



Palm OS[®] Resource File Formats

Palm OS[®] Developer Suite

Written by Brian Maas.

Technical assistance from Eric Cloninger, Kevin Cooper, David Fedor, Keith Rollin, JB Parrett, Dianne Hackborn, Jie Su and Dan Sandler.

Copyright © 1996–2004, 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, Palm Powered, 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 File Formats

Document Number 3121-005

October 26, 2004

For the latest version of this document, visit

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

PalmSource, Inc.

1240 Crossman Avenue

Sunnyvale, CA 94089

USA

www.palmsource.com

Table of Contents

About This Book	vii
Who Should Read This Book	vii
What This Book Contains	vii
Additional Resources	viii
The Exploring Palm OS Series	viii
Palm OS Developer Suite Documentation	x
 1 Creating XML Resource Files	 1
Introducing XML	2
File Structure and Formatting Rules	2
XML Text Files	2
Creating XRD Files	3
XRD File Content Structure	4
Understanding XML Resource Descriptions	6
Common Resource Attributes	6
Resource ID Attribute.	6
Locale Attribute	7
Overlay Status Attribute	8
Comment Attribute.	8
 2 Specifying XRD Data Types	 9
Binary Data	10
Boolean	10
Enum (Enumerated Value-Label)	10
External File Reference	11
Four-Character Code	11
Integer Number	12
Quoted Text	12
Special Characters	12
Using Escape Sequences.	13
 3 XRD Resource Element Reference	 15
Reference Conventions	15
Common Variables	15

Common Resource Elements.	16
Palm OS Resource File.	17
Resource Element Categories.	21
Application Resource Types	21
Data Resource Types	22
User Interface Types	22
Resource Element Reference	23
Alert Resource	24
Application Default Category Resource	26
Application Extended Preferences Resource	26
Application Icon Bitmap Resource	27
Application Icon Name Resource	33
Application Information Strings Resource	33
Application Launch Preferences Resource	34
Application Preferences Resource	38
Application Version Resource	39
Bitmap Resource	39
Byte Integer List Resource	43
Code Signing Security Certificate Resource	44
Code Signing Security Signature Resource	44
Color Table Resource	45
DWord Integer List Resource	46
Extended Font Resource	47
Font Resource	51
Form Resource	54
Form Navigation Resource	58
Form Object Descriptions	61
Graphic Family Resource	97
Menu Bar Resource	99
MIDI Resource	104
Overlay Resource	105
Raw Resource	106
Schema Database Resource	107
Soft Constant Resource	113
String List Resource	113

String Resource	114
TrueType Font Resource	114
Wave Sound Resource	115
Window Constraints Resource	116
Word Integer List Resource	118
4 Custom Resources	119
Reference Syntax	120
Palm OS Custom Resource Type Description	120
Palm OS Custom Resource Type Use	121
Custom Resource Element	122
Custom Types Files Element	123
Custom Enumeration Description.	124
Define Enum Type Element	124
Custom Structure Description	126
Define Struct Type Element	126
Custom Resource Description	127
Define Resource Type Element	127
Name Format Requirements	129
Data Model Objects	130
Data Field Items	131
Meta-Data Field Items	143
Format Control Items	145
Index	147

About This Book

Palm OS Resource File Formats describes the formats for files used by the Palm OS® resource tools.

This book does not cover the following:

- How to programmatically work with the user interface elements that are part of the Palm OS UI Library. For information on the user interface API functions, see *Exploring Palm OS: User Interface*.
- How to use a resource editor to create UI elements. See the documentation that came with your development environment on how to use the resource editor.
- How to design an interface that is user-friendly and conforms to Palm OS guidelines. See the book *Palm OS User Interface Guidelines* for this type of information.

Who Should Read This Book

Palm OS Resource File Formats is intended for the following readers:

- Palm OS application developers who want to use the ARM-native Palm OS tools to develop applications.
- Tools vendors or implementers who are creating tools that cooperate with the ARM-native Palm OS tools.

This book is intended to be used with *Palm OS Resource Tools Guide*. *Palm OS Resource Tools Guide* provides guidance information on how to use the Palm OS resource tools developed by PalmSource.

What This Book Contains

This book has the following organization.

- [Chapter 1, “Creating XML Resource Files,”](#) on page 1 introduces XML and describes how to use XML files to define Palm OS resources.
- [Chapter 2, “Specifying XRD Data Types,”](#) on page 9 describes how to define the basic data types: binary, boolean, enumerated, external file, four-character code, integer number, and quoted text data.

About This Book

Additional Resources

- [Chapter 3, “XRD Resource Element Reference,”](#) on page 15 provides an alphabetical reference of all of the XRD resource elements, and a summary categorized list of the elements.
- [Chapter 4, “Custom Resources,”](#) on page 119 describes how you can create custom resources in XML: enumerations, structures, and resource types.

Additional Resources

- Documentation
PalmSource publishes its latest versions of this and other documents for Palm OS developers at
<http://www.palmos.com/dev/support/docs/>
- Training
PalmSource and its partners host training classes for Palm OS developers. For topics and schedules, check
<http://www.palmos.com/dev/training>
- Knowledge Base
The Knowledge Base is a fast, web-based database of technical information. Search for frequently asked questions (FAQs), sample code, white papers, and the development documentation at
<http://www.palmos.com/dev/support/kb/>

The Exploring Palm OS Series

The Palm OS Protein API documentation is called the *Exploring Palm OS* series. Together, the books in this series document and explain how to use the APIs exposed to third-party developers by the fully ARM-native versions of Palm OS, beginning with Palm OS Cobalt. Each of the books in the *Exploring Palm OS* series explains one aspect of the Palm operating system, and each contains both conceptual and reference documentation for the pertinent technology.

IMPORTANT: The *Exploring Palm OS* series is intended for developers creating Palm OS Protein applications for Palm OS Cobalt. If you are interested in developing applications that work through PACE and that also run on earlier Palm OS releases, read the latest versions of the *Palm OS Programmer's API Reference* and *Palm OS Programmer's Companion* instead.

