



Palm OS® Resource Editor Guide

Palm OS® Developer Suite

Written by Brian Maas and Jenny Green.

Engineering contributions by Keith Rollin, Brad Jarvinen, Eric Cloninger, and JB Parrett.

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, Graffiti, 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 Editor Guide

Document Number 3131-002

November 11, 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 Document	vii
What This Book Contains	vii
Palm OS Developer Suite Documentation	viii
Suggested Reading	ix
Additional Resources	ix
 1 Palm OS Resource Editor Fundamentals	 1
Palm OS Resource Editor Fundamentals Overview	1
What Is a Palm OS Resource?.	2
Palm OS Resource Tools	2
Resource Files	4
Resource Workspaces	4
Resource Description Files.	5
Custom Resource Description Files	7
Resource Header Files	7
Opening a Resource Description File in Resource Editor	8
Resource Editor Interface	9
Main Window	9
Windows Pane.	10
Files Pane	12
Tasks Pane	13
Properties Pane	14
Keyboard Toolbar Button	17
Resource Types.	18
Forms and Form Objects	18
Menus	19
Character Strings, String Lists, and Category Names	19
Alerts.	20
Icons and Bitmaps	20
Fonts	20
Sounds	20
Data Types	20
Palm OS Protein Resource Types and 68K Resource Types	21

2 Working with Application Resources	23
Overview of Application Resources	23
Using the Application Preferences Editor	24
Understanding the App Prefs Editor	24
Using the Application Extended Preferences Editor	24
Understanding App Extended Prefs Editor	25
Using the Application Information Strings Editor	25
Understanding the App Info String Editor	25
 3 Working with Forms	 27
Working with Forms Overview	27
Creating a New Form	27
Modifying an Existing Form	28
Editing a Form's Title	28
Editing Popup Objects.	29
Using the Toolbar.	29
Using the Toolbox	30
The Pointer Tool	30
Toolbox Pane	30
Creating a Form Navigation Resource	57
Hand-Coding the Form Navigation Resource	57
Generating the Form Navigation Resource.	57
 4 Working with Menus	 59
Working with Menus Overview	59
Creating a New Menu Bar	59
Creating a New Menu Item	60
Modifying an Existing Menu.	60
Using the Menu Editor	60
 5 Working with Strings	 61
Working with Strings Overview	61
Creating New Strings	61
Modifying an Existing String	62
Modifying an Existing List of Strings	62
Editing Strings and String Lists.	62

Using the String Editor	62
Using the String List Editor	63
6 Working with Alert Dialogs	65
Working with Alerts Overview	65
Creating a New Alert Dialog.	65
Modifying an Existing Alert Dialog.	66
Adding a Button to an Alert Dialog.	66
Deleting a Button from an Alert Dialog	66
7 Working with Icons, Bitmaps and Other Images	67
Overview of Icons, Bitmaps, and Other Images	67
Creating New Bitmaps	67
Modifying an Existing Bitmap	68
Using the Bitmap Editor	68
Setting Image Family Properties	68
Viewing Image Family Elements	69
Adding and Deleting Image Family Elements	69
Importing Images	70
Using the Color Palette	71
Using the Bitmap Editor Toolbar	71
Working with Other Formats—Graphic Families	72
Creating a New Graphic Family Resource	72
Importing Images to a Graphic Family	73
8 Working with Other Resource Types	75
Overview of Other Resource Types and How to Work with Them	75
Using the XML Editor	75
Creating a New Resource	76
Modifying an Existing Resource	76
Working with the Hex Viewer	76
9 Menu Reference	79
Menu Reference Overview.	79
File	79
Edit	82

Form	85
Image	86
Window	88
Help	90

Index	91
--------------	-----------

About This Document

Palm OS Resource Editor Guide provides conceptual, guidance, and reference information for developers who want to use Palm OS Resource Editor to create resources for Palm OS® applications and shared libraries.

Palm OS Resource Editor runs on Windows 2000 and Windows XP.

What This Book Contains

This book has the following organization:

- [Chapter 1, “Palm OS Resource Editor Fundamentals,”](#) on page 1, provides an overview of how to use Palm OS Resource Editor to create resources for Palm OS applications and shared libraries.
- [Chapter 2, “Working with Application Resources,”](#) on page 23 describes how to work with application-specific resources.
- [Chapter 3, “Working with Forms,”](#) on page 27, tells how to create and edit forms, including dialogs and about boxes.
- [Chapter 4, “Working with Menus,”](#) on page 59, tells how to create and edit menu bars and menus.
- [Chapter 5, “Working with Strings,”](#) on page 61, tells how to create and edit character strings and help text.
- [Chapter 6, “Working with Alert Dialogs,”](#) on page 65, tells how to create and edit alerts.
- [Chapter 7, “Working with Icons, Bitmaps and Other Images,”](#) on page 67, tells how to create and edit icons, icon families, bitmaps, and bitmap families.
- [Chapter 8, “Working with Other Resource Types,”](#) on page 75, tells how to add other resources to your application.
- [Chapter 9, “Menu Reference,”](#) on page 79, provides a reference of the menus and menu items in the Palm OS Resource Editor application.

About This Document

Palm OS Developer Suite Documentation

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.
<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

Document	Description
<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.

Suggested Reading

This book describes how to use Palm OS Resource Editor to create XML-based resource description (XRD) files for Palm OS applications. However, this book does not describe the XML formats used in XRD files.

The XML formats used in XRD files are documented in the book *Palm OS Resource File Formats*, which is part of the Palm OS Developer Suite package.

In addition, to compile the XRD files, you will need to use the tool PalmRc. PalmRc is documented in the book *Palm OS Resource Tools Guide*.

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>

About This Document

Additional Resources

- 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/>

Palm OS Resource Editor Fundamentals

This chapter introduces you to and provides a high-level overview of Palm OS Resource Editor.

Palm OS Resource Editor Fundamentals Overview

Palm OS Resource Editor is a visual resource editor that you can use to create and edit XML resource description (XRD) files for Palm OS applications.

The following topics are included in this chapter.

- [“What Is a Palm OS Resource?”](#) on page 2
- [“Palm OS Resource Tools”](#) on page 2
- [“Resource Files”](#) on page 4
- [“Opening a Resource Description File in Resource Editor”](#) on page 8
- [“Resource Editor Interface”](#) on page 9
- [“Resource Types”](#) on page 18

What Is a Palm OS Resource?

A Palm OS software **resource** is a data structure that describes the characteristics of an application element. For example, user interface elements, such as forms, dialogs, or text strings are all resources; in addition, code segments are also stored in resources.

A Palm OS software application uses resources to present something to the user. For example, a database application for Palm OS software might have one form for entering and editing records, and another form for displaying a list of records. Each form is stored in the application as a form resource, which contains a description of the buttons, text fields, and other elements that make up the form. In addition, the same application may use other resources to store help text, a default list of category names for the database, bitmap pictures, and an application icon.

All resources are assigned an integer resource ID. For applications, resource IDs should be less than 9999. Resource IDs of 10000 or above are reserved for use by Palm OS.

For information on using Resource Editor to create a new resource, see “[Creating a New Resource](#)” on page 6.

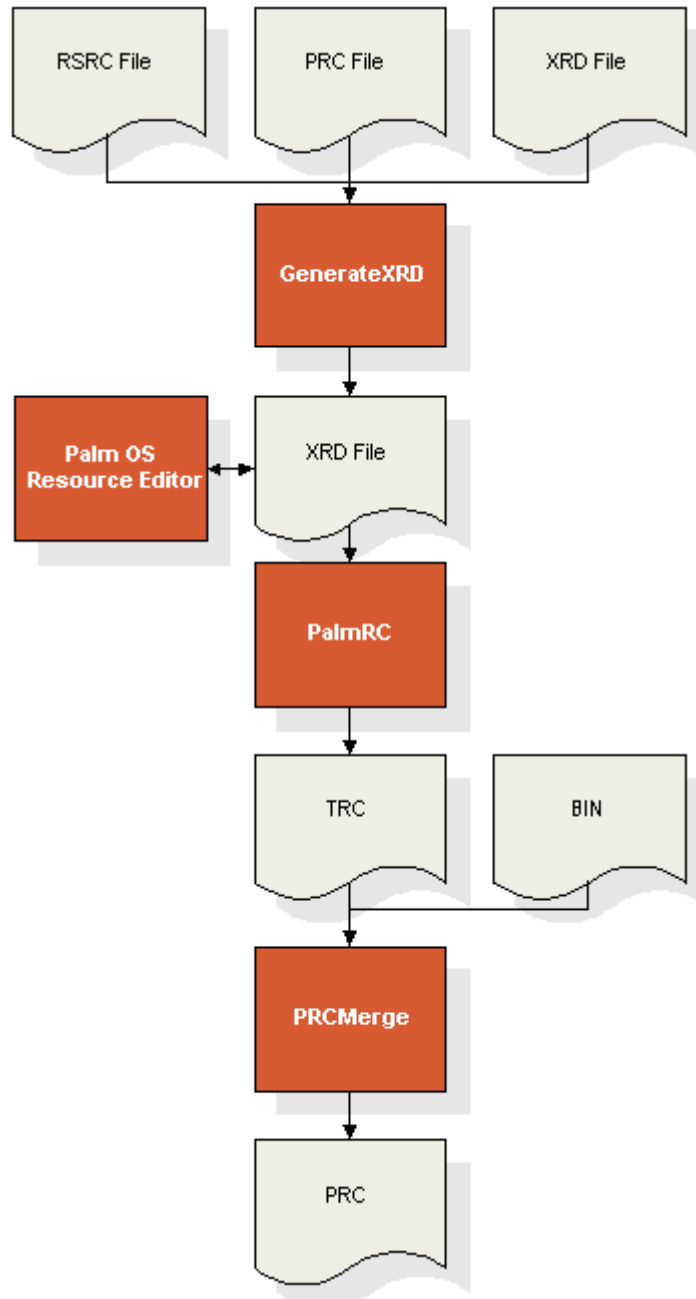
Palm OS Resource Tools

Palm OS Resource Editor is designed to work with the family of Palm OS resource tools. All of the resource tools operate on XRD files. An **XRD file** is an XML resource description file, which contains XML tags that represent the resources for an application.

[Figure 1.1](#) on page 3 provides a graphical representation of how the Palm OS resource tools are used to create resources for a Palm OS application.

- GenerateXRD converts existing Palm OS PRC files and Macintosh RSRC files into XRD files.
- Palm OS Resource Editor creates and edits XRD files.
- PalmRC compiles XRD files into TRC files.
- PRCMerge takes the compiled output from PalmRC and merges it with the code resources (BIN) to produce a Palm OS application (PRC).

Figure 1.1 Palm OS resource tools



Resource Files

Palm OS Resource Editor uses a collection of files as a convenient way to manage resource definitions.

Resource Workspaces

Resource Editor uses the concept of a **workspace** for organizing resource files. A workspace is simply a collection of resource files that are loaded, saved, and closed as a group. Physically, a workspace file is a document containing a list of resource files that are collected together.

Workspace files have the extension *XRW*.

When you first create a workspace, it appears as an empty workspace in the Files pane. When you create or add resource files to the workspace, the workspace and all files it references are shown in a tree view in the Files pane.

You can open only one workspace at a time. If you try to create a new workspace or open an existing workspace while another workspace is open, you are prompted first to save the current workspace if you have made changes to it. The current workspace is closed before the new workspace is opened or created.

Opening a Workspace

To open a workspace:

1. In the Start menu on your Microsoft Windows Desktop, select **Programs > PalmSource > Tools > Palm OS Resource Editor**. The Palm OS Resource Editor window opens.
2. In Palm OS Resource Editor menu, select **File > Open**. The Open dialog box appears.
3. Browse to the workspace (*XRW*) file and double-click it to open it. The workspace name appears in the Files pane.

Within Palm OS Developer Suite, you can also open a workspace by double-clicking an existing *XRD* file.

Creating a New Workspace

To create a new workspace:

1. In the Start menu on your Microsoft Windows Desktop, select **Programs > PalmSource > Tools > Palm OS Resource Editor**. The Palm OS Resource Editor window opens.
2. In Palm OS Resource Editor menu, select **File > New**. The New dialog box appears.
3. In the New dialog box, select **XML Resource Workspace** and click OK. A new workspace appears in the Files pane.
4. To rename the workspace, right-click the workspace name in the Files pane and select **Save as** from the pop-up menu.
5. To add resources to the workspace, see “[Creating a New Resource Description](#)” on page 5.

Resource Description Files

Palm OS resource description files use XML structure and text encoding to specify application resources. These XML resource description files are commonly called XRD files.

XRD files follow a format described in the book *Exploring Palm OS: Resource File Formats*. There is a one-to-one correspondence between Palm OS resources and XML resource descriptions in the XRD file. That is, for every resource in the Palm OS application, you need to specify an XML resource element name and associated attributes.

Resource Editor streamlines the creation of XRD files. Rather than directly editing an XML-based text file, you use the visual layout windows provided by Resource Editor to create your XRD files.

Creating a New Resource Description:

To create a new resource description (XRD) file:

1. In the Start menu on your Microsoft Windows Desktop, select **Programs > PalmSource > Tools > Palm OS Resource Editor**. The Palm OS Resource Editor window opens.
2. If you want to add the resource description to an existing workspace, open the workspace as described in “[Opening a Workspace](#)” on page 4. If you want the resource description to be part of a new workspace, do not select a workspace; a

new workspace will be created automatically when you create the resource description.

3. In Palm OS Resource Editor menu, select **File > New**. The New dialog box appears.
4. In the New dialog box, select **XML Resource Description** and click OK. A new resource description appears in the Files pane, listed under the name of the workspace to which the resource description belongs
5. To rename the resource description, right-click the workspace name in the Files pane and select **Save as** from the pop-up menu.
6. To add resources to the resource description, see “[Creating a New Resource](#).”

Within Palm OS Developer Suite, you can create a new resource description by selecting **File > New > XRD File**.

Creating a New Resource

In addition to using Resource Editor to create new resource description files, you can also use Resource Editor to create individual resources (such as forms, alerts, and menus) within a resource description file. To create an individual resource:

1. In the Files pane, select the resource description (XRD) file to which you want to add a resource.
2. In the main menu bar, select **Edit > New Resource**. The New Resource dialog box appears.
3. In the New Resource dialog box, select the resource type that you want to create.
4. If you wish, you can select the following checkboxes:
 - Show only common resource types – Select this checkbox to display only the most common resource types. Deselect this checkbox to display all the resource types that can be edited in the Resource Editor.
 - Generate Unique ID – This checkbox controls the ID assigned to the newly created resource. If this checkbox is selected, Resource Editor chooses an ID for the resource that is unique among all the other resources of the same type in the document receiving the new resource. If this

checkbox is not selected, you can type a resource ID of your choice in the Resource ID text box.

