file 10
overlayFilter option, PalmRC 25

P

p option, PalmRC 26
p option, PRCMerge 33
Palm OS binary resource files 8

Palm OS Developer Suite vi

PalmRC

command line syntax 23
functions 22

PalmRC command line options

append 23, 32
densityTarget 23
help 23
helpTargets 24
helpversion 24
i 24
locale 24
makeDeps 25
noLocaleCheck 25
noWarnSize 25
overlayFilter 25
p 26
q 26
quiet 26
target 27

PLR file 10

PRC file 10, 30

PRC file conversion 7

PRCCompare

command line syntax 36

PRCCompare command line options

help 37
helpversion 13, 37
ignoreheader 37
ignoreinfo 37
ignoreovlytimes 37
ignoreres 37
ignoreresdata 37
ignorestrrespad 38
nolist 38
omit 38
only 38
q 39
quiet 39

PRCMerge

command line syntax 31
functions 30

PRCMerge command line options

b 32
c 32
dbCreator 32

- dbFlagBackup 32
- dbFlagBundle 32
- dbFlagExtendedDB 32
- dbFlagHidden 32
- dbFlagProtect 33
- dbFlagReset 34
- dbName 32
- dbType 34
- dbVersion 34
- h 32
- help 32
- helpversion 32
- n 32
- omit 33
- only 33
- p 33
- q 33
- quiet 33
- r 34
- t 34
- v 34

Q

- q option, GenerateXRD 17
- q option, PalmRC 26
- q option, PRCCompare 39
- q option, PRCMerge 33
- quiet option, GenerateXRD 17
- quiet option, PalmRC 26
- quiet option, PRCCompare 39
- quiet option, PRCMerge 33

R

- r option, PRCMerge 34
- raw option, GenerateXRD 17
- remapChars, GenerateXRD 17
- resource tools

- build tools 4
- migration tool 4
- RSR file 10
- RSRC file 10, 30
- RSRC file conversion 7

S

- s option, GenerateXRD 18

T

- t option, PRCMerge 34
- target option, GenerateXRD 18
- target option, PalmRC 27
- temporary resource file 22
- tools documentation vi
 - Introduction to Palm OS Tools vi
 - Language and Library Reference vii
 - Palm OS Debugger vii
 - Palm OS Resource Editor vii
 - Resource Tools vii
 - Virtual Phone vii
- TRC file 10, 30
 - definition 22

U

- Unicode encoding 11
- UTF16 11
- UTF8 11

V

- v option, PRCMerge 34

X

- XRD file 10
 - definition 2