As of this writing, the complete *Exploring Palm OS* series consists of the following titles:

- *Exploring Palm OS: Programming Basics*
- *Exploring Palm OS: Memory, Databases, and Files*
- *Exploring Palm OS: User Interface*
- *Exploring Palm OS: System Management*
- *Exploring Palm OS: Text and Localization*
- *Exploring Palm OS: Input Services*
- *Exploring Palm OS: High-Level Communications*
- *Exploring Palm OS: Low-Level Communications*
- *Exploring Palm OS: Telephony and SMS*
- *Exploring Palm OS: Multimedia*
- *Exploring Palm OS: Security and Cryptography*
- *Exploring Palm OS: Porting Applications to Palm OS Cobalt*
- *Exploring Palm OS: Palm OS File Formats*

About This Book

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.

About This Book

Palm OS Developer Suite Documentation

Creating XML Resource Files

This chapter covers the following topics:

Introducing XML	2
Creating XRD Files	3
Understanding XML Resource Descriptions	6
Common Resource Attributes	6

Introducing XML

XML is a text format, similar to HTML in structure, although more powerful. An XML file represents a hierarchical data structure. The basic building blocks of an XML file are elements and attributes. An element is a named container. An element may have attributes, text data, and child elements. Attributes are named value tags.

Listing 1.1 XML File Elements and Attributes

```
<SAMPLE ID="1" NAME="my_samp">
  <BAR> 3 </BAR>
</SAMPLE>
```

In this example, shown in [Listing 1.1](#), `SAMPLE` is the main element. It has two attributes, one named `ID` and one named `NAME`. It has one child element named `BAR` with text 3.

File Structure and Formatting Rules

- Element and attribute names are case sensitive. For example, `SAMPLE` is not the same as `Sample`.
- Attribute values must always be quoted.
- White space (blanks, tabs, line endings) is not significant between tags.
- The order of attributes is not significant.
- The order of elements is significant.

XML Text Files

To minimize operating system differences, hardware differences, and language differences, Palm OS resource XML files are stored in Unicode format. (Unicode is a character encoding format that is becoming standard for text file interchange.) The Palm OS resource tools currently support UTF-16 and UTF-8 Unicode character encoding.

IMPORTANT: Palm OS resource tools work with XML files in either UTF-16 or UTF-8 Unicode character encoding. UTF-8 is recommended for compatibility with text editors and source control tools that may not handle the binary data in UTF-16 files correctly.

If you use a text editor that saves files in an encoding format other than UTF-16 or UTF-8 Unicode character encoding, such as a Windows native code page, you may have problems when compiling or editing the XRD file with Palm OS Resource Tools like PalmRC and PRCMerge.

Creating XRD Files

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

Palm OS XRD files use XML structure and text encoding. However, XRD files limit the use of XML in the following ways:

- To add comments to your XRD files, you should use comment attributes and elements rather than XML-style comments. See “[Comment Attribute](#)” on page 8 for information on comment attributes.

Because comment attributes and elements are stored as logical attributes of the resource file, they can be retained in the memory representation of the resource file.

IMPORTANT: XML-style comments will be ignored by PalmRC, and they will be stripped out if you use other Palm OS resource tools to save or regenerate them.

- External data files, such as bitmap files or MIDI files, are supported using content syntax rather than XML external entity references. See “[External File Reference](#)” on page 11 for information on specifying external data files.

XRD File Content Structure

XRD files must follow the structure defined in [Listing 1.2](#).

Listing 1.2 XRD File Content Structure

```
XML header
PALMOS_RESOURCE_FILE element
    optional DATABASE_HEADER element
    0 or more resource elements
```

XML Header

The XML header identifies the file as an XML file. The XML header may also specify that the document is using a DTD (document type definition). The DTD may be used to validate the structure of the comment by validating XML parsers, editors, and other processing tools. [Listing 1.3](#) shows an example of an XML header.

Listing 1.3 XML Header Example

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
```

PALMOS_RESOURCE_FILE

The PALMOS_RESOURCE_FILE element is a standard element. There are no attributes defined for this element. For more information about this element, see “[Palm OS Resource File](#)” on page 17.

DATABASE_HEADER element

The DATABASE_HEADER element stores the physical fields of a PRC database header. Normally, this element is not used in resource files because physical field attributes are set by the PRCMerge tool when a Palm OS executable is built. The main use for the DATABASE_HEADER element is to test conversion of PRC files.

If GenerateXRD is used on a PRC file with the “-d” option, the DATABASE_HEADER element is generated in the resulting XRD file.

When the XRD file is compiled by PalmRC, the resulting PRC file has the same database header as the original PRC file

(with the exception of the creation and modification date and timestamp fields, which PalmRC updates).

[Listing 1.4](#) shows an example of a DATABASE_HEADER element.

Listing 1.4 DATABASE_HEADER Example

```
<DATABASE_HEADER>
  <DB_NAME> "Starter" </DB_NAME>
  <DB_FLAG_RESET> FALSE </DB_FLAG_RESET>
  <DB_FLAG_BACKUP> TRUE </DB_FLAG_BACKUP>
  <DB_FLAG_HIDDEN> FALSE </DB_FLAG_HIDDEN>
  <DB_FLAG_COPY_PROTECT> FALSE </DB_FLAG_COPY_PROTECT>
  <DB_VERSION> 2 </DB_VERSION>
  <DB_MOD_NUM> 0 </DB_MOD_NUM>
  <DB_TYPE> 'appl' </DB_TYPE>
  <DB_CREATOR> 'STRT' </DB_CREATOR>
  <DB_UNIQUE_ID> 0x00000000 </DB_UNIQUE_ID>
  <DB_APP_INFO> </DB_APP_INFO>
  <DB_SORT_INFO> </DB_SORT_INFO>
</DATABASE_HEADER>
```

resource elements

The main content of the resource file is a set of resource element descriptions. The resource descriptions may be defined in any order, but it is best if you keep them in a canonical order and format. Canonical order helps when you use text comparison and source control tools to track resource changes during development.

The canonical order of resources is alphabetical by resource element name, then in numerical order by resource ID, then in order by locale. In addition, RAW_RESOURCE elements are sorted by their RES_TYPE element values before being sorted by resource ID.

For information about specific resource elements, see “[Resource Element Reference](#)” on page 23.

Listing 1.5 Example of a Valid XRD File

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<PALMOS_RESOURCE_FILE>
```

Creating XML Resource Files

Understanding XML Resource Descriptions

```
<STRING_RESOURCE RESOURCE_ID="1000">
  <TEXT> "Palm OS" </TEXT>
</STRING_RESOURCE>

</PALMOS_RESOURCE_FILE>
```

Understanding XML Resource Descriptions

There is a one-to-one correspondence between XML resource descriptions and Palm OS resources in the PRC file. In general, the XML resource element name should match the logical resource type name.