5. Click the **New** button to create the resource.

The name and number of the new resource appear in the Files pane, the resource's properties appear in the Properties pane, and the resource itself appears in an appropriate editor window (for example, if the resource is a form, it is displayed in the Form Editor).

Custom Resource Description Files

The XRD file format allows for a special type of resource called a **custom resource**. Using the custom resource type element, you can create your own XML-based format for resources that are not already defined.

Resource Editor allows you to use the Raw Editor to import any arbitrary chunk of data as a custom resource type. Custom resource type description files must have an XRT file extension.

Resource Header Files

Resource Editor does not currently support creating a C header file for the resources defined in your XRD files. If your application needs a C header file with preprocessor definitions for your resources, you need to create the header file independently.

Opening a Resource Description File in Resource Editor

To open a resource description (XRD) file in Resource Editor:

1. In the Start menu on your Microsoft Windows Desktop, select **Programs > PalmSource > Tools > Palm OS Resource Editor**. The Palm OS Resource Editor window opens.
2. In Palm OS Resource Editor menu, select [File > Open](#), and browse to the resource file you want to open. Double-click the file name to open the file in Resource Editor.
3. The resource file appears in the Files pane, with a list of resources contained in the file. The selected resource's properties are displayed in the Properties pane. You can use the [Resource Editor Interface](#) to edit the resource file.

Resource Editor Interface

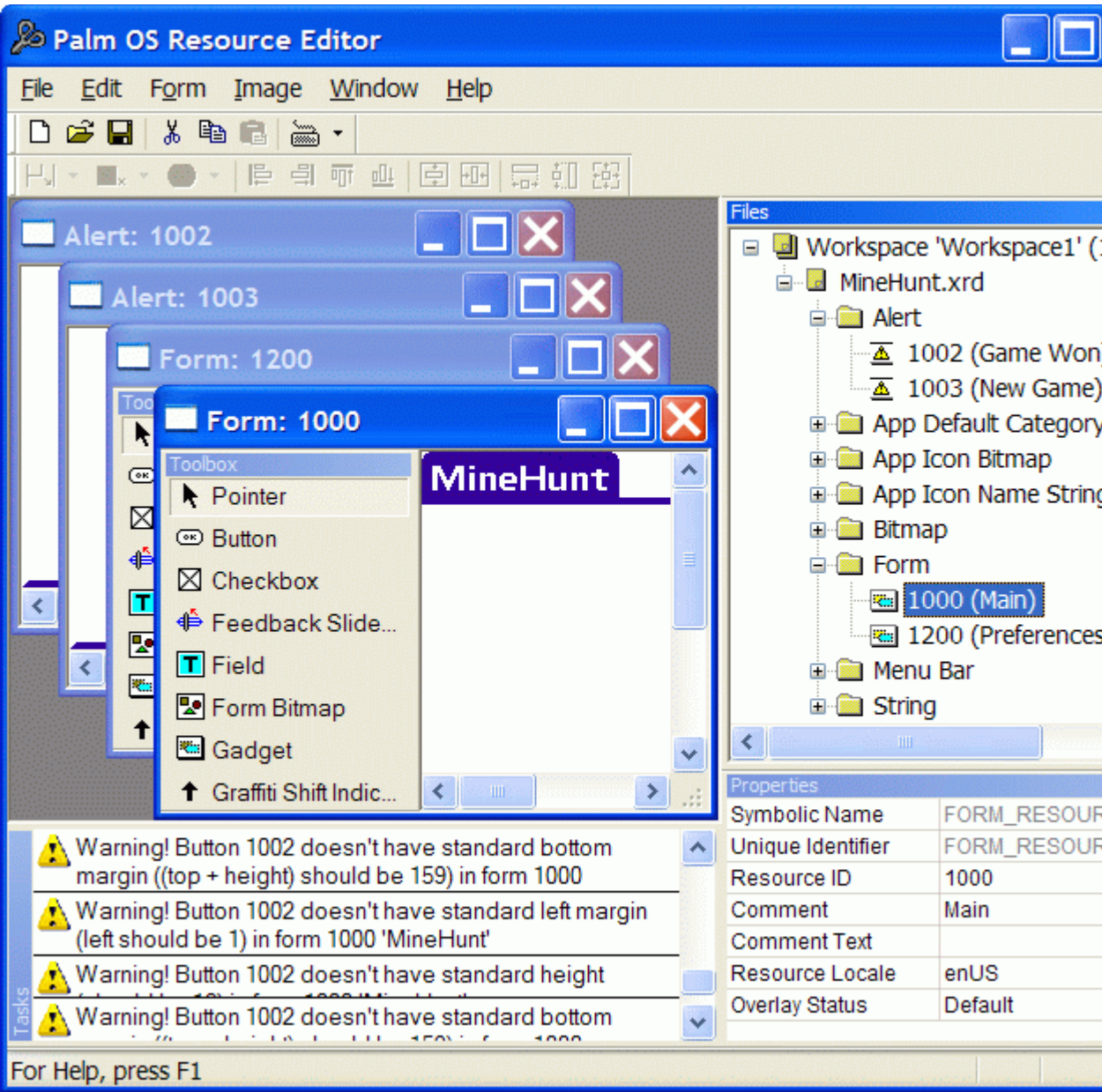
Palm OS Resource Editor consists of a collection of panes that provide access to a list of resource files, a palette of user interface controls, a graphical layout area, and a set of editors for each resource type. It also includes a Keyboard toolbar button that lists locales and lets you select fonts appropriate to each locale.

Main Window

The Resource Editor's main window, shown in [Figure 1.2](#) on page 10, provides a menu bar and toolbar, and consists of the following panes:

- [Windows Pane](#)
- [Files Pane](#)
- [Tasks Pane](#)
- [Properties Pane](#)

Figure 1.2 Resource Editor main window

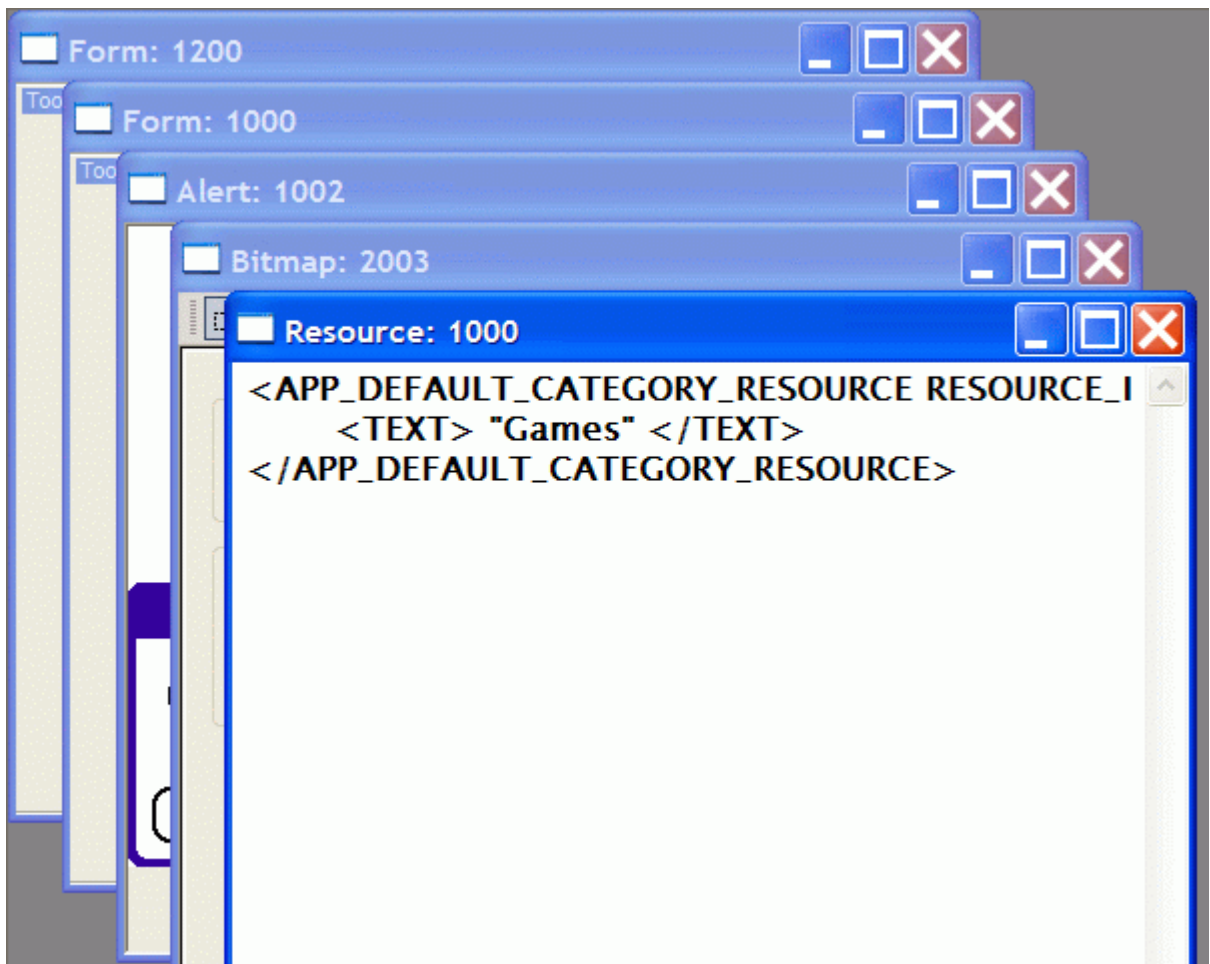


Windows Pane

The Windows pane, shown in [Figure 1.3](#), is where Palm OS Resource Editor displays the resource editors. You can use the **Windows** menu items to change how the editor windows are

displayed, to close the editor windows, or to select which editor window is in the foreground.

Figure 1.3 Windows pane, with cascading windows



Resource editors include:

- Custom resource editors for the most common resource types, including:
 - Form
 - Alert
 - Menu
 - Bitmap

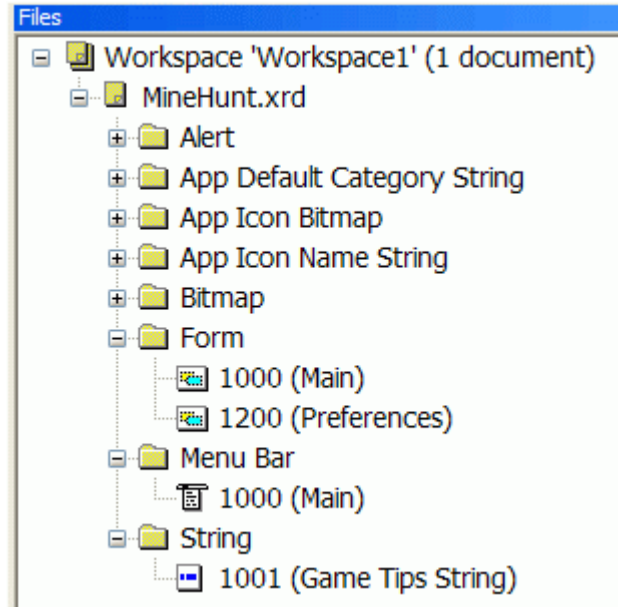
- String
- String List
- Application Preferences
- Application Extended Preferences
- Application Information Strings
- An XML Editor that shows the XML resource description for the resource. This editor is to be used for resources for which no custom resource editor is provided. For information on how to select this editor, see “[Edit Resource](#)” on page 84.
- A Hex Viewer that shows the binary form of the resource. For information on how to select this viewer, see “[Edit Resource](#)” on page 84.

Files Pane

The Files pane, shown in [Figure 1.4](#), contains a tree view of all the open files and their contents. The tree view has four levels:

1. Workspace name: You can only open one workspace. The label shows the workspace name, and shows the number of open documents in the workspace.
2. XRD filename: The label shows the filename (or it says *Untitled* if the file has not yet been saved).
3. Resource type: All resources in an XRD file are sorted by resource type. The label shows either the name of the resource type or the resource type's four-character ID.
4. Resource identifier: The label shows the resource ID for the specific resource, and any associated text string for the resource.

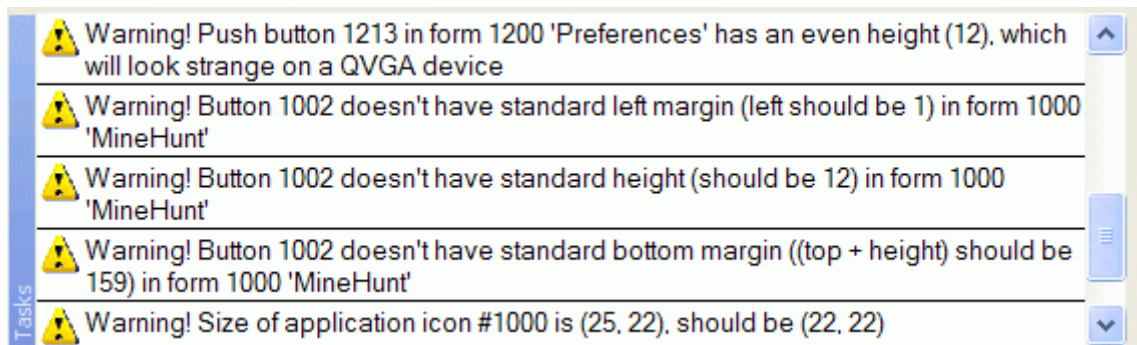
Figure 1.4 Files pane



Tasks Pane

The Tasks pane, shown in [Figure 1.5](#), displays messages about actions that you can take. Resource Editor generates these messages either automatically or in response to some operation you have performed. For example, if you use the **Validate Resources** function, Resource Editor displays information about your resources in the Tasks pane.

Figure 1.5 Tasks pane



Properties Pane

Most resources have two kinds of properties. The basic properties are general properties that all resource types have—for instance, the Resource ID property. The specific properties vary from one resource type to another. For instance, menu bars have a Hidden property, whereas other resource types do not. The Properties pane displays the basic or specific set of properties, depending how many times you click the resource in the Files pane.

Basic Properties

To see the values of basic properties for a given resource, single-click the resource in the Files pane. The basic properties of are listed in “[Basic Properties of Resources](#)” on page 15.

Table 1.1 Basic Properties of Resources

Attribute	Description
Symbolic Name	Logical name for resource type. Name of the XML element used to specify this resource type.
Unique Identifier	Three characteristics that uniquely identify a resource within an application—resource type, resource ID, locale (if any).
Resource ID	Integer ID assigned by Resource Editor.
Comment	Optional. Generates a COMMENT attribute for the current resource element. For most resource types, it is the name of the resource.
Comment Text	Optional. Generates a COMMENT_TEXT element. Used for longer comments, such as localization instructions.
Resource Locale	Optional. Generates a LOCALE attribute for the current resource element. Used by PalmRC to strip resources into locale-independent base resources and localized resources. Use a 4-character code that combines ISO 639 Language and ISO 3166 Country Codes, such as enUS.
Overlay Status	Optional. Generates an OVERLAY_STATUS attribute. Only explicit value is ADD. If attribute not specified, or specified as the empty string (""), the resource is considered a REPLACE overlay. Used to validate overlay resource files. A resource present in a specific locale, but not in the base locale, must have this attribute set to ADD.

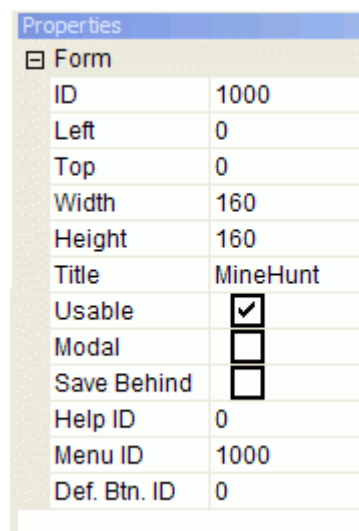
Specific Properties

To see the specific properties for a resource, double-click the resource in the Files pane. If you are in the Form Editor, click the visual representation of the form object whose properties you wish

to see. The Properties pane shown in [Figure 1.6](#) displays properties specific to a given resource type.

The first column in the properties pane contains the property name; the second column contains the property value. If a property value is editable, you can click on the Properties pane and change the displayed information.

Figure 1.6 Specific Properties



Properties	
Form	
ID	1000
Left	0
Top	0
Width	160
Height	160
Title	MineHunt
Usable	<input checked="" type="checkbox"/>
Modal	<input type="checkbox"/>
Save Behind	<input type="checkbox"/>
Help ID	0
Menu ID	1000
Def. Btn. ID	0

The properties are displayed in the font appropriate for the locale you have selected using the Keyboard button; see “[Keyboard Toolbar Button](#)” on page 17 for information.

You can change the width of the columns shown by selecting and dragging the vertical bar separating the attributes from their values.

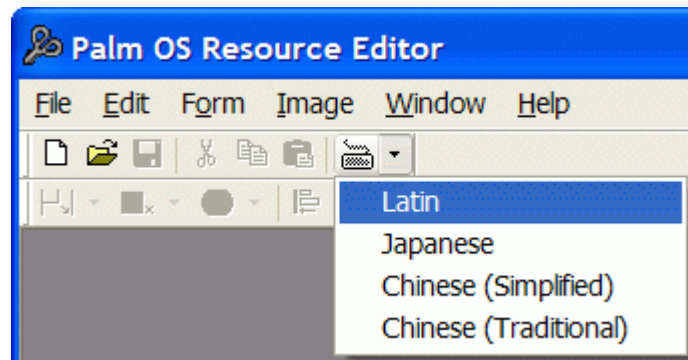
Properties of Form Objects

Form objects—such as buttons and check boxes—are treated differently from other resources. Form objects have only one set of properties. To view them, open the Form Editor for a particular form and click a form object. The properties appear in the Properties pane.

Keyboard Toolbar Button

The toolbar contains a Keyboard button, shown in [Figure 1.7](#), that lists locales and lets you select fonts appropriate to each locale. When clicked, the button displays a menu that lists all the recognized locale modules that are installed in the Resource Editor. Click a locale on the menu to change the font used to display text in the Properties pane, the String editor, and the String List editor.

Figure 1.7 Keyboard toolbar button



If you do not have the appropriate font installed for the selected locale, the Resource Editor uses the Arial font by default. [Table 1.2](#) lists the fonts and encodings that are associated with the various locales.

Table 1.2 Fonts and Encodings Associated with Locales

Palm OS Encoding	Microsoft Windows Encoding	Microsoft Windows Font
charEncodingPalmLatin	ANSI CHARSET	Arial Arial Unicode MS
charEncodingPalmSJIS	SHIFTJIS CHARSET	MS Mincho Arial Unicode MS
charEncodingPalmGB	GB2312 CHARSET	MingLiU Arial Unicode MS

Table 1.2 Fonts and Encodings Associated with Locales

Palm OS Encoding	Microsoft Windows Encoding	Microsoft Windows Font
charEncodingPalmBig5	CHINESEBIG5 CHARSET	MingLiU Arial Unicode MS
charEncodingPalmKR	HANGUL CHARSET	Gulim Batang Arial Unicode MS
default	ANSI CHARSET	Arial Arial Unicode MS

Resource Types

This section describes the Palm OS resource types, grouped according to how they are used in a Palm OS application.

- “[Forms and Form Objects](#)” on page 18
- “[Menus](#)” on page 19
- “[Character Strings, String Lists, and Category Names](#)” on page 19
- “[Alerts](#)” on page 20
- “[Icons and Bitmaps](#)” on page 20
- “[Fonts](#)” on page 20
- “[Sounds](#)” on page 20
- “[Data Types](#)” on page 20

Forms and Form Objects

- Form
- Form Title
- Buttons
 - Form Button

- Form Graphic Button
 - Form Push Button
 - Form Graphic Push Button
 - Form Repeating Button
 - Form Graphic Repeating Button
- Form Label
- Form Checkbox
- Form Field
- Form Graffiti State
- Form List
- Form Table
- Form Popup
- Form Popup Trigger
- Form Selector Trigger
- Form Slider
- Form Feedback Slider
- Form Scrollbar
- Window Constraints

Menus

- Menu Bar

Character Strings, String Lists, and Category Names

- Application Default Category
- Application Extended Preferences
- Application Icon Name
- Application Info Strings
- Application Launch Preferences
- Application Preferences

- Application Version
- String
- String List

Alerts

- Alert

Icons and Bitmaps

- Application Icon Bitmap
- Bitmap
- Color Table
- Form Bitmap

Fonts

- Font
- Extended Font
- TrueType Font

Sounds

- MIDI
- Wave Sound

Data Types

- Byte Integer List
- DWord Integer List
- Raw
- Soft Constant
- Word Integer List

Palm OS Protein Resource Types and 68K Resource Types

Palm OS Protein (ARM-native) resource types are not the same as native 68K resource types.

- The four-character resource type code that gets generated by the resource compiler, PalmRC, may be different.
- The binary resources generated by PalmRC may have endianness differences.

However, you can use the Palm OS® Cobalt resource tools to create resources for both 68K-native applications and Palm OS Protein applications. To do this:

1. Create the XML resource definition (XRD) file using Palm OS Resource Editor.
2. Specify the PalmRC command line option `-p` to specify which OS version you want to use:

68K

To build resources in the format used by Palm OS 4.1 and prior (68K resources).

Use this option when you are creating 68K applications.

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 Cobalt Palm OS Cobalt and later (Palm OS Protein resources).

Use this option when you are creating Palm OS Protein applications.

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

Palm OS Resource Editor Fundamentals

Palm OS Protein Resource Types and 68K Resource Types

Working with Application Resources

This chapter describes the editors for creating and modifying application preferences, application extended preferences, and application information strings.

Overview of Application Resources

To work with application resources, proceed as follows:

1. In the Navigator or C/C++ Projects tab of Palm OS Developer Suite, double-click the appropriate XRD file.
Palm OS Resource Editor will load and display the XRD file in the File pane.
2. In Palm OS Resource Editor, click the XRD file in the Files pane.
3. Create the application resources by selecting **Edit > New Resources** menu. Then select the appropriate resource type from the list.

The correct application resources editor will open.

For more information, read the following:

- [“Using the Application Preferences Editor”](#) on page 24.
- [“Using the Application Extended Preferences Editor”](#) on page 24.
- [“Using the Application Information Strings Editor”](#) on page 25.

Using the Application Preferences Editor

To create an application preferences resource, select **Edit > New Resources**, and then select App Preferences from the list.

If you are editing an existing resource, open the App Preferences folder in the Files pane. Double-click the application preferences resource to launch the editor.

To modify the values, see “[Understanding the App Prefs Editor](#)” on page 24.

Understanding the App Prefs Editor

The Application Preferences editor is a window with several text boxes displaying the default settings. You can modify those settings, if needed. They are:

- Priority—an integer specifying the priority for the application
- Stack Size—size of stack, in bytes, required by your application
- Minimum Heap Size—minimum size of heap, in bytes, required by your application.

Click the X in the corner of the editor window to dismiss it. Any new values you entered will be saved.

Using the Application Extended Preferences Editor

To create an application preferences resource, select **Edit > New Resources**, and then select App Extended Preferences from the list.

If you are editing an existing resource, open the App Extended Preferences folder in the Files pane. Double-click the application extended preferences resource to launch the editor.

To modify the value, see “[Understanding App Extended Prefs Editor](#)” on page 25.

Understanding App Extended Prefs Editor

The Application Extended Preferences editor is a window with setting. You can modify that setting by selecting the check box, if needed. The setting is:

- No Overlay—if checked, indicates that the application has no overlay.

Click the X in the corner of the editor window to dismiss it. Any new values you entered will be saved.

NOTE: The application extended preferences resource is used primarily in applications that target Palm OS Garnet or earlier. If you are targeting Palm OS Cobalt, use the Launch Preferences resource instead.

Using the Application Information Strings Editor

To create an application preferences resource, select **Edit > New Resources**, and then select App Info Strings from the list.

If you are editing an existing resource, open the App Info Strings folder in the Files pane. Double-click the application information strings resource to launch the editor.

To modify the values, see “[Understanding the App Info String Editor](#)” on page 25.

Understanding the App Info String Editor

The Application Information Strings editor is basically a String List editor with 16 sample-text strings already created. For general information on the String List editor, see “[Using the String List Editor](#)” on page 63.

Edit the default app info strings as follows:

1. Double-click on the sample text —such as, Sample Category 1—to get an insertion point.
2. Highlight the sample text using the arrow keys, and then press the Delete key.

Working with Application Resources

Using the Application Information Strings Editor

3. Enter a string to replace the sample text. This will usually be one of the categories that will ship with your application.
4. Drag and drop the strings to change the order in which the entries will appear. Normally, this is the list of categories that appear when the user selects the Categories pop-up trigger.
5. Click the X in the corner of the editor window to dismiss it. Any new values you entered will be saved.

Working with Forms

This chapter provides information about creating and editing form resources using Palm OS Resource Editor.

Working with Forms Overview

To work with forms, start Palm OS Resource Editor and open the resource file you want to edit, as described in “[Opening a Resource Description File in Resource Editor](#)” on page 8.

You can either create a new form or edit an existing form.

- To create a new form, see “[Creating a New Form](#)” on page 27.
- To edit an existing form, see “[Modifying an Existing Form](#)” on page 28.

Creating a New Form

To create a new form:

1. In the Files pane, select the resource description (XRD) file to which you want to add the form.
2. In the main menu bar, select **Edit > New Resource**. The New Resource dialog box appears.
3. In the New Resource dialog box, select **Form** and then click the **New** button.
4. The name and number of the new form appear in the Files pane, and the form is displayed in the Form Editor. For information on how to edit the form, see “[Modifying an Existing Form](#)” on page 28.

Modifying an Existing Form

To modify an existing form:

1. In the Files pane, select the form resource you want to modify. When you select a form resource, the form's properties are displayed in the Properties pane.
2. In the Files pane, double-click the form resource to display it in the Form Editor. The Form Editor appears. Now you can edit the form in the following ways:
 - To edit the form's title, use the Properties pane. See ["Editing a Form's Title"](#) on page 28 for details.
 - To edit popup lists in the form, see ["Editing Popup Objects"](#) on page 29.
 - To add items to the form, use the tools in the Toolbox. The Toolbox is located on the left side of the Form Editor. For detailed information about the tools, see ["Using the Toolbox"](#) on page 30.
 - To modify items in the form, you can drag and drop items in the Form Editor, or use the Form menu in the Resource Editor menu bar, or use the Form Editor toolbar. For detailed information about the Form Editor toolbar, see ["Using the Toolbar"](#) on page 29.
 - To add a form navigation resource for this form, select **Form >Create Navigation Resource**. For more information, see ["Creating a Form Navigation Resource"](#) on page 57.
 - To change form properties, go to the Properties pane and double-click the property you want to change.
3. When you are finished modifying the form, you can save it using the **Save** or **Save As** menu items.

Editing a Form's Title

To edit a form's title:

1. In the Form Editor, click the form's title.
2. In the Properties pane, double-click the **Title** attribute's value.

3. Edit the title.

Editing Popup Objects

Popup lists cannot be edited directly in the Resource Editor. Instead, you create a [Popup Trigger Object](#) and add items to a list object associated with that trigger.

1. Create a list object.
2. Create a popup trigger object.
3. Select the popup trigger object. In the Properties Pane for the popup trigger object, click the List ID value and enter the resource ID for the list object you created in the first step. This value links the popup trigger object to the list object.

Using the Toolbar

When the Form Editor is active, the Form Editor displays a toolbar near the top of the Palm OS Resource Editor window. The icons in the toolbar are alternatives for selecting certain menu items. From left to right, the items on the toolbar are:

- Change display density and size
- Change display depth (bits per pixel)
- Change locale
- Align Lefts
- Align Rights
- Align Tops
- Align Bottoms
- Center Vertically
- Center Horizontally
- Make Same Width
- Make Same Height
- Make Same Size

The first three items are always enabled. The remaining items are enabled only when you select objects in the form.

Using the Toolbox

The Toolbox appears on the left side of the Form Editor. The Toolbox displays a palette of resources you can add to the form. It also displays a pointer tool.

When you select a tool in the Toolbox, it becomes active. For some tools, you can use drag-and-drop to add content to the form. You can also double-click some tools to add content to the form.