For example, a Palm OS "string" resource, which is a Palm OS 'tSTR' resource, is a `STRING_RESOURCE` resource in the XML resource description file.

Listing 1.6 Example of a Resource Description File

```
<STRING_RESOURCE
  RESOURCE_ID="1000"
  LOCALE="enUS"
  COMMENT="Language Name">
  <TEXT> "English" </TEXT>
</STRING_RESOURCE>
```

Common Resource Attributes

The following resource attributes are common to most of the XML resource descriptions.

Resource ID Attribute

`RESOURCE_ID="resource_id"`

Every resource element in the Palm OS resource file must have a `RESOURCE_ID` attribute. Developer-assigned resource IDs must be an unsigned integer in the range 0 to 9999.

Locale Attribute

`LOCALE="llCC"`

Each resource may optionally have a `LOCALE` attribute. This is used in conjunction with the parameters to `PalmRC` to strip resources into locale independent base resources, which are stored in `BPRC` files, and localized resources, which are stored in locale specific overlay resource files (`OPRC` files).

If the resource `LOCALE` attribute is not specified, or specified as the empty string (`""`), the resource is not locale specific and is placed in the base `PRC` file. If the resource `LOCALE` is specified, it indicates that the resource goes into the overlay `PRC` file for the specified locale.

Locales are indicated using 4-character codes specified by ISO 639 Language and ISO 3166 Country Codes. The locale values used by Palm OS currently are:

`enUS`

English

`frFR`

French

`itIT`

Italian

`deDE`

German

`esES`

Spanish

`jaJP`

Japanese. **Note:** For Palm OS Cobalt, you should use `jaJP`; for earlier versions of Palm OS, you should use `jpJP` instead.

`zhCN`

Simplified Chinese.

For more information, see the following web pages:

<http://www.loc.gov/standards/iso639-2/englangn.html>

See 639-1 alpha-2.

Creating XML Resource Files

Common Resource Attributes

<http://www.niso.org/3166.html>

See 3166-1.

Overlay Status Attribute

`OVERLAY_STATUS="overlay_info"`

Each resource may optionally have an `OVERLAY_STATUS` attribute. The only explicit value currently supported is "ADD". If the attribute is not specified, or is specified as the empty string (" "), the resource is treated as a `REPLACE` overlay.

The `OVERLAY_STATUS` attribute is used to validate that overlay resource files are correct. If a resource is present in a specific locale, but not in the base locale, it should specify the attribute as `ADD`.

Comment Attribute

`COMMENT="comment_text"`

Each resource may optionally have an `COMMENT` attribute. This should typically be a short string (that is, a few words at most) which describes the resource or its usage. For larger amounts of text, such as instructions for localization, use the `COMMENT_TEXT` element.

Specifying XRD Data Types

This chapter describes the Palm OS resource descriptions for the following basic data types:

Binary Data 10
Boolean 10
Enum (Enumerated Value-Label) 10
External File Reference 11
Four-Character Code 11
Integer Number 12
Quoted Text 12

The Palm OS resource file structure allows data to be specified only in leaf elements. The data is specified as the XML text content of the element.

As defined by XML, white space is not significant in the text content, but it may be interpreted by the XML application. To avoid any possible misinterpretation of white space (when it is necessary to preserve text content accurately, such as in string resources), you should use the [Quoted Text](#) syntax that explicitly preserves the white space regardless of the XML text formatting.

Specifying XRD Data Types

Binary Data

Binary Data

Binary data values are specified as a sequence of bytes encoded as hexadecimal. The canonical format is two characters per byte with a space between bytes.

Example

```
<DATA> 48 65 6C 6C 6F </DATA>
<DATA>
    00 00 00 30 00 00 00 08 00 00 00 08 00 00 00 20
    00 00 3F 3C 00 01 A9 F0
</DATA>
```

Boolean

A boolean value is a specific case of an enumerated type. The text labels accepted are FALSE and TRUE.

Examples

```
<HAS_COLOR_TABLE> FALSE </HAS_COLOR_TABLE>
<HAS_TRANSPARENCY> TRUE </HAS_TRANSPARENCY>
```

Enum (Enumerated Value-Label)

Palm OS defines some fields with a specific set of logical values, and assigns text labels to these logical values. Generally, you should use the text labels in the XRD file; but in most contexts, the integer number value of the enum is also accepted. Text labels are not case sensitive, but it is best to use the canonical uppercase form.

Examples

```
<COMPRESSION> NONE </COMPRESSION>
<COMPRESSION> SCANLINE </COMPRESSION>
```

External File Reference

An external file reference is a quoted string with a path to an external file. Normally, paths should be specified as relative to the location of the XRD file. Paths use a syntax similar to URL syntax.

For cross platform considerations, file and directory names should be 31 characters or less and should use a restricted character set. For example, use only Roman alphanumeric characters and underscore characters (' _ '). Refrain from using special characters. In particular, colon (' : '), backslash (' \ '), and question mark (' ? ') characters should not be used in file or directory names.

Examples

```
<DATA_FILE> "mydata.bin" </DATA_FILE>
<DATA_FILE> "./mydata.bin" </DATA_FILE>
<DATA_FILE> "./datafiles/mydata.bin" </DATA_FILE>
<DATA_FILE> "../datafiles/mydata.bin" </DATA_FILE>
```

Four-Character Code

A four-character code is used as a type code for physical resource types, PRC database types and creators, and similarly defined Palm OS constructs. A four-character code is denoted by single quote marks surrounding exactly 4 characters. Each character must be alphanumeric in low ASCII; that is, either lower case letters (' a ' through ' z '), uppercase letters (' A ' through ' Z ') or numbers (' 0 ' through ' 9 '). In some contexts, four-character codes may also be specified as integer values (typically specified in hexadecimal format).

Examples

```
<DB_TYPE> 'appl' </DB_TYPE>
<DB_CREATOR> 'MyAp' </DB_CREATOR>
```

Specifying XRD Data Types

Integer Number

Integer Number

Integer number values may be specified as decimal or hexadecimal. Hexadecimal values are prefixed with the characters "0x". In some contexts, integers may also be specified as four-character codes.

Examples

```
<ID> 255 </ID>
<ID> 0xFF </ID>
<DWORD> 0x6170706C </DWORD>
```

Quoted Text

A quoted text value is a string value. The string is specified with one or more segments which are concatenated together. A segment starts and ends with a double quotation mark ("). All text between the quotes is considered part of the string.