The Pointer Tool

The pointer tool is used to select, move, and resize form objects. It can only be selected. It cannot be dragged and dropped, nor does double clicking perform any action.

How the pointer tool behaves depends on a number of factors, including:

- Hit location (whether a selected item, unselected item, or no item was hit)
- Modifier keys (whether the Shift key, Control key, both, or neither are pressed)
- Mouse action (whether the user just clicks, or clicks and drags out a selection)

Toolbox Pane


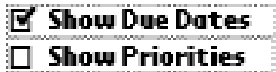


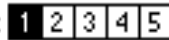
All tools but the pointer tool represent form objects that can be added to the form. Dragging a tool from the toolbox to the form creates a new form object of the dragged type at the location indicated by the drag and drop feedback. Double-clicking a tool creates a new form object of the clicked type at a default location in the form near the top left corner of the form.

When a tool is selected, the cursor changes to a cross-hair cursor while it is over the form, and you can sketch out a new form object of the indicated type at a desired location and with a desired size.

You can use the Toolbox pane to create the form objects listed in [Table 3.1](#). The following tables describe the properties of form

objects. To see the properties of a form object, open the Form Editor and click the visual representation of an object on the form



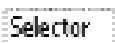


Table 3.1 Toolbox objects

Resource Name	Resource
Button Object	
Checkbox Object	
Feedback Slider Object	
Field Object	Look Up: Text.....
Form Bitmap Object	(container for Bitmap resource)
Gadget Object	(application defined)
Graffiti Shift Indicator Object	↑
Graphic Button Object	
Graphic Push Button Object	
Graphic Repeating Button Object	
Label Object	Set Date:
List Object	
Popup Trigger Object	▼ Unfiled
Push Button Object	: 

Working with Forms

Using the Toolbox

Table 3.1 Toolbox objects

Resource Name	Resource
Repeating Button Object	
Scroll Bar Object	
Selector Trigger Object	
Slider Object	
Table Object	

Button Object

The button object creates a command button. Most forms have a single row of command buttons at the bottom of the form.

Table 3.2 Button object attributes

Attribute	Description
Comment	Name of the form object. Must be a valid C name—for instance, it cannot have spaces.
ID	Integer ID assigned by Resource Editor.
Left	1 for left-most button on modeless form. 5 for left-most button on modal form.

Determine left origin for button n by:

$$\text{button}(n-1) \text{ left} + \text{button}(n-1) \text{ width} + 6$$

Table 3.2 Button object attributes (continued)

Attribute	Description
Top	147 or 160 – <i>height</i> – 1 for modeless forms <i>formHeight</i> – <i>buttonHeight</i> – 5 for modal forms
Width	36 or <i>label width</i> + 10
Height	12 or <i>font height</i> + 3 if not using standard font
Usable	Uncheck this box if you don't want the object to appear on the screen initially. Nonusable objects can programmatically be set to usable.
Enabled	Uncheck this box if you don't want the user to be able to interact with the object initially. Non-enabled objects can programmatically be set to enabled.
Text	Text displayed inside the button.
Font	Specifies the font used by the form object: STD_FONT, BOLD_FONT, LARGE_FONT SYMBOL_FONT, SYMBOL_11_FONT, SYMBOL_7_FONT, LED_FONT LARGE_BOLD_FONT

Table 3.2 Button object attributes (*continued*)

Attribute	Description
Left Anchor	Controls how the object resizes itself when its text label is changed. If checked, the left bound of the object is fixed; if unchecked, the right bound is fixed.
Frame	The <code>BUTTON_FRAME</code> element is used to specify the frame used by the form object. The element may be specified as one of the following enum values: <code>NO_BUTTON_FRAME,</code> <code>STANDARD_BUTTON_FRAME,</code> <code>BOLD_BUTTON_FRAME,</code> <code>RECTANGLE_BUTTON_FRAME</code>

Checkbox Object

The checkbox object creates a check box. Check boxes often appear in a group. If so, the group should be non-exclusive (so that the user can enable multiple check boxes) and should be aligned vertically or horizontally. If you want to make sure that only one of a series of options is selected, use push buttons or a list instead of check boxes.

Table 3.3 Check box resource attributes

Attribute	Description
Comment	Name of the form object. Must be a valid C name—for instance, it cannot have spaces.
ID	Integer ID assigned by Resource Editor.
Left	Form-relative position of left side of object.
Top	Form-relative position of top of object.
Width	Determined at runtime.
Height	12 pixels minimum Height determines tappable area Check box is always centered in chosen height

Table 3.3 Check box resource attributes (continued)

Attribute	Description
Usable	Uncheck this box if you don't want the object to appear on the screen initially. Nonusable objects can programmatically be set to usable.
Enabled	Uncheck this box if you don't want the user to be able to interact with the object initially. Non-enabled objects can programmatically be set to enabled.
Text	Text displayed to the right of the check box.
Font	Use Bold font for most check boxes.
Selected	Initial selection state of the check box. If this box is checked (the default), the check box is initially checked.
Group ID	Do not use.

Feedback Slider Object

The feedback slider object creates a feedback slider. The difference between a feedback slider and a regular slider is that feedback sliders send `ctlRepeatEvents` while the user drags the slider thumb. The regular slider sends a single `ctlSelectEvent` when the user is done dragging the thumb.

Table 3.4 Feedback slider object attributes

Attribute	Description
Comment	Name of the form object. Must be a valid C name—for instance, it cannot have spaces.
ID	Integer ID assigned by Resource Editor.
Left	Form-relative position of the left side of the object.

Table 3.4 Feedback slider object attributes (*continued*)

Attribute	Description
Top	Form-relative position of top of object.
Width	The default background bitmap looks best at these widths: 72, 93, 114, 135, or 156
Height	15 or height of thumb bitmap
Usable	Uncheck this box if you don't want the object to appear on the screen initially. Nonusable objects can programmatically be set to usable.
Enabled	Uncheck this box if you don't want the user to be able to interact with the object initially. Non-enabled objects can programmatically be set to enabled.
Value	Initial setting of the slider thumb.
Minimum	Minimum value the slider can represent.
Maximum	Maximum value the slider can represent.
Page Size	Amount by which the thumb moves if the user taps to its left or right.
Thumb Bitmap	ID of the bitmap resource or bitmap family resource used to create the thumb. Use 0 to use the default thumb bitmap.
Background Bitmap	ID of the bitmap resource or bitmap family resource used to create the background. Use 0 to use the default background bitmap.

Field Object

Use the field object to create an editable text field that is either a single-line long or multiple lines long. You can also use the field resource to create noneditable text that is displayed on the form. It is

easier to use noneditable text fields instead of labels if the text of the label changes dynamically.

Table 3.5 Field object attributes

Attribute	Description
Comment	Name of the form object. Must be a valid C name—for instance, it cannot have spaces.
ID	Integer ID assigned by Resource Editor.
Left	Form-relative position of left side of object.
Top	Form-relative position of top of object.
Width	Ideally wide enough to show maximum characters
Height	11 or <i>font height</i> + 2 for single-line fields For multi-line fields, a multiple of the line height.
Usable	Uncheck this box if you don't want the object to appear on the screen initially. Nonusable objects can programmatically be set to usable.
Editable	If this box is checked, the field is editable. Noneditable fields don't accept user input but can be changed programmatically.
Underlined	Check this box for editable fields. Uncheck for noneditable fields.
Single Line	Check this box for single-line fields so the field doesn't scroll horizontally and doesn't accept Return or Tab characters.
Dynamic Size	Uncheck this box if the Single Line box is checked. Check this box if you want the field to grow as the user enters more characters.

Table 3.5 Field object attributes (*continued*)

Attribute	Description
Justification	<p>Left-justify editable single-line fields. Right-justify single-line numeric fields or noneditable fields used as labels.</p> <p>Not applicable to multi-line fields.</p>
Max Chars	<p>Maximum number of bytes that the user can enter into an editable field. Note that the number of bytes is not the same as the number of characters when multi-byte characters (for example, Japanese) are being used.</p> <p>This attribute has no effect on noneditable fields.</p>
Auto Shift	<p>If checked, Graffiti or auto-shift rules are applied. This attribute should be enabled for most editable fields.</p>
Has Scroll Bar	<p>Check this box if you want to associate a scroll bar with a multi-line field.</p>
Numeric	<p>If checked, only the characters 0 through 9 and associated separators are allowed to be entered in the field. The associated separators are the thousands separator and the decimal character. The values of these two characters depend on the settings in the Formats preferences panel.</p> <p>Note that numeric fields do not allow plus signs.</p>
Max. Vis. Lines	<p>Maximum number of lines visible for a multi-line entry field.</p>

Form Bitmap Object

Use the form bitmap object to display a bitmap at a fixed location on a form. For example, an about dialog often displays the application icon along with the application name, version number, and other pertinent information. You can use a form bitmap resource to display the icon on this dialog.

Table 3.6 Form bitmap object attributes

Attribute	Description
Comment	Name of the form object. Must be a valid C name—for instance, it cannot have spaces.
ID	Integer ID assigned by Resource Editor.
Left	Form-relative position of left side of object.
Top	Form-relative position of top of object.
Bitmap ID	ID of a bitmap resource or bitmap family resource to be drawn in the specified location.
Usable	Uncheck this box if you don't want the object to appear on the screen initially. Nonusable objects can programmatically be set to usable.

Gadget Object

A gadget object creates an application-defined resource. You must write code to draw the user interface element and respond to events associated with that element.

Table 3.7 Gadget object attributes

Attribute	Description
Comment	Name of the form object. Must be a valid C name—for instance, it cannot have spaces.
ID	Integer ID assigned by Resource Editor.
Left	Form-relative position of left side of object.
Top	Form-relative position of top of object.
Width	Width in pixels.
Height	Height in pixels
Usable	Uncheck this box if you don't want the object to appear on the screen initially. Nonusable objects can programmatically be set to usable.
Visible	This value is set to TRUE when the gadget is drawn, and set to FALSE when the gadget is erased. <code>FrmHideObject()</code> sets this value to FALSE as well. You should set it explicitly in the gadget's callback function (if it has one) in response to a draw request.

Graffiti Shift Indicator Object

The Graffiti Shift Indicator (GSI) object should be displayed on all forms that contain an editable text field. After you place the GSI on a form, the system automatically displays the current shift status at that location.

Table 3.8 Graffiti Shift Indicator object attributes

Attribute	Description
Comment	Name of the form object. Must be a valid C name—for instance, it cannot have spaces.
Left	135 (modeless form) 142 (modal form)
Top	150 (modeless form) <i>form height</i> – 13 (modal form)

Graphic Button Object

The graphic button object creates a command button whose label is a bitmap rather than a text label.

Table 3.9 Graphic button object attributes

Attribute	Description
Comment	Name of the form object. Must be a valid C name—for instance, it cannot have spaces.
ID	Integer ID assigned by Resource Editor.
Left	1 for left-most button on modeless form. 5 for left-most button on modal form.
	Determine left origin for button <i>n</i> by: <i>button(n-1) left</i> + <i>button(n-1) width</i> + 6
Top	147 or 160 – <i>height</i> – 1 for modeless forms <i>formHeight</i> – <i>buttonHeight</i> – 5 for modal forms

Table 3.9 Graphic button object attributes (*continued*)

Attribute	Description
Width	36 or <i>label width</i> + 10
Height	<i>bitmap height</i> + 3
Usable	Uncheck this box if you don't want the object to appear on the screen initially. Nonusable objects can programmatically be set to usable.
Enabled	Uncheck this box if you don't want the user to be able to interact with the object initially. Non-enabled objects can programmatically be set to enabled.
Bitmap ID	Resource ID of the bitmap family that draws the graphic on this button. Assign a resource ID that matches the bitmap you want to use.
Selected ID	Resource ID of the bitmap family that draws the graphic when the button is selected. It's important to provide a different graphic for the selected bitmap on a color screen. The default is to invert the pixels, which draws funny on a color screen.
Left Anchor	Controls how the object resizes itself when its bitmap is changed. If checked, the left bound of the object is fixed; if unchecked, the right bound is fixed.
Frame	The <code>BUTTON_FRAME</code> element is used to specify the frame used by the form object. The element may be specified as one of the following enum values: <code>NO_BUTTON_FRAME,</code> <code>STANDARD_BUTTON_FRAME,</code> <code>BOLD_BUTTON_FRAME,</code> <code>RECTANGLE_BUTTON_FRAME</code>

Graphic Push Button Object

Use the graphic push button object to create a push button whose label is a bitmap rather than text. You usually create more than one push button at a time. Push buttons should be aligned horizontally or vertically and have no space between.

Table 3.10 Graphic push button object attributes

Attribute	Description
Comment	Name of the form object. Must be a valid C name—for instance, it cannot have spaces.
ID	ID assigned by Resource Editor.
Left	For a horizontal row of push buttons, the position of button(<i>n</i>) is: <i>button(n-1) left + button(n-1) width + 1</i>
Top	All push buttons in same group have same top (if horizontal row).
Width	<i>label width + 2</i> minimum All push buttons in same group have same width.
Height	<i>bitmap height + 3</i> All push buttons in same group have same height.
Usable	Uncheck this box if you don't want the object to appear on the screen initially. Nonusable objects can programmatically be set to usable.
Enabled	Uncheck this box if you don't want the user to be able to interact with the object initially. Non-enabled objects can programmatically be set to enabled.
Bitmap ID	Resource ID of the bitmap family that draws the graphic on this button. Assign a resource ID that matches the bitmap you want to use.

Table 3.10 Graphic push button object attributes (continued)

Attribute	Description
Selected ID	Resource ID of the bitmap family that draws the graphic when the button is selected. It's important to provide a different graphic for the selected bitmap on a color screen. The default is to invert the pixels, which draws funny on a color screen.
Group ID	Nonzero value between 1 and 65535 to identify the group. If 0, the push button is not assigned to a group.

Graphic Repeating Button Object

Use the graphic repeating button object to create a repeating button whose label is a bitmap rather than text. A repeating button can look like a command button or not. The difference between the two is that the repeating button sends events repeatedly while the user holds the pen down on the button. Command buttons wait until the user releases the pen and then send a single event.

Table 3.11 Graphic repeating button object attributes

Attribute	Description
Comment	Name of the form object. Must be a valid C name—for instance, it cannot have spaces.
ID	Integer ID assigned by Resource Editor.
Left	Form-relative position of left side of object.
Top	Form-relative position of top of object.
Width	Width of object in pixels
Height	Height of object in pixels
Usable	Uncheck this box if you don't want the object to appear on the screen initially. Nonusable objects can programmatically be set to usable.

Table 3.11 Graphic repeating button object attributes

Attribute	Description
Enabled	Uncheck this box if you don't want the user to be able to interact with the object initially. Non-enabled objects can programmatically be set to enabled.
Bitmap ID	Resource ID of the bitmap family that draws the graphic on this button. Assign a resource ID that matches the bitmap you want to use.
Selected ID	Resource ID of the bitmap family that draws the graphic when the button is selected. It's important to provide a different graphic for the selected bitmap on a color screen. The default is to invert the pixels, which draws funny on a color screen.
Left Anchor	Controls how the object resizes itself when its bitmap is changed. If checked, the left bound of the object is fixed; if unchecked, the right bound is fixed.
Frame	The BUTTON_FRAME element is used to specify the frame used by the form object. The element may be specified as one of the following enum values: NO_BUTTON_FRAME, STANDARD_BUTTON_FRAME, BOLD_BUTTON_FRAME, RECTANGLE_BUTTON_FRAME

Label Object

Use a label object to display noneditable text or labels on a form.

Table 3.12 Label object attributes

Attribute	Description
Comment	Name of the form object. Must be a valid C name—for instance, it cannot have spaces.
Label ID	Integer ID assigned by Resource Editor.
Left	Form-relative position of left side of object. Leave 3 pixels of white space between the label and its control.
Top	Form-relative position of top of object. Align labels with the top of the object they label.
Text	Text of the label. Typing a carriage return places a carriage return in the label text.
Font	Use Bold font for most labels.
Usable	Uncheck this box if you don't want the object to appear on the screen initially. Nonusable objects can programmatically be set to usable.

List Object

Use a list object to create a stand-alone list or the list portion of a pop-up list. (The pop-up trigger resource defines the trigger for the pop-up list.)

Table 3.13 List object attributes

Attribute	Description
Comment	Name of the form object. Must be a valid C name—for instance, it cannot have spaces.
ID	Integer ID assigned by Resource Editor.
Left	Form-relative position of left side of object. If creating pop-up list, use <i>trigger left</i> if trigger is left-aligned or <i>trigger left – list width</i> if trigger is right-aligned.
Top	Form-relative position of top of object. If creating pop-up list, top is often set to <i>trigger top</i> . Palm OS repositions the list if necessary.
Width	5 + width of longest item label 10 + width of longest item label to allow scrolling If creating a Categories pop-up list, the width is set at runtime. Prior to Palm OS 3.5, if the list is narrower than one of its items, the list items would draw outside of the list. Take care to ensure that this does not occur with your interface.
Font	Use Standard font for most lists.
Usable	If creating a pop-up list, uncheck this box.

Table 3.13 List object attributes (continued)

Attribute	Description
Num. Vis. Items	Number of items the list displays. 13 items for a full screen pop-up list 11 items for a full screen stand-alone list (modeless form) 10 items for a full screen stand-alone list (modal form) 0 for the Categories pop-up list (its size is determined at runtime) Set to the total number of items in your list if less than these numbers.
Num. Items	Number of items in list.

Popup Trigger Object

A pop-up trigger object creates the pop-up trigger portion of a pop-up list.

Table 3.14 Pop-up trigger object attributes

Attribute	Description
Comment	Name of the form object. Must be a valid C name—for instance, it cannot have spaces.
ID	Integer ID assigned by Resource Editor.
Left	Form-relative position of left side of button. Use 160 to align trigger with right edge of screen if it resizes to the left (Use this for Categories pop-up trigger.)
Top	Form-relative position of top of trigger.

Table 3.14 Pop-up trigger object attributes (continued)

Attribute	Description
Width	Determined at runtime. For editing purposes, you can set to several pixels wider than the list to be sure you can select the trigger separately from the list.
Height	12 <i>or font height + 3</i>
Usable	Uncheck this box if you don't want the object to appear on the screen initially. Nonusable objects can programmatically be set to usable.
Text	Usually leave this field blank and set the label programmatically.
Font	Use Standard font for most triggers.
Left Anchor	Check this box if the trigger should resize to the right when the label changes. If the trigger is on the right edge of the screen and should resize to the left, uncheck this box. Uncheck this box for the Categories pop-up trigger.
List ID	ID of the list resource to be popped up when the user taps the trigger.

Push Button Object

A push button object creates a push button. A group of push buttons represents a set of options where only one option can be selected at a time. You typically create several push buttons aligned horizontally or vertically.

Table 3.15 Push button object attributes

Attribute	Description
Comment	Name of the form object. Must be a valid C name—for instance, it cannot have spaces.
ID	ID assigned by Resource Editor.
Left	For a horizontal row of push buttons, the position of button(<i>n</i>) is: $button(n-1) \text{ left} + button(n-1) \text{ width} + 1$
Top	All push buttons in same group have same top (if horizontal row).
Width	$label \text{ width} + 2$ minimum All push buttons in same group have same width.
Height	$bitmap \text{ height} + 3$ All push buttons in same group have same height.
Usable	Uncheck this box if you don't want the object to appear on the screen initially. Nonusable objects can programmatically be set to usable.
Enabled	Uncheck this box if you don't want the user to be able to interact with the object initially. Non-enabled objects can programmatically be set to enabled.
Text	Specifies the text displayed inside the push button.

Table 3.15 Push button object attributes (continued)

Attribute	Description
Font	Specifies the font used by the form object: STD_FONT, BOLD_FONT, LARGE_FONT SYMBOL_FONT, SYMBOL_11_FONT, SYMBOL_7_FONT, LED_FONT LARGE_BOLD_FONT
Group ID	Nonzero value between 1 and 65535 to identify the group. If 0, the push button is not assigned to a group.

Repeating Button Object

The repeating button object creates a repeating button. Repeating buttons can look identical to command buttons or not. The difference between the two is that the repeating button sends events repeatedly while the user holds the pen down on the button. Command buttons wait until the user releases the pen and then send a single event.

The most common use of repeating buttons is to draw the scroll buttons in the lower-right portion of a form. They are also often used as increment/decrement arrows.

Table 3.16 Repeating button object attributes

Attribute	Description
Comment	Name of the form object. Must be a valid C name—for instance, it cannot have spaces.
ID	Integer ID assigned by Resource Editor.
Left	1 for left-most button on modeless form. 5 for left-most button on modal form.

Determine left origin for button n by:

$$\text{button}(n-1) \text{ left} + \text{button}(n-1) \text{ width} + 6$$

Table 3.16 Repeating button object attributes (*continued*)

Attribute	Description
Top	147 or 160 – <i>height</i> – 1 for modeless forms <i>formHeight</i> – <i>buttonHeight</i> – 5 for modal forms
Width	36 or <i>label width</i> + 10
Height	12 or <i>font height</i> + 3 if not using standard font
Usable	Uncheck this box if you don't want the object to appear on the screen initially. Nonusable objects can programmatically be set to usable.
Enabled	Uncheck this box if you don't want the user to be able to interact with the object initially. Non-enabled objects can programmatically be set to enabled.
Text	Specifies the text that appears inside the repeating button. Use "\x01" to specify the scroll up button. Use "\x02" to specify the scroll down button.
Font	Specifies the font used by the form object: STD_FONT, BOLD_FONT, LARGE_FONT SYMBOL_FONT, SYMBOL_11_FONT, SYMBOL_7_FONT, LED_FONT LARGE_BOLD_FONT

Table 3.16 Repeating button object attributes (continued)

Attribute	Description
Left Anchor	Controls how the object resizes itself. TRUE - the object's left bound is fixed. FALSE - the object's right bound is fixed.
Frame	The BUTTON_FRAME element is used to specify the frame used by the form object. The element may be specified as one of the following enum values: NO_BUTTON_FRAME, STANDARD_BUTTON_FRAME, BOLD_BUTTON_FRAME, RECTANGLE_BUTTON_FRAME

Scroll Bar Object

Use the scroll bar object to create a scroll bar.

Table 3.17 Scroll bar object attributes

Attribute	Description
Comment	Name of the form object. Must be a valid C name—for instance, it cannot have spaces.
ID	Integer ID assigned by Resource Editor.
Left	Specifies, in standard coordinates, the left-most edge of the form object.
Top	Specifies, in standard coordinates, the top-most edge of the form object.
Width	Specifies, in standard coordinates, the width of the form object.
Height	Specifies, in standard coordinates, the height of the form object.

Table 3.17 Scroll bar object attributes (continued)

Attribute	Description
Usable	Uncheck this box if you don't want the object to appear on the screen initially. Nonusable objects can programmatically be set to usable.
Value	Initial setting for the scroll bar.
Text	Specifies the text that appears inside the repeating button. Use "\x01" to specify the scroll up button. Use "\x02" to specify the scroll down button.
Minimum	The minimum value the scroll bar can represent.
Maximum	The maximum value the scroll bar can represent.
Page Size	Number of lines to scroll at one time. This value is often not set until run time. It should be one less than the number of lines that can be displayed at one time to provide context. That is, if the field can display ten lines of text, the page jump value should be nine.

Selector Trigger Object

Use the selector trigger object to create a selector trigger.

Table 3.18 Selector trigger object attributes

Attribute	Description
Comment	Name of the form object. Must be a valid C name—for instance, it cannot have spaces.
ID	Integer ID assigned by Resource Editor.
Left	Form-relative position of left side of button.
Top	Form-relative position of top of trigger.

Table 3.18 Selector trigger object attributes (continued)

Attribute	Description
Width	Determined at runtime. For editing purposes, you can set to several pixels wider than the list to be sure you can select the trigger separately from the list.
Height	12 or <i>font height + 3</i>
Usable	Uncheck this box if you don't want the object to appear on the screen initially. Nonusable objects can programmatically be set to usable.
Text	Specifies the text for the selector trigger on the form. Usually leave this field blank and set the label programmatically.
Font	Use Standard font for most triggers.
Left Anchor	Check this box if the trigger should resize to the right when the text changes. If the trigger is on the right edge of the screen and should resize to the left, uncheck this box.

Slider Object

The slider object creates a slider.

Table 3.19 Slider object attributes

Attribute	Description
Comment	Name of the form object. Must be a valid C name—for instance, it cannot have spaces.
ID	Integer ID assigned by Resource Editor.
Left	Form-relative position of the left side of the object.

Table 3.19 Slider object attributes (*continued*)

Attribute	Description
Top	Form-relative position of top of object.
Width	The default background bitmap looks best at these widths: 72, 93, 114, 135, or 156
Height	15 or height of thumb bitmap
Usable	Uncheck this box if you don't want the object to appear on the screen initially. Nonusable objects can programmatically be set to usable.
Enabled	Uncheck this box if you don't want the user to be able to interact with the object initially. Non-enabled objects can programmatically be set to enabled.
Value	Initial setting for the slider.
Minimum	Minimum value the slider can represent.
Maximum	Maximum value the slider can represent.
Page Size	Amount by which the thumb moves if the user taps to its left or right.
Thumb Bitmap	ID of the bitmap resource or bitmap family resource used to create the thumb. Use 0 to use the default thumb bitmap.
Background Bitmap	ID of the bitmap resource or bitmap family resource used to create the background. Use 0 to use the default background bitmap.

Table Object

Use the table object to create a table.

Creating a Form Navigation Resource

A form navigation resource is used in one-handed navigation. It determines how the navigation highlight jumps from object to object when the one-handed navigation button is pressed. This resource defines a tab order for the right and left directions of the one-handed navigation button. To support the up and down directions, this resource specifies, for each object, which object lies directly above and which object lies directly below.

You can create a navigation resource for each form.

When creating a form navigation resource, you have two choices.

- You can code the resource by hand, as described in “[Hand-Coding the Form Navigation Resource](#)” on page 57.
- or
- You can have Palm OS Resource Editor generate a default navigation resource for you. For more, see “[Generating the Form Navigation Resource](#)” on page 57.

Hand-Coding the Form Navigation Resource

A form navigation resource is defined in XML as a `FORM_NAVIGATION_RESOURCE` element. You can code the XML definition manually in the XML editor of Palm OS Resource Editor, if you choose. Make sure the form navigation resource has the same resource ID as its associated form. For reference information on the XML definition, consult the “Form Navigation Resource” section in *Palm OS Resource File Formats*.

To have Palm OS Resource Editor create a form navigation resource for you, see “[Generating the Form Navigation Resource](#)” on page 57.

Generating the Form Navigation Resource

You can have Palm OS Resource Editor automatically generate a form navigation resource for your form. The resulting XML will define a navigation resource for the form. The right-left navigation is determined by the object IDs, sorted in ascending order. The up-down navigation elements —`ABOVE_OBJECT_ID` and

Working with Forms

Creating a Form Navigation Resource

BELOW_OBJECT_ID—are initialized to 0. In order to enable up-down navigation, you must specify, for each object, the object IDs of the above and below objects. You can fine-tune the default navigation resource in other ways, if desired.

To generate a a form navigation resource:

1. In the Files pane, open the Form folder and select the form for which you wish to create a navigation resource.
2. From the right-mouse popup menu, select **Create Navigation Resource**. The XML Editor appears, displaying the new navigation resource.
3. Edit the XML definition, if required.

Working with Menus

This chapter introduces Menu Bar resources, used to create menu bars and menus for Palm OS® software applications.

Working with Menus Overview

To work with menus, start Palm OS Resource Editor and open the resource file you want to edit, as described in “[Opening a Resource Description File in Resource Editor](#)” on page 8.

You can either create a new menu or menu bar, or edit an existing menu or menu bar.

- To create a new menu bar, see “[Creating a New Menu Bar](#)” on page 59.
- To create a new menu item, see “[Creating a New Menu Item](#)” on page 60.
- To edit an existing menu bar, see “[Modifying an Existing Menu](#)” on page 60.

Creating a New Menu Bar

To create a new menu:

1. In the Files pane, select the resource description (XRD) file to which you want to add the menu bar.
2. In the main menu bar, select **Edit > New Resource**. The New Resource dialog box appears.
3. In the New Resource dialog box, select **Menu Bar** and then click the **New** button.
4. The name and number of the new menu bar appear in the Files pane, and the menu bar is displayed in the Menu Editor. For information on how to edit the menu bar, see “[Using the Menu Editor](#)” on page 60.

Creating a New Menu Item

To add a new menu item, see the instructions under “[Using the Menu Editor](#)” on page 60.