To avoid corruption of the string during XML processing, segments should never contain embedded line endings (for example, line feed or carriage return). That is, segments should never span lines in the XML file.

Special Characters

You can use XML character reference syntax to enter special characters in quoted text, especially for characters that would otherwise be interpreted as XML description characters.

You may use either predefined character references or numeric character references. For more information on XML character references, see this URL: <http://www.w3.org/TR/REC-xml>.

Examples of special characters

&	Ampersand (&)
<	Less than (<)
>	Greater than (>)
'	Apostrophe (')
"	Double quote (")
…	Unicode code point hex 2026, an ellipsis (...)

IMPORTANT: Numeric character references specify the Unicode code point of the character, not the Palm OS code point. PalmRC transliterates the character to the Palm OS code point depending on the target text encoding.

Using Escape Sequences

To include some special characters in quoted text, you may need to use an escape sequence.

- To embed a line feed in a string, use the escape sequence `\n`.
- To embed a double quotation mark in a string, use the escape sequence `\"`.
- To embed a backslash character itself, use the escape sequence `\\`.

It is also possible to embed control characters using the escape sequence `\xHH` where HH is the hexadecimal code of the character.

IMPORTANT: The embedded control character is interpreted as part of a Unicode string and is subject to transliteration, as usual, when compiling.

Listing 2.1 Escape Sequences

```
\n    Line feed
\r    Carriage return (not normally used by Palm OS!)
\t    Tab
\\    Backslash
\"    Double quotation mark
\xHH  Control character as hexadecimal character code
```

Examples

```
<TEXT> "OK" </TEXT>
```

```
<TEXT>
    "Use this application as a\n"
    "starting point for your own\n"
```

Specifying XRD Data Types

Quoted Text

```
"exciting Palm applications!"  
</TEXT>
```

XRD Resource Element Reference

This chapter provides reference for the XRD resource elements.

- “[Reference Conventions](#)” on page 15
- “[Palm OS Resource File](#)” on page 17
- “[Resource Element Categories](#)” on page 21
- “[Resource Element Reference](#)” on page 23

Reference Conventions

The following conventions are used in the description of the XRD resource elements.

Common Variables

The reference descriptions in this chapter use the following variables for common data types, attributes, and elements:

binary_data

As described in the section “[Binary Data](#)” on page 10.

boolean

As described in the section “[Boolean](#)” on page 10.

enum

As described in the section “[Enum \(Enumerated Value-Label\)](#)” on page 10.

ext_file

As described in the section “[External File Reference](#)” on page 11.

four_character_code

As described in the section “[Four-Character Code](#)” on page 11.

XRD Resource Element Reference

Reference Conventions

integer

As described in the section “[Integer Number](#)” on page 12.

quoted_text

As described in the section “[Quoted Text](#)” on page 12.

resource_id

As described in the section “[Resource ID Attribute](#)” on page 6.

text_element

As described in the section “[Text Element](#)” on page 16.

Common Resource Elements

Many resource elements use COMMENT_TEXT and TEXT elements. These elements are described in this section.

Comment Text Element

Purpose Use the COMMENT_TEXT element to include comments in the XRD file.

Tag `<COMMENT_TEXT> quoted_text </COMMENT_TEXT>`
The text of the comment. The text must follow the XRD quoted text format.

Text Element

Purpose Use the TEXT element to include text where needed.

Tag `<TEXT> quoted_text </TEXT>`
The text must follow the XRD quoted text format.

Palm OS Resource File

The Palm OS resource file element, `PALMOS_RESOURCE_FILE`, is the XML element that encloses all other XRD resource elements. This element defines an XRD file.

The Palm OS resource file element consists of an optional `DATABASE_HEADER` element followed by the additional resource elements, which are described in “[Resource Element Reference](#)” on page 23.

Start Tag	<code><PALMOS_RESOURCE_FILE></code>
Child Elements	<code><DATABASE_HEADER> database_def</code> <code></DATABASE_HEADER></code> <p>This element is not normally used in resource files because these attributes are generally set using the PRCMerge tool when a Palm OS application is built.</p> <p>If you use GenerateXRD with a PRC file and specify the <code>-d</code> option, GenerateXRD adds the <code>DATABASE_HEADER</code> element to the created XRD file.</p> <p>The <code>database_def</code> can include any of the following elements:</p> <p><code><DB_NAME> quoted_text </DB_NAME></code> <code>quoted_text</code> - The name of the database as it will appear on the Palm Powered™ device.</p> <p><code><DB_FLAG_RESET> boolean </DB_FLAG_RESET></code> TRUE - You want Palm OS to notify the application of a device reset.</p> <p>FALSE - You do not want Palm OS to notify the application of a device reset.</p> <p><code><DB_FLAG_BACKUP> boolean </DB_FLAG_BACKUP></code> TRUE - The application should be backed up to the desktop computer if no application-specific conduit is available.</p> <p>FALSE - You do not want the application to be backed up to the desktop computer.</p>

XRD Resource Element Reference

Palm OS Resource File

`<DB_FLAG_HIDDEN> boolean </DB_FLAG_HIDDEN>`
TRUE - The database is marked as hidden from view.
For PRC databases, this value tells the Launcher not to show the application. For PDB databases, this value tells the Launcher to disregard the database's records when showing a count of records.
FALSE - The database is not marked as hidden.

`<DB_FLAG_COPY_PROTECT> boolean`
`</DB_FLAG_COPY_PROTECT>`
TRUE - The database is marked as protected from being copied. This value prevents the database from being copied by methods such as IR beaming.
FALSE - The database is not marked as protected from being copied.

`<DB_VERSION> integer </DB_VERSION>`
integer - The application-specific version of the database.

`<DB_MOD_NUM> integer </DB_MOD_NUM>`
integer - The modification number of the database.

`<DB_TYPE> four_character_code </DB_TYPE>`
four_character_code - The database type identifier. For PDB databases, the value of this field depends on the creator application. For PRC databases, this field usually has the value 'appl'.

`<DB_CREATOR> four_character_code`
`</DB_CREATOR>`
four_character_code - The database creator identifier. For more information about creator IDs, see the book *Exploring Palm OS: Programming Basics*.