Modifying an Existing Menu

To modify an existing menu:

1. In the Files pane, select the menu resource you want to modify. When you select a menu resource, the menu’s properties are displayed in the Properties pane.
2. In the Files pane, double-click the menu resource to display it in the Menu Editor. The Menu Editor appears. You can edit some menu items in the Menu Editor (see “[Using the Menu Editor](#)” on page 60 for details) and some menu items in the Properties pane.

Using the Menu Editor

The Menu Editor consists of a Windows tree view control that displays the menu bar. The top level of the tree represents the top-level menus, and the second level of the tree represents the menu items in the menu.

When you select a menu or menu item, the properties for the menu or menu item appear in the Properties pane.

In the Menu Editor, you can add and remove menus and menu items by right-clicking them, or you can use the Delete and Insert keys on your keyboard.

To change an item’s order in the menu, right-click the item to display a context menu. From the context menu, select the **Move Up** or **Move Down** commands. You can also use the up arrow and down arrow keys on your keyboard.

The **Cut**, **Copy**, **Paste**, **Clear**, and **Duplicate** menu items are all functional.

Working with Strings

This chapter introduces character string resources, which can be used for help text and default values for text fields.

Working with Strings Overview

To work with strings, start Palm OS Resource Editor and open the resource file you want to edit, as described in “[Opening a Resource Description File in Resource Editor](#)” on page 8.

You can either create a new string or list of strings, or edit an existing string or list of strings.

- To create a new string or list of strings, see “[Creating New Strings](#)” on page 61.
- To edit an existing string, see “[Modifying an Existing String](#)” on page 62.
- To edit an existing list of strings, see “[Modifying an Existing List of Strings](#)” on page 62.

Creating New Strings

To create a new menu:

1. In the Files pane, select the resource description (XRD) file to which you want to add the menu bar.
2. In the main menu bar, select **Edit > New Resource**. The New Resource dialog box appears.
3. In the New Resource dialog box, select **String** or **String List** and then click the **New** button.
4. The name and number of the new string or string list appear in the Files pane, and the string or string list is displayed in the String Editor or String List Editor. For information on how to edit the strings, see “[Editing Strings and String Lists](#)” on page 62.

Modifying an Existing String

To modify an existing string:

1. In the Files pane, select the string resource you want to modify. When you select a string resource, the string's properties are displayed in the Properties pane.
2. In the Files pane, double-click the string resource to display it in the String Editor. The String Editor appears. For information on how to use the String Editor, see "[Editing Strings and String Lists](#)" on page 62.

Modifying an Existing List of Strings

To modify an existing string list:

1. In the Files pane, select the string list resource you want to modify. When you select a string list resource, the string list's properties are displayed in the Properties pane.
2. In the Files pane, double-click the string list resource to display it in the String List Editor. The String List Editor appears. For information on how to use the String List Editor, see "[Editing Strings and String Lists](#)" on page 62.

Editing Strings and String Lists

You edit strings in the String Editor, and you edit String Lists in the String List Editor.

When you select a string or string list, the properties of the object appear in the Properties pane. Double-click the string or string list to display it in an editor window.

Using the String Editor

The String Editor is a window where you can string of text characters. In the String Editor window, you can type a Unicode character string using the operating system's UI character set.

You can type any Unicode character that the character set being used allows. For example, if the character set of the OS is English, then the characters are input in English. If the character set is Chinese, then the Windows OS uses a Chinese Input Method Editor

to create the characters and add them to the String Editor window as Chinese characters. Note that the String Editor does not validate the individual characters of the string to verify if the character is valid in the target text encoding of the Palm OS application. Validation of the string contents is performed at resource compilation time.

A string may contain characters that cannot be represented in the current character set of the operating system UI, since the character sets on the Palm OS do not necessarily match the character set of the OS on which the Resource Editor is running. In such cases, the character is represented within the string editor using the character's hex values in the form of `'\x####'`.

Locale Information

The String Editor supports the Keyboard toolbar button by setting the font according to the locale selected. For details on the Keyboard toolbar item, see "[Keyboard Toolbar Button](#)" on page 17.

Note that there is no strict relation between the locale property for a string resource and the actual character set used to construct the string value. The locale property and the actual character set are independent and could be different. For example, the locale value could be `enUS` but the string could be represented by Simplified Chinese Unicode characters.

Character Map

The String Editor can be used in conjunction with the Character Map. The Character Map is a Resource Editor tool that displays the characters in the various standard fonts and their variations according to the Palm OS version. You can cut and paste characters from the Character Map into the String Editor window.

To display the Character Map, go to the Resource Editor main toolbar and select **Window > Character Map**.

Using the String List Editor

The String List Editor is a window where you can edit a list of strings; that is, you can edit a string list resource. A string list

Working with Strings

Editing Strings and String Lists

resource consists of an optional prefix string and a list of zero or more individual character strings.

The String List Editor works much like the String Editor. See “[Using the String Editor](#)” on page 62 for details.

In the String List Editor, you can select and edit each string in the list. To add a new string to the list, click the Insert key on your keyboard. To delete a string, select the string and click the Delete key on your keyboard. To move a string, select it with the mouse and drag and drop the string in the desired location.

The String List editor follows the same rules regarding Unicode character and Palm OS character set handling as described in “[Using the String Editor](#)” on page 62.

Working with Alert Dialogs

This chapter introduces Alert dialog resources, used to communicate important messages to users.

Working with Alerts Overview

To work with Alert dialogs, start Palm OS Resource Editor and open the resource file you want to edit, as described in “[Opening a Resource Description File in Resource Editor](#)” on page 8.

You can either create a new alert dialog, or edit an existing alert dialog.

- To create a new alert dialog, see “[Creating a New Alert Dialog](#)” on page 65.
- To modify an existing alert, see “[Modifying an Existing Alert Dialog](#)” on page 66.

Creating a New Alert Dialog

To create a new alert dialog:

1. In the Files pane, select the resource description (XRD) file to which you want to add the alert dialog.
2. In the main menu bar, select **Edit > New Resource**. The New Resource dialog box appears.
3. In the New Resource dialog box, select **Alert** and then click the **New** button.
4. The name and number of the new alert dialog appear in the Files pane, and the alert dialog is displayed in the Alert Editor. For information on how to edit the alert, see “[Modifying an Existing Alert Dialog](#)” on page 66.

Modifying an Existing Alert Dialog

To modify an alert dialog, edit the alert dialog's properties in the Properties pane. To display an alert dialog in the Properties pane:

1. In the Files pane, select the alert resource you want to display. When you select an alert resource, the alert dialog's properties are displayed in the Properties pane.
2. In the Files pane, double-click the menu resource to display it in the Alert Editor. The Alert Editor appears.

Adding a Button to an Alert Dialog

To add a button to an alert dialog, use the **Edit > New Alert Button** menu item.

To change the text of a button:

1. Select the alert resource in the Files pane
2. Click on the Alert Editor.
3. Double-click the button text in the Properties pane.

Deleting a Button from an Alert Dialog

To delete a button, select the button in the Alert Editor and select **Edit > Clear**.

NOTE: An Alert dialog is required to have a least one button; you cannot delete all of the buttons from an Alert dialog.

Working with Icons, Bitmaps and Other Images

This chapter introduces the Bitmap Editor, used to create and modify bitmaps, icons, bitmap families, and icon families within a variety of resources.

Overview of Icons, Bitmaps, and Other Images

To work with icons and bitmaps, start Palm OS Resource Editor and open the resource file you want to edit, as described in “[Opening a Resource Description File in Resource Editor](#)” on page 8.

You can either create a new bitmap, or edit an existing bitmap.

- To create a new bitmap, see “[Creating New Bitmaps](#)” on page 67.
- To modify an existing bitmap, see “[Modifying an Existing Bitmap](#)” on page 68.

To work with other raster images in your Palm OS application, see “[Using the Bitmap Editor](#)” on page 68.

Creating New Bitmaps

To create a new bitmap:

1. In the Files pane, select the resource description (XRD) file to which you want to add the bitmap.
2. In the main menu bar, select **Edit > New Resource**. The New Resource dialog box appears.
3. In the New Resource dialog box, select **Bitmap** and then click the **New** button.

4. The name and number of the new bitmap appear in the Files pane, and the bitmap is displayed in the Bitmap Editor. For information on how to edit the bitmap, see “[Modifying an Existing Bitmap](#)” on page 68.

Modifying an Existing Bitmap

To modify an existing bitmap:

1. In the Files pane, select the bitmap resource you want to modify. When you select a bitmap resource, the bitmap’s properties are displayed in the Properties pane.
2. In the Files pane, double-click the bitmap resource to display it in the Bitmap Editor. The Bitmap Editor appears. For information on how to use the Bitmap Editor, see “[Using the Bitmap Editor](#)” on page 68.

Using the Bitmap Editor

In the Bitmap Editor you can edit Palm OS bitmaps. A Palm OS bitmap is actually a family of bitmaps. A bitmap image family is a logical bitmap which is composed of a set of individual bitmap images at different bit-depths and densities. The Bitmap Editor provides a bitmap image family editor and a bitmap image editor.

The left side of the Bitmap Editor window displays the image family properties and all available bitmap images in the family. On the right side of the Bitmap Editor you can edit any bitmap image in the image family, and you can set element-specific properties for the bitmap. The Bitmap Editor window also contains a color palette.

Setting Image Family Properties

To change the image family’s height and width, click the **Set** button in the Bitmap Editor. This displays a New Bitmap Size dialog where you can specify the new dimensions of the bitmaps in this image family. The width and height are expressed as values for the single-density image family members. Family members with other densities are scaled according to their density.

You can also specify whether or not to scale the images to the new size. If not, the image will be truncated in any dimension that is made smaller, and will have white filling any newly exposed areas.

Viewing Image Family Elements

The Bitmap Editor shows all the bitmap elements that are in the image family. Bitmap elements are uniquely identified by density and bit-depth.

The densities currently defined are 1X, 1.5X, and 2X. The bit-depths currently defined are 1-bit (black & white), 2-bit (4 grays), 4-bit (16 grays), 8-bit (Palm OS palette of 256 colors), and 16-bit (5-6-5 RGB).

The bitmaps in the image family are displayed as a list of thumbnail images, listed in order of increasing depth and then density, with each element labeled clearly. If there are more thumbnails than fit in the view area, a vertical scrollbar appears.

Thumbnails are shown in normalized scale. That is, the 1X and 2X thumbnails are shown at the same size, not with the 2X image being twice as large as the 1X. This way the “logical” value is being shown, making it easy to compare the actual device appearance of the bitmaps independent of the nominal density.

Selecting a thumbnail draws a selection rectangle around it and makes it active in the Bitmap Editor.

Adding and Deleting Image Family Elements

To add image elements to the image family, click the **Add** button in the Bitmap Editor. This displays a New Bitmap dialog listing all the possible image element types. You can select one or more image element types to create them. Element types that already exist in the bitmap family are indicated by being checked and disabled.

Working with Icons, Bitmaps and Other Images

Using the Bitmap Editor

IMPORTANT: Resource Editor allows you to use the same external image file in more than one resource file. However, if you modify a property of the external image, you may inadvertently overwrite an existing bitmap used by another resource or resource file. As a result, you should generally make a copy of an external image file rather than reusing it in more than one resource file.

To delete image elements, you can use the mouse to select the elements you want to delete and then click the **Delete** button in the Bitmap Editor. Note that only the image family element from the resource is deleted; no external bitmap files are deleted.

You cannot delete all the image elements. If you try to delete the last element in a family, the Bitmap Editor will not allow it unless you first create a new element. This is necessary, as bitmap resources must contain at least one image element.

Importing Images

To define image elements, you select them and draw them in the Bitmap Editor, or else you import an image from a pre-existing file on disk. To import an image, click the **Import** button in the Bitmap Editor.

Note that saving a bitmap resource does not necessarily result in the edited imported image being saved in the file from which it came.

If the import image is a different size than the bitmap family member into which it is being stored, the Import Bitmap dialog appears, asking the user to resolve the difference. The Import Bitmap dialog allows the user to select one of four options:

- **Scale new image:** The new image is scaled from its original size to the size of the other bitmap family members.
- **Truncate new image:** The new image is truncated to fit the bounds implied by the other bitmap family members. If the new image is smaller than the existing members, it is enlarged and the newly exposed areas are filled with white.

- Scale existing images to new image size: The existing bitmap family members are made to conform to the new image's size, scaling them as appropriate.
- Truncate existing images to new image size: The existing bitmap family members are made to conform to the new image's size, truncating or expanding them as appropriate.

Using the Color Palette

The Bitmap Editor contains a Color Palette. The Color Palette changes based on the color format of the image being edited. For palette images (1-bit, 2-bit, 4-bit, and 8-bit) it displays a swatch of each of the colors available. For true color images (16-bit RGB) it displays a palette of 644 evenly distributed colors.

NOTE: If the XRD source file specifies that a bitmap uses a color table, then Resource Editor will attempt to use that color table. However, if the color table is missing, the color palette will be empty. You may see this problem if you have an existing application where the `hasColorTable` flag is set, but no color table exists.

In the top left corner of the Color Palette is a control that shows the currently selected foreground and background colors. Clicking on any of the available color swatches or on the color picker sets the foreground color. Right-clicking on any of the available color swatches or on the color picker sets the background color.

Using the Bitmap Editor Toolbar

The Bitmap Editor has a toolbar containing the following items:

- Rectangle Selection tool
- Color Selection tool
- Fill tool
- Magnification tool
- Pencil tool
- Brush tool
- Air Brush tool

Working with Icons, Bitmaps and Other Images

Working with Other Formats—Graphic Families

- Line tool
- Rectangle tool
- Outlined Rectangle tool
- Filled Rectangle tool
- Rounded Rectangle tool
- Outlined Rounded Rectangle tool
- Filled Rounded Rectangle tool
- Ellipse tool
- Outlined Ellipse tool
- Filled Ellipse tool

When you click a tool in the toolbar to select it, the cursor changes to an appropriate cursor for the action that the tool performs.

Working with Other Formats—Graphic Families

Bitmaps and icons are not the only image formats available for use in Palm OS applications. Currently, you can add families of PNG images to your application in Palm OS Developer Suite.

1. Create and edit PNG images using an appropriate graphics editor (not part of Palm OS Developer Suite).
2. In Palm OS Developer Suite, create a graphic family resource. For more information, see “[Creating a New Graphic Family Resource](#)” on page 72.
3. Import the individual images into the graphic family resource copying the data or by reference an external file. For more information, see

Creating a New Graphic Family Resource

To create a new graphic family resource:

1. In the Files pane, select the resource description (XRD) file to which you want to add the bitmap.
2. In the main menu bar, select **Edit > New Resource**. The New Resource dialog box appears.

3. In the New Resource dialog box, select **Graphic Family** and then click the **New** button.
4. The number of the new graphic family resource appears in the Files pane, and a skeletal XML definition of a graphic family appears in the XML editor.

Complete the XML description for the graphic family as a whole. Then add an XML description for each of the images in the family. Follow the XML format described under “Graphic Family Resource” in *Palm OS Resource File Formats*.

NOTE: You add the images to the family by copying the binary data or by referencing an external file. For more information, see [“Importing Images to a Graphic Family”](#) on page 73.

Importing Images to a Graphic Family

You can import images to a graphic family by copying the data into the XRD file or referencing an external file.

- Use the `GRAPHIC_FAMILY_DATA` element to copy the binary data of the image directly into the XRD file. Data must be in hexadecimal format.
- Use the `GRAPHIC_FAMILY_EXTERNAL` element to reference an external image file.

For a complete description of the XML format, see the “Graphic Family Resource” section in *Palm OS Resource File Formats*.

Working with Icons, Bitmaps and Other Images

Working with Other Formats—Graphic Families

Working with Other Resource Types

The previous chapters described how to work with resource types for which a default editor (such as a Form Editor or Alert Editor) is provided. This chapter describes how to work with resources for which no default editor is provided.

Overview of Other Resource Types and How to Work with Them

To work with resource types for which no default editor is provided you can use either the XML Editor or the Hex Viewer.

- The XML Editor shows the XML resource description for the resource. This editor is to be used for resources for which no custom resource editor is provided. For information on how to select this editor, see [“Edit Resource”](#) on page 84.
- The Hex Viewer shows the binary form of the resource. This editor is to be used for resources for which no custom resource editor is provided. For information on how to select this editor, see [“Edit Resource”](#) on page 84.

Using the XML Editor

To work with resources in the XML Editor, start Palm OS Resource Editor and open the resource file you want to edit, as described in [“Opening a Resource Description File in Resource Editor”](#) on page 8.

You can either create a new resource, or edit an existing resource.

- To create a new resource, see [“Creating a New Resource”](#) on page 76.

Working with Other Resource Types

Working with the Hex Viewer

- To edit an existing resource, see “[Modifying an Existing Resource](#)” on page 76.

Creating a New Resource

To create a new resource:

1. In the Files pane, select the resource description (XRD) file to which you want to add the sound resource.
2. In the main menu bar, select **Edit > New Resource**. The New Resource dialog box appears.
3. In the New Resource dialog box, select the resource type you want to create, and then click the **New** button.
4. The name and number of the new resource appear in the Files pane, and the resource is displayed in an XML Editor. For information on how to edit the resource, see “[Modifying an Existing Resource](#)” on page 76.

Modifying an Existing Resource

You can edit some resource items in the XML Editor window, and some resource items in the Properties pane.

To modify a resource:

1. In the Files pane, select the resource you want to modify. When you select a resource, the resource’s properties are displayed in the Properties pane.
2. In the Files pane, double-click the resource to display it in an XML Editor window. The XML Editor displays the XML for the resource. You can edit the XML in the XML Editor. For example, for a WAV sound the XML Editor might display:

```
<WAVE_SOUND_RESOURCE RESOURCE_ID="1000">  
  <WAVE_FILE> "WAVE_1000.wav" </WAVE_FILE>  
</WAVE_SOUND_RESOURCE>
```

Working with the Hex Viewer

To view a resource in the Hex Viewer:

1. In the Files pane, select the resource.

2. In the Resource Editor main toolbar, select **Edit > Edit Resource > Hex Viewer**. The resource is displayed in binary format in the Hex Viewer window.

Working with Other Resource Types

Working with the Hex Viewer

Menu Reference

This chapter provides a description of each Palm OS Resource Editor menu item.

Menu Reference Overview

The Palm OS Resource Editor menus include:

- [File](#)
- [Edit](#)
- [Form](#)
- [Image](#)
- [Window](#)
- [Help](#)

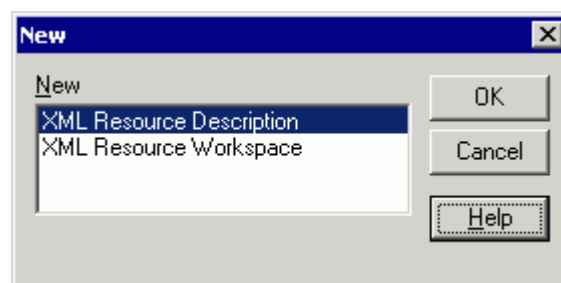
File

Use the **File** menu to manage the resource files in your workspace.

New

Opens a dialog box, shown in [Figure 9.1](#), so that you can create an XML Resource Description or an XML Resource Workspace.

Figure 9.1 New dialog box



Menu Reference

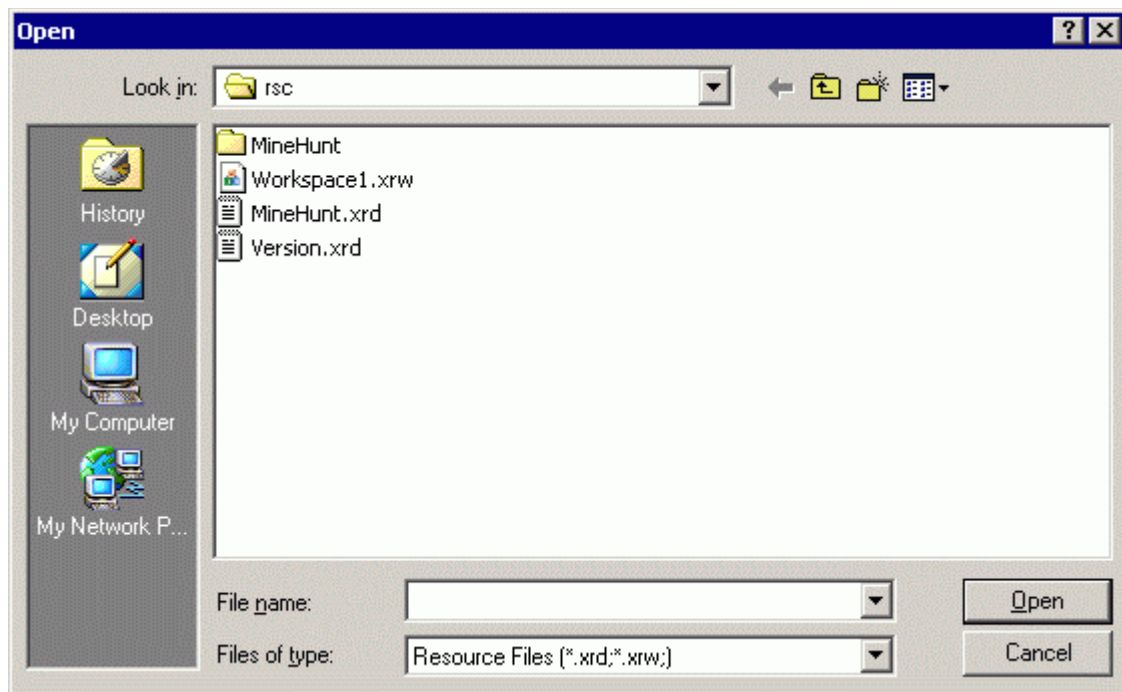
File

You can open only one workspace at a time. If you try to create a new workspace while another workspace is open, you are asked whether you want to save the existing workspace, if you have made changes to it. The current workspace is closed before the new workspace is created.

Open

Opens a dialog box, shown in [Figure 9.2](#), so that you can select an existing XML resource description (XRD file) or XML resource workspace (XRW file).

Figure 9.2 Open dialog box



You can open only one workspace at a time. If you try to open an existing workspace while another workspace is open, you are asked whether you want to save the open workspace, if you have made changes to it. The open workspace is closed before the new workspace is opened.

Close

Closes all the files associated with the currently open workspace. If you have made changes to the workspace, Resource Editor asks whether you want to save the workspace before closing it.

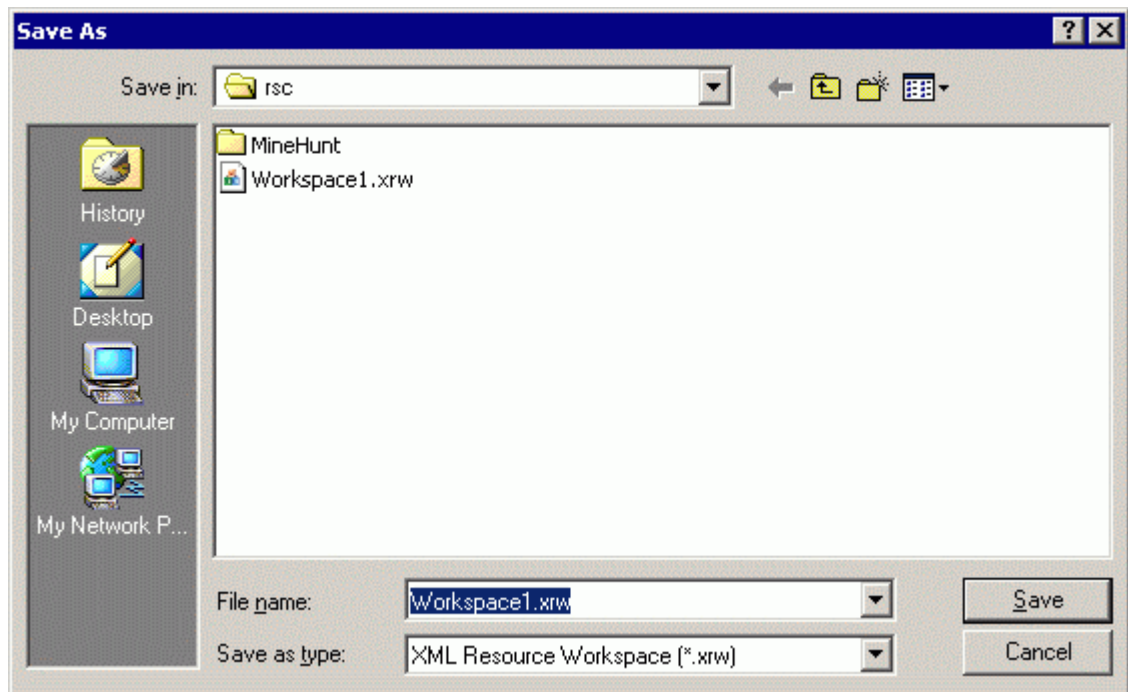
Save

Saves the currently-selected resource file or workspace.

Save As

Opens a dialog box, shown in [Figure 9.3](#), that allows you to save the currently selected XML resource description (XRD file) or XML resource workspace (XRW file).

Figure 9.3 Save As dialog box



Save All

Saves any XML resource description (XRD file) or XML resource workspace (XRW file) that has been changed during this editing session.

Menu Reference

Edit

Recent Documents

Lists the names of the most recently opened XML resource descriptions (XRD files).

Recent Workspaces

Lists the names of the most recently opened XML resource workspaces (XRW files).

Exit

Closes all the files associated with the currently open workspace, and exits Palm OS Resource Editor.

Edit

Use the **Edit** menu to access editing functions in the currently open editor window.

Undo

Undoes the most recently performed action.

Cut

Removes the current selection and puts it on the system clipboard.

Copy

Copies the current selection to the system clipboard.

Paste

Pastes the contents of the system clipboard into the active window.

Clear

Clears the current selection in the active window.

Find

Opens a **Find** dialog box so that you can search for text in the active window.

Find Next

Resumes the prior search without displaying the **Find** dialog box.

Replace

Opens a **Replace** dialog box so that you can search for a text string and replace the text string with a new text string.

Select All

Selects all items in the active window.

Duplicate

Makes a copy of the currently selected item, pasting it to a default location.

Validate Resources

Runs a validation check on the currently selected resource or resource description file. This means checking the resource or resource description file for UI and OS compliance. Noncompliant items will be listed in the Tasks pane.

You can either select an item in the Tasks pane to edit or delete the item, or you can double-click on an item to open it in the default editor.

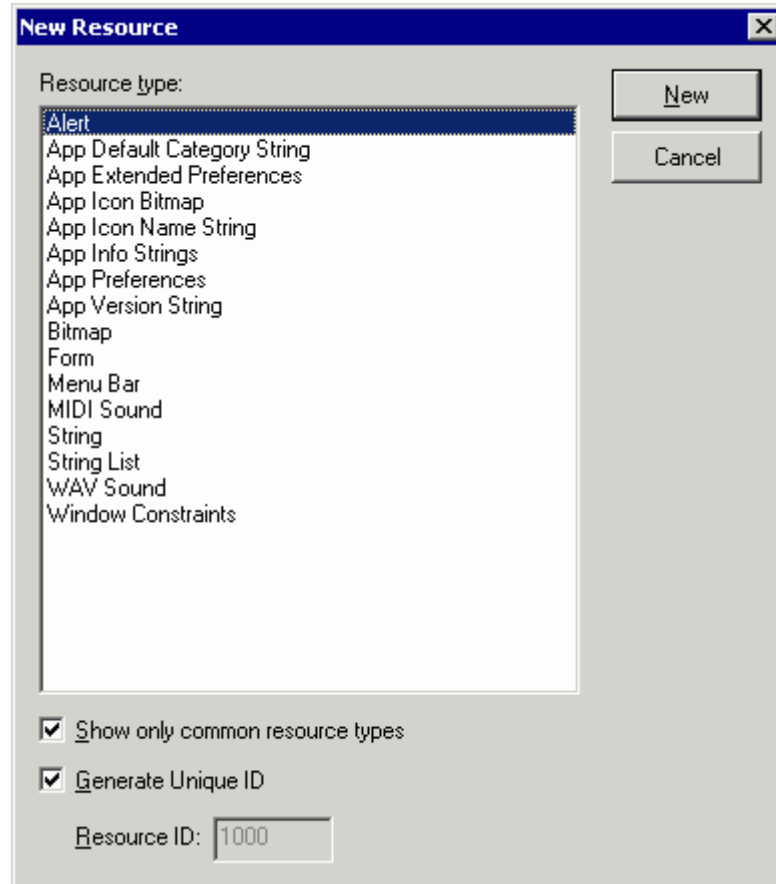
New Resource

Creates a new item of the same resource type as the currently selected resource.