`<DB_UNIQUE_ID> integer </DB_UNIQUE_ID>`
integer - Used internally by the Palm OS to generate unique identifiers for records on the Palm device when the database is loaded into the device. For PRC databases, this value is normally not used and is set to 0.

<DB_APP_INFO> *binary_data* </DB_APP_INFO>

binary_data - The local offset from the beginning of the database header data to the start of the optional, application-specific application information (app info) block.

<DB_SORT_INFO> *binary_data* </DB_SORT_INFO>

binary_data - The local offset from the beginning of the PDB header data to the start of the optional, application-specific sort information (sort info) block. This value is set to NULL for databases that do not include an sort info block.

<FILE_PREFERENCES> *file_prefs*

</FILE_PREFERENCES>

This element is used by Palm OS Resource Editor to generate a C header file based on the XRD file contents. This element does not generate a binary resource in the output PRC file.

The *file_prefs* can include any of the following elements:

<PREFS_GENERATE_HEADER> *boolean*

</PREFS_GENERATE_HEADER>

TRUE - You want Palm OS Resource Editor to generate a C header file based on the XRD file contents.

FALSE - You do not want Palm OS Resource Editor to generate a C header file.

<PREFS_HEADER_FILE> *quoted_text*

</PREFS_HEADER_FILE>

quoted_text - Optional file name of the header file to generate. If absent, the name of the XRD file is used, with an .h extension.

<VARIABLE_TEMPLATE> *quoted_text*

</VARIABLE_TEMPLATE>

quoted_text - Optional string template to use to generate symbol names based on the contents of the resource.

Resource Elements

Include any of the resource elements defined in this chapter. See “[Resource Element Reference](#)” on page 23 for more information about additional resource elements.

XRD Resource Element Reference

Palm OS Resource File

End Tag `</PALMOS_RESOURCE_FILE>`

Example `<PALMOS_RESOURCE_FILE>`
 `<COMMENT_TEXT>`
 `"AllResTypes.xrd\n"`
 `</COMMENT_TEXT>`
 `<DATABASE_HEADER>`
 `<DB_NAME> "Starter" </DB_NAME>`
 `<DB_FLAG_RESET> FALSE </DB_FLAG_RESET>`
 `<DB_FLAG_BACKUP> FALSE </DB_FLAG_BACKUP>`
 `<DB_FLAG_HIDDEN> FALSE </DB_FLAG_HIDDEN>`
 `<DB_FLAG_COPY_PROTECT> FALSE </DB_FLAG_COPY_PROTECT>`
 `<DB_VERSION> 2 </DB_VERSION>`
 `<DB_MOD_NUM> 0 </DB_MOD_NUM>`
 `<DB_TYPE> 'appl' </DB_TYPE>`
 `<DB_CREATOR> 'aTrK' </DB_CREATOR>`
 `<DB_UNIQUE_ID> 0x00000000 </DB_UNIQUE_ID>`
 `<DB_APP_INFO> </DB_APP_INFO>`
 `<DB_SORT_INFO> </DB_SORT_INFO>`
 `</DATABASE_HEADER>`
 `...`
 `...`
 `</PALMOS_RESOURCE_FILE>`

Resource Element Categories

To help you browse through the types of resource elements available, this section collects the resource elements into categories.

For an alphabetical listing of the resource elements, see “[Resource Element Reference](#)” on page 23.

Application Resource Types

This section lists the Palm OS application resource types.

Table 3.1 Application Resource Types

Resource Element	Described in Section
Application Default Category	“ Application Default Category Resource ” on page 26
Application Extended Preferences	“ Application Extended Preferences Resource ” on page 26
Application Icon Bitmap	“ Application Icon Bitmap Resource ” on page 27
Application Icon Name	“ Application Icon Name Resource ” on page 33
Application Info Strings	“ Application Information Strings Resource ” on page 33
Application Launch Preferences	“ Application Launch Preferences Resource ” on page 34
Application Preferences	“ Application Preferences Resource ” on page 38
Application Version	“ Application Version Resource ” on page 39

Data Resource Types

This section lists the resource types that define data resources.

Table 3.2 Data Resource Types

Resource Element	Described in Section
Byte Integer List	“Byte Integer List Resource” on page 43
DWord Integer List	“DWord Integer List Resource” on page 46
Raw	“Raw Resource” on page 106
Soft Constant	“Soft Constant Resource” on page 113
String	“String Resource” on page 114
String List	“String List Resource” on page 113
Word Integer List	“Word Integer List Resource” on page 118

User Interface Types

This section lists the resource types that can be used to define an application’s user interface.

Table 3.3 User Interface Resource Types

Resource Element	Described in Section
Alert	“Alert Resource” on page 24
Bitmap	“Bitmap Resource” on page 39
Color Table	“Color Table Resource” on page 45
Form	“Form Resource” on page 54
Form Navigation	“Form Navigation Resource” on page 58
Graphic Family	“Graphic Family Resource” on page 97
Menu Bar	“Menu Bar Resource” on page 99
MIDI	“MIDI Resource” on page 104
Wave Sound	“Wave Sound Resource” on page 115
Window Constraints	“Window Constraints Resource” on page 116

Resource Element Reference

The following section lists the resource elements alphabetically. For a categorized list of XRD resources, see “[Resource Element Categories](#)” on page 21.

Alert Resource 24
Application Default Category Resource 26
Application Extended Preferences Resource 26
Application Icon Bitmap Resource 27
Application Icon Name Resource 33
Application Information Strings Resource 33
Application Launch Preferences Resource 34
Application Preferences Resource 38
Application Version Resource 39
Bitmap Resource 39
Byte Integer List Resource 43
Code Signing Security Certificate Resource 44
Code Signing Security Signature Resource 44
Color Table Resource 45
DWord Integer List Resource 46
Extended Font Resource 47
Font Resource 51
Form Resource 54
Form Navigation Resource 58
Graphic Family Resource 97
Menu Bar Resource 99
MIDI Resource 104
Overlay Resource 105
Raw Resource 106
Schema Database Resource 107
Soft Constant Resource 113

XRD Resource Element Reference

Alert Resource

String List Resource	113
String Resource	114
TrueType Font Resource	114
Wave Sound Resource	115
Window Constraints Resource	116
Word Integer List Resource	118

Alert Resource

Purpose	Use the alert resource to define an alert dialog.
Target Format	Format generated by PalmRC: <ul style="list-style-type: none">• Palm OS 4: 'Talt'• Palm OS 5: 'aalt'• Palm OS 6: 'Talt'
Start Tag	<ALERT_RESOURCE RESOURCE_ID="resource_id">
Child Elements	<div><ALERT_TYPE> enum </ALERT_TYPE></div> <div>The ALERT_TYPE element may be specified as one of the following enumeration values:</div> <div>INFORMATION_ALERT</div> <div>Use for alert dialogs when the user should be aware of the information, but the information does not indicate potential problems at a later time.</div> <div>CONFIRMATION_ALERT</div> <div>Use for alert dialogs when the user must acknowledge the alert before continuing.</div> <div>WARNING_ALERT</div> <div>Use for alert dialogs when the application is initiating an irreversible action.</div> <div>ERROR_ALERT</div> <div>Use for alert dialogs that indicate an error has developed that cannot be reversed.</div>

`<HELP_ID> resource_id </HELP_ID>`

Specifies the resource ID of the help text that appears when the user taps the Help icon in an alert.