New Resource...

Opens the **New Resource** dialog box, shown in [Figure 9.4](#) on page 84, so that you can select the type of the resource you want to create.

Figure 9.4 New Resource dialog box



Edit Resource

Resource Editor includes custom editors for the most common resources. It also includes an XML Editor and a Hex Viewer that you can use to work with resource types for which no custom editor is available. The items in the Edit Resource menu are:

- Default Editor - The custom editor for the currently selected resource, if one exists, or the XML Editor.
- XML Editor - A text editor showing the XML resource description for the currently selected resource.
- Hex Viewer - A text viewer showing the binary form of the currently selected resource.

Form

Use the **Form** menu to arrange controls in the Form Editor.

Align Lefts

Aligns selected items along the left of the left-most selected item.

Align Centers

Aligns selected items halfway between the left of the left-most selected item and the right of the right-most selected item.

Align Rights

Aligns selected items along the right of the right-most selected item.

Align Tops

Aligns selected items along the top of the topmost selected item.

Align Middles

Aligns selected items halfway between the top of the topmost selected item and the bottom of the bottommost selected item.

Align Bottoms

Aligns selected items along the bottom of the bottommost selected item.

Make Same Widths

Changes the size of the selected items so that they all have the same width as the first item that you selected.

Make Same Height

Changes the size of the selected items so that they all have the same height as the first item that you selected.

Make Same Size

Changes the size of the selected items so that they all have the same size as the first item that you selected.

Center Vertically

Moves the selected items so that the center of the “bounding box” containing all of the items is halfway down the length of the form. The position of the items within the “bounding box” is unchanged.

Center Horizontally

Moves the selected items so that the center of the “bounding box” containing all of the items is halfway down the width of the form. The position of the items within the “bounding box” is unchanged.

Create Navigation Resource

Creates a default form navigation resource for the currently selected form. The resulting XML can be edited in the XML Editor.

Image

Use the **Image** menu to access tools in the Bitmap Editor

Rectangle Selection Tool

Selects the rectangle selection tool, so that you can draw a rectangle selection area in the editor window.

Color Selection Tool

Selects the color selection tool, so that you can select a single color in the editor window.

Fill Tool

Selects the fill tool, so that you can fill an area defined by a single color with another color.

Magnification Tool

Selects the magnification tool, so that you can zoom in on a rectangular area.

Pencil Tool

Selects the pencil tool, so that you can place single pixels of the currently selected foreground color.

Brush Tool

Selects the brush tool, so that you can draw by sketching with a selected brush.

Air Brush Tool

Selects the air brush tool, so that you can spray random pixels of the foreground color onto the image.

Line Tool

Selects the line tool, so that you can draw a point-to-point line with the currently selected foreground color.

Rectangle Tool

Selects the rectangle tool, so that you can draw a rectangle with the currently selected foreground color. (The interior of the rectangle is left unchanged.)

Outlined Rectangle Tool

Selects the outlined rectangle tool, so that you can draw a rectangle framed with the currently selected foreground color and filled with the currently selected background color.

Filled Rectangle Tool

Selects the filled rectangle tool, so that you can draw a rectangle framed and filled with the currently selected foreground color.

Rounded Rectangle Tool

Selects the rounded rectangle tool, so that you can draw a rounded rectangle with the currently selected foreground color. (The interior of the rectangle is left unchanged.)

Outlined Rounded Rectangle Tool

Selects the outlined rounded rectangle tool, so that you can draw a rounded rectangle framed with the currently selected foreground color and filled with the currently selected background color.

Menu Reference

Window

Filled Rounded Rectangle Tool

Selects the filled rounded rectangle tool, so that you can draw a rounded rectangle framed and filled with the currently selected foreground color.

Ellipse Tool

Selects the ellipse tool, so that you can draw an ellipse with the currently selected foreground color. (The interior of the ellipse is left unchanged.)

Outlined Ellipse Tool

Selects the outlined ellipse tool, so that you can draw an ellipse framed with the currently selected foreground color and filled with the currently selected background color.

Filled Ellipse Tool

Selects the filled ellipse tool, so that you can draw an ellipse framed and filled with the currently selected foreground color.

Window

Use the Window menu to change how the editor windows are displayed, to close the editor windows, or to select which editor window is in the foreground.

Arrange Icons

Arranges any minimized editor windows.

Cascade Windows

Arranges any non-minimized and non-maximized editor windows so that they are the same size and overlapping down a diagonal line from upper-left to lower-right.

Tile Horizontally

Arranges any non-minimized and non-maximized editor windows as horizontal tiles, making them all visible at the same time.

Tile Vertically

Arranges any non-minimized and non-maximized editor windows as vertical tiles, making them all visible at the same time.

Close Editor

Closes the currently selected editor window.

Close All Editors

Closes all of the currently open editor windows.

Toolbar

Toggles the display of the toolbar panel.

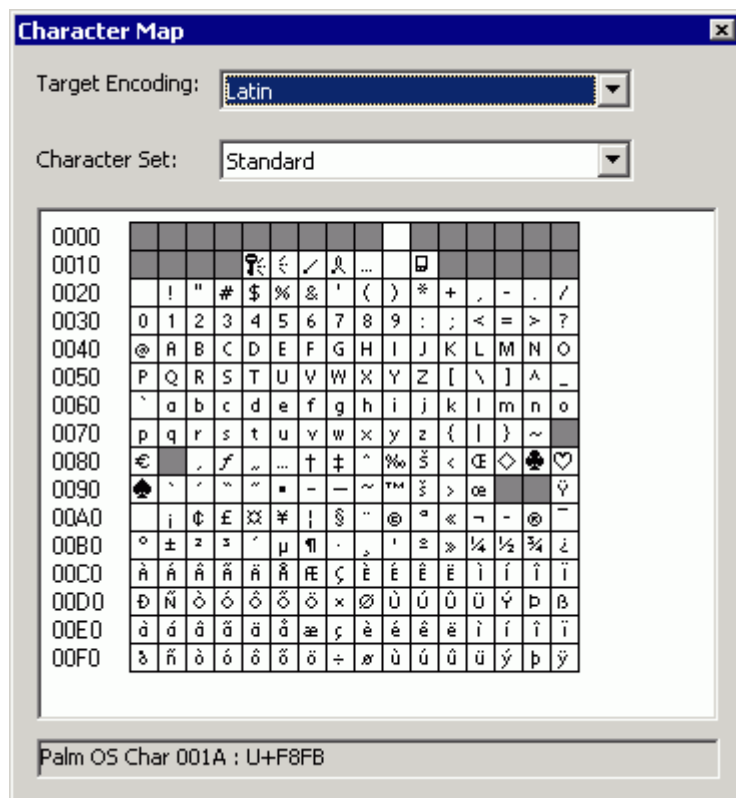
Color Palette

Toggles the display of the color palette in the Bitmap Editor.

Character Map

Toggles the display of the character map window, shown in [Figure 9.5](#).

Figure 9.5 Character map window



Help

Use the **Help** menu to get information about Palm OS Resource Editor.

Help Topics

Opens the Palm OS Resource Editor help file.

About Palm OS Resource Editor

Opens the Palm OS Resource Editor *About* dialog box.

Index

Numerics

68K application resource 21

A

about Palm OS Resource editor 90
air brush tool 87
align bottoms 85
align centers 85
align lefts 85
align middles 85
align rights 85
align tops 85
aligning resources 85
arrange icons 88
auto-shift (field) 38

B

binary editor 75, 84
binary viewer 12
bitmap editor 86
book organization vii
brush tool 87

C

cascade windows 88
center horizontally 86
center vertically 86
centering resources 86
character map window 63, 89
clear 82
close 81
close all editors 89
close editor 89
closing files 81, 82
color palette 89
color selection tool 86
copy 82
copying a resource 83
create navigation resource 86
creating files 79
creating new resources 83
custom editor 84

custom resources 7

cut 82

D

default editor 84
default resource editor 84
developer tools viii
dialog box
 new 79
 new resource 84
 open 80
 save as 81
drawing a bitmap 86
duplicate 83

E

edit menu
 clear menu item 82
 copy menu item 82
 cut menu item 82
 duplicate menu item 83
 edit resource 84
 find menu item 82
 find next menu item 83
 new item menu item 83
 new resource menu item 83
 paste menu item 82
 replace menu item 83
 select all menu item 83
 undo menu item 82
 validate resources menu item 83
edit menu item 82
edit resource 84
editing resources 82
editor
 bitmap 86
 default 84
 form 85
 hex 75, 84
 XML 12, 75, 84
ellipse tool 88
exit 82

F

- field resource
 - auto-shift 38
 - has scrollbar 38
- file 79
- file menu
 - close menu item 81
 - exit menu item 82
 - new menu item 79
 - open menu item 80
 - recent documents menu item 82
 - recent workspaces menu item 82
 - save all menu item 81
 - save as menu item 81
 - save menu item 81
- file menu item 79
- fill tool 86
- filled ellipse tool 88
- filled rectangle tool 87
- filled rounded rectangle tool 88
- filling with color 86
- find 82
- find next 83
- finding text 82
- form editor 85
- form menu
 - align bottoms menu item 85
 - align centers menu item 85
 - align lefts menu item 85
 - align middles menu item 85
 - align rights menu item 85
 - align tops menu item 85
 - center horizontally menu item 86
 - center vertically menu item 86
 - create navigation resource menu item 86
 - make same heights menu item 85
 - make same size menu item 85
 - make same widths menu item 85
- form menu item 85

G

- GenerateXRD 2
- generating 68K resources 21
- generating Palm OS Protein resources 21

H

- has scrollbar (field) 38
- help menu
 - about Palm OS Resource Editor menu item 90
 - help topics menu item 90
- help menu item 90
- hex editor 75, 84
- hex viewer 12
- hiding the character map window 89
- hiding the color palette 89
- hiding the toolbar 89

I

- ID
 - setting the resource ID 6
- image menu
 - air brush tool menu item 87
 - brush tool menu item 87
 - color selection tool menu item 86
 - ellipse tool menu item 88
 - fill tool menu item 86
 - filled ellipse tool menu item 88
 - filled rectangle tool menu item 87
 - filled rounded rectangle tool menu item 88
 - line tool menu item 87
 - magnification tool menu item 86
 - outlined ellipse tool menu item 88
 - outlined rectangle tool menu item 87
 - outlined rounded rectangle tool menu item 87
 - pencil tool menu item 86
 - rectangle selection tool menu item 86
 - rectangle tool menu item 87
 - rounded rectangle tool menu item 87
- image menu item 86

L

- line tool 87
- locales
 - selecting 17

M

- magnification tool 86
- make same heights 85
- make same size 85

make same widths 85
menu item
 edit 82
 edit > clear 82
 edit > copy 82
 edit > cut 82
 edit > duplicate 83
 edit > edit resource 84
 edit > find 82
 edit > find next 83
 edit > new item 83
 edit > new resource 83
 edit > paste 82
 edit > replace 83
 edit > select all 83
 edit > undo 82
 edit > validate resources 83
file 79
 file > close 81
 file > exit 82
 file > new 79
 file > open 80
 file > recent documents 82
 file > recent workspaces 82
 file > save 81
 file > save all 81
 file > save as 81
form 85
 form > align bottoms 85
 form > align centers 85
 form > align lefts 85
 form > align middles 85
 form > align rights 85
 form > align tops 85
 form > center horizontally 86
 form > center vertically 86
 form > create navigation resource 86
 form > make same heights 85
 form > make same size 85
 form > make same widths 85
help 90
 help > about Palm OS Resource Editor 90
 help > help topics 90
image 86
 image > air brush tool 87
 image > brush tool 87
 image > color selection tool 86

 image > ellipse tool 88
 image > fill tool 86
 image > filled ellipse tool 88
 image > filled rectangle tool 87
 image > filled rounded rectangle tool 88
 image > line tool 87
 image > magnification tool 86
 image > outlined ellipse tool 88
 image > outlined rectangle tool 87
 image > outlined rounded rectangle tool 87
 image > pencil tool 86
 image > rectangle selection tool 86
 image > rectangle tool 87
 image > rounded rectangle tool 87
window 88
 window > arrange icons 88
 window > cascade windows 88
 window > character map 89
 window > close all editors 89
 window > close editor 89
 window > color palette 89
 window > tile horizontally 88
 window > tile vertically 89
 window > toolbar 89
minimized editor window 88

N

new 79
new dialog box 79
new item 83
new resource 83
new resource dialog box 84

O

open 80
open dialog box 80
opening files 80
outlined ellipse tool 88
outlined rectangle tool 87
outlined rounded rectangle tool 87

P

Palm OS Developer Suite viii
Palm OS Protein application resource 21

PalmRC 2
paste 82
pencil tool 86
PRC file 2
PRCMerge 2
Protein
 see Palm OS Protein 21

R

Raw Editor 7
recent documents 82
recent workspaces 82
rectangle selection tool 86
rectangle tool 87
replace 83
replacing text 83
resizing resources 85
resource
 creating 2
 definition 2
resource description files
 creating 5
 custom 7
 opening 8
resource editor 84
resource ID
 how to set 6
 restrictions 2
rounded rectangle tool 87
RSRC file 2

S

save 81
save all 81
save as 81
save as dialog box 81
saving files 81
searching for text 82
select all 83
selecting bitmap area 86
selecting bitmap color 86
showing the character map window 89
showing the color palette 89
showing the toolbar 89

T

tile horizontally 88
tile vertically 89
toolbar 89
tools documentation viii
 Introduction to Palm OS Tools viii
 Language and Library Reference viii
 Palm OS Debugger viii
 Palm OS Resource Editor viii
 Resource Tools viii
 Virtual Phone ix
TRC file 2

U

undo 82

V

validating resources 83
viewer
 hex 12

W

window menu
 arrange icons menu item 88
 cascade windows menu item 88
 character map menu item 89
 close all editors menu item 89
 close editor menu item 89
 color palette menu item 89
 tile horizontally menu item 88
 tile vertically menu item 89
 toolbar menu item 89
window menu item 88
workspaces 4
 creating 5
 opening 4

X

XML editor 12, 75, 84
XRD file 79
XRD files
 definition 2
XRT file extension 7
XRW file 79

XRW files 80

Z

zoom 86