Entering an ID of 0 means that the alert does not have help text.

`<DEFAULT_BUTTON> integer </DEFAULT_BUTTON>`

Specifies the button index number of the form's default button. Palm OS software simulates the tapping of this button when the form is dismissed.

`<TITLE> quoted_text </TITLE>`

Specifies the title of the alert, which appears at the top of the alert dialog.

`<MESSAGE> quoted_text </MESSAGE>`

Specifies the text that appears in the body of the alert.

`<BUTTONS> button_text </BUTTONS>`

Specifies one or more `<TEXT>` resource elements that each define the text string for a button.

End Tag

`</ALERT_RESOURCE>`

Example

```
<ALERT_RESOURCE RESOURCE_ID="1000">
  <ALERT_TYPE> WARNING_ALERT </ALERT_TYPE>
  <HELP_ID> 0 </HELP_ID>
  <DEFAULT_BUTTON> 2 </DEFAULT_BUTTON>
  <TITLE> "Remove Category" </TITLE>
  <MESSAGE> "Delete category name and all records?"
  </MESSAGE>
  <BUTTONS>
    <TEXT> "Records" </TEXT>
    <TEXT> "Name Only" </TEXT>
    <TEXT> "Cancel" </TEXT>
  </BUTTONS>
</ALERT_RESOURCE>
```

Application Default Category Resource

Purpose	Use the application default category resource to specify a default category for an application.
Target Format	Format generated by PalmRC: <ul style="list-style-type: none">• All OS targets: 'taic'
Start Tag	<pre><APP_DEFAULT_CATEGORY_RESOURCE RESOURCE_ID="resource_id"> resource_id</pre> <p><i>resource_id</i> - The resource identifier 1000 is generally used for the application default category resource.</p>
Child Elements	<pre><TEXT> quoted_text </TEXT></pre> <p>Specifies the default category for the application.</p> <p>You should specify only one default category for the application. PalmSource recommends that you use one of the default categories provided by Palm OS (Games, Main, System, or Utilities) rather than creating your own category.</p>
End Tag	<pre></APP_DEFAULT_CATEGORY_RESOURCE></pre>
Example	<pre><APP_DEFAULT_CATEGORY_RESOURCE RESOURCE_ID="1000"> <TEXT> "Games" </TEXT> </APP_DEFAULT_CATEGORY_RESOURCE></pre> <hr/>

Application Extended Preferences Resource

Purpose	<p>Use the application extended preferences resource to specify version information of this resource and overlay information about the application.</p> <p>If you are compiling for an OS target prior to Palm OS Cobalt and you need to disable overlays for an application or a shared library, you can use this resource. However, for Palm OS Cobalt and later, you should use the application launch preferences resource instead. See “Application Launch Preferences Resource” on page 34.</p>
Target Format	Format generated by PalmRC: <ul style="list-style-type: none">• Palm OS 4: 'xprf'• Palm OS 5: 'axpf'• Palm OS 6: 'xprf'

Start Tag	<code><APP_EXTENDED_PREFS_RESOURCE RESOURCE_ID="resource_id"></code>
Child Elements	<code><VERSION> integer </VERSION></code> Indicates the extended preferences version. For Palm OS Cobalt, use the integer value 1. <code><NO_OVERLAY> boolean </NO_OVERLAY></code> TRUE - Indicates application has no overlay. FALSE - Indicates application has an overlay. <code><RUNTIME_WIDTH_CALCS> boolean </RUNTIME_WIDTH_CALCS></code> TRUE - Indicates that the user interface object widths should be calculated at runtime. FALSE - Indicates that the user interface object widths be set in the resource file. Note: The RUNTIME_WIDTH_CALCS element is not supported by all versions of Palm OS.
End Tag	<code></APP_EXTENDED_PREFS_RESOURCE></code>
Example	<hr/> <code><APP_EXTENDED_PREFS_RESOURCE RESOURCE_ID="1000"> <VERSION> 1 </VERSION> <NO_OVERLAY> FALSE </NO_OVERLAY> </APP_EXTENDED_PREFS_RESOURCE></code> <hr/>

Application Icon Bitmap Resource

- Purpose** The application icon bitmap is used by the Palm OS Launcher to display the icon for an application.
- The bitmap data format is the same as the bitmap resource. The bitmap stores a set of images which represent the same logical image in one or more image formats. For example, it may provide a monochrome (black-and-white) image, grayscale image, and/or color versions of the image.

XRD Resource Element Reference

Application Icon Bitmap Resource

A bitmap resource has one element named BITMAPS. The BITMAPS element should contain one or more BITMAP elements. The BITMAP elements must be defined in order of increasing bit-depth.

Usually, every element of the bitmap family should have the same logical dimensions specified in WIDTH and HEIGHT. The logical width and height are specified redundantly as elements of each BITMAP element. This design accurately reproduces the data stored in Palm OS bitmap structures.

Target Format Format generated by PalmRC:

- All OS targets: 'tAIB'

Start Tag <APP_ICON_BITMAP_RESOURCE
 RESOURCE_ID="resource_id">

To define the large application icon resource, you should generally use the resource ID 1000.

To define the small application icon resource, you should generally use the resource ID 1001.

Child Elements <BITMAPS> one or more BITMAP elements </BITMAPS>
The BITMAPS element consists of a collection of BITMAP elements.

<BITMAP> *bitmap_defn* </BITMAP>

bitmap_defn consists of the following elements:

<WIDTH> *integer* </WIDTH>

integer - For large application icons, the width is generally 22 at normal density, 33 at one-and-one-half density, and 44 at double density. For small application icons, the width is generally 15 at normal density, 22 at one-and-one-half density, and 30 at double density.

<HEIGHT> *integer* </HEIGHT>

integer - For large application icons, the height is generally 22 at normal density, 33 at one-and-one-half density, and 44 at double density. For small application icons, the height is generally 9 at normal density, 13 at one-and-one-half density, and 18 at double density.

`<BIT_DEPTH> integer </BIT_DEPTH>`

integer - Valid values: 1, 2, 4, 8, and 16.

`<ROW_BYTES> integer </ROW_BYTES>`

This element is optional. Normally, this should be omitted, in which case the correct bytes-per-row of image data is calculated based on the image depth and width. However, to accurately reproduce the data stored in Palm OS bitmap structures, this field may be explicitly defined.

`<COMPRESSION> enum </COMPRESSION>`

May be specified as enum values NONE, SCANLINE, RLE, PACKBITS, or BEST, as defined in `CmnBitmapTypes.h`. For more information about compression enum values, see *Exploring Palm OS: User Interface*.

`<BITMAP_DATA> binary_data </BITMAP_DATA>`

The image data of the bitmap may be specified using either a BITMAP_DATA element or a BITMAP_FILE element.

A BITMAP_DATA element specifies the raster image data inline in the XRD file. The image data is specified as binary data. The image data is the uncompressed raster data specified as `rowBytes` times `numScanLines` (`rowBytes * numScanLines`) bytes of data, where the scan lines are stored top to bottom.

`<BITMAP_FILE> ext_file </BITMAP_FILE>`

The image data of the bitmap may be specified using either a BITMAP_DATA element or a BITMAP_FILE element.

A BITMAP_FILE element specifies the image data as the quoted string path of an external image file, which must be in a supported image format such as BMP or PICT. The dimensions of the image in the image file

XRD Resource Element Reference

Application Icon Bitmap Resource

must match exactly the dimensions specified by the BITMAP element.

Palm OS automatically converts the image in the image file via bit-depth conversion to the bit depth specified by the BITMAP element. For example, if you reference a 24-bit red-green-blue BMP file from a 1-bit bitmap, Palm OS automatically converts the image to black and white. Bit depth conversion maps each source pixel to the closest color available in the destination bit depth.

The actual algorithm used for color matching is subject to change. If your application is dependent on specific color use, then make sure you use all of the specific bit depths required rather than relying on automatic conversion.

When using external image files, you can store the image as a BMP file. For simplicity, you can use BITMAP_FILE paths that specify a path relative to the XRD file.

`<BITMAP_DENSITY> enum </BITMAP_DENSITY>`

This optional element allows you to specify multiple density bitmaps in a bitmap resource. Bitmap elements must be defined in ascending order of density, then in increasing order of bit depth for a given density.

72 - Normal (1X) density (also referred to single density). This is the default, if this element is omitted.

108 - 1.5X density (also referred to one-and-one-half density).

144 - Double (2X) density.

216 - Triple (3X) density.

The width and height values defined for each bitmap element must match the physical dimensions.

The normalized dimensions for all bitmap elements in a set must match. For example, if the width of a normal density bitmap element is 10, then the one-

and-one-half density element(s) must have width 15,
the double density element(s) must have width 20,
and the triple density element(s) must have width 30.

End Tag </APP_ICON_BITMAP_RESOURCE>

Example This section includes examples of the following types of application icon bitmap resources:

- [Inline Bitmap Data](#)
- [Bitmap Data from External Image File](#)
- [High Density Bitmap Elements](#)

Inline Bitmap Data

```
<APP_ICON_BITMAP_RESOURCE RESOURCE_ID="1000">
  <BITMAPS>
    <BITMAP>
      <WIDTH> 22 </WIDTH>
      <HEIGHT> 22 </HEIGHT>
      <BIT_DEPTH> 1 </BIT_DEPTH>
      <ROW_BYTES> 4 </ROW_BYTES>
      <COMPRESSION> NONE </COMPRESSION>
      <BITMAP_DATA>
        FF FF FF FF
        80 00 00 01
        80 00 00 01
        ...
      </BITMAP_DATA>
    </BITMAP>
  </BITMAPS>
</APP_ICON_BITMAP_RESOURCE>
```

Bitmap Data from External Image File

```
<APP_ICON_BITMAP_RESOURCE RESOURCE_ID="1000">
  <BITMAPS>
    <BITMAP>
      <WIDTH> 22 </WIDTH>
      <HEIGHT> 22 </HEIGHT>
      <BIT_DEPTH> 1 </BIT_DEPTH>
      <COMPRESSION> NONE </COMPRESSION>
      <BITMAP_FILE> ". /Images/AppIconLarge_1.bmp"
      </BITMAP_FILE>
    </BITMAP>
    <BITMAP>
      <WIDTH> 22 </WIDTH>
```

XRD Resource Element Reference

Application Icon Bitmap Resource

```
<HEIGHT> 22 </HEIGHT>
<BIT_DEPTH> 2 </BIT_DEPTH>
<COMPRESSION> NONE </COMPRESSION>
  <BITMAP_FILE> " ./Images/AppIconLarge_2.bmp "
  </BITMAP_FILE>
</BITMAP>
</BITMAPS>
</APP_ICON_BITMAP_RESOURCE>
```

High Density Bitmap Elements

```
<APP_ICON_BITMAP_RESOURCE RESOURCE_ID="1000">

  <BITMAPS>

    <BITMAP>
      <WIDTH> 22 </WIDTH>
      <HEIGHT> 22 </HEIGHT>
      <BIT_DEPTH> 1 </BIT_DEPTH>
      <COMPRESSION> NONE </COMPRESSION>
      <BITMAP_FILE> " ./Images/AppIconLarge_1.bmp "
      </BITMAP_FILE>
    </BITMAP>

    <BITMAP>
      <WIDTH> 22 </WIDTH>
      <HEIGHT> 22 </HEIGHT>
      <BIT_DEPTH> 2 </BIT_DEPTH>
      <COMPRESSION> NONE </COMPRESSION>
      <BITMAP_FILE> " ./Images/AppIconLarge_2.bmp "
      </BITMAP_FILE>
    </BITMAP>

    <BITMAP>
      <BITMAP_DENSITY> 144 </BITMAP_DENSITY>
      <WIDTH> 44 </WIDTH>
      <HEIGHT> 44 </HEIGHT>
      <BIT_DEPTH> 1 </BIT_DEPTH>
      <COMPRESSION> NONE </COMPRESSION>
      <BITMAP_FILE> " ./Images/AppIconLarge_X2_1.bmp "
      </BITMAP_FILE>
    </BITMAP>

    <BITMAP>
      <BITMAP_DENSITY> 144 </BITMAP_DENSITY>
      <WIDTH> 44 </WIDTH>
      <HEIGHT> 44 </HEIGHT>
```

```
<BIT_DEPTH> 2 </BIT_DEPTH>
<COMPRESSION> NONE </COMPRESSION>
<BITMAP_FILE> ". /Images/AppIconLarge_X2_2.bmp"
</BITMAP_FILE>
</BITMAP>

</BITMAPS>
</APP_ICON_BITMAP_RESOURCE>
```

Application Icon Name Resource

Purpose	The application icon name resource stores a single string. Palm OS software displays the text in this resource under the application icon. If an application does not have this resource, the application's database name is used.
Target Format	Format generated by PalmRC: <ul style="list-style-type: none">• All OS targets: 'tAIN'
Start Tag	<APP_ICON_NAME_RESOURCE RESOURCE_ID="resource_id">
Child Elements	<TEXT> <i>quoted_text</i> </TEXT> <i>quoted_text</i> - Specifies the text that Palm OS displays under the application icon. If an application does not have this resource, the application's filename is used. This text is a single string, limited to 31-characters long.
End Tag	</APP_ICON_NAME_RESOURCE>
Example	<APP_ICON_NAME_RESOURCE RESOURCE_ID="1000"> <TEXT> "SuperApp" </TEXT> </APP_ICON_NAME_RESOURCE>

Application Information Strings Resource

Purpose	The application information strings resource ("app info") stores a set of category strings. An app info string list resource is often used to hold the application's initial categories in the application's database.
----------------	--

XRD Resource Element Reference

Application Launch Preferences Resource

	Palm OS expects exactly 16 category strings, but this expectation is not enforced by the XML format or by PalmRC.
Target Format	Format generated by PalmRC: <ul style="list-style-type: none">• All OS targets: 'tAIS'
Start Tag	<code><APP_INFO_STRINGS_RESOURCE RESOURCE_ID="resource_id"></code>
Child Elements	<code><STRINGS> text_elements </STRINGS></code> The <code>STRINGS</code> element defines the category strings for the app info strings resource.
End Tag	<code></APP_INFO_STRINGS_RESOURCE></code>
Example	<pre><APP_INFO_STRINGS_RESOURCE RESOURCE_ID="1000"> <STRINGS> <TEXT> "Unfiled" </TEXT> <TEXT> "Business" </TEXT> </STRINGS> </APP_INFO_STRINGS_RESOURCE></pre>

Application Launch Preferences Resource

Purpose	<p>For Palm OS Cobalt, the application launch preference resource replaces the application preferences and application extended preferences resources. For more information about application launch preferences, see <i>Exploring Palm OS: Programming Basics</i>.</p> <p>If you are compiling for an OS target prior to Palm OS Cobalt and you need to disable overlays for an application or a shared library, you will need to use the application extended preferences resource instead. See “Application Extended Preferences Resource” on page 26.</p>
Target Format	Format generated by PalmRC: <ul style="list-style-type: none">• Palm OS 4: Not applicable.• Palm OS 5: Not applicable.• Palm OS 6: 'alpf'

Start Tag <APP_LAUNCH_PREFS_RESOURCE
 RESOURCE_ID="resource_id">
 resource_id - In order to support the Global Find facility,
 make sure this resource ID value is 0. For more information
 about the Global Find facility, see *Exploring Palm OS: Text and
 Localization*.

Child Elements <ALPF_VERSION> *integer* </ALPF_VERSION>
 integer - Specifies the version of this resource. Use the
 integer 1 for Palm OS Cobalt.

 <ALPF_BUILD_OS_VERSION> *quoted_text*
 </ALPF_BUILD_OS_VERSION>
 quoted_text - The version of Palm OS that is the target for
 this application. For Palm OS Cobalt, use
 '6.0.0.RELEASE.0' (with single quotes).

Note: The operating system does not currently use the value
 specified for this element. To prevent incompatibility with
 future releases of the operating system, you should specify
 the value '6.0.0.RELEASE.0' as described above.

 <ALPF_MIN_OS_VERSION> *quoted_text*
 </ALPF_MIN_OS_VERSION>
 quoted_text - The minimum version of Palm OS required
 for this application. For Palm OS Cobalt, use
 '6.0.0.RELEASE.0' (with single quotes).

Note: The operating system does not currently use the value
 specified for this element. To prevent incompatibility with
 future releases of the operating system, you should specify
 the value '6.0.0.RELEASE.0' as described above.

 Instead of checking the value specified by this XRD element,
 the operating system checks either:

- The value specified in the application's SLD file for the
OSVERSION keyword,
- Or the value specified for the PSLib command line
argument -OSversion, which overrides the OSVERSION
keyword in the SLD file.

XRD Resource Element Reference

Application Launch Preferences Resource

<ALPF_STACK_SIZE> *integer* </ALPF_STACK_SIZE>

integer - Specifies the stack space required by the application. To use the default size, use the integer 0.

Note: The operating system does not currently use the value specified for this element. To prevent incompatibility with future releases of the operating system, you should specify the integer value 0 as described above.

<ALPF_FLAGS> *flags* </ALPF_FLAGS>

The *flags* for ALPF_FLAGS are defined by the following child elements.

<ALPF_FLAG_NOTIFY_RESET> *boolean*

</ALPF_FLAG_NOTIFY_RESET>

TRUE - You want Palm OS to notify the application of a device reset.

FALSE - You do not want Palm OS to notify the application of a device reset.

<ALPF_FLAG_NOTIFY_FIND> *boolean*

</ALPF_FLAG_NOTIFY_FIND>

TRUE - You want Palm OS to notify the application of Find search.

FALSE - You do not want Palm OS to notify the application of a Find search.

For more information about the Global Find facility, see *Exploring Palm OS: Text and Localization*.

<ALPF_FLAG_NOTIFY_TIME_CHANGE> *boolean*

</ALPF_FLAG_NOTIFY_TIME_CHANGE>

TRUE - You want Palm OS to notify the application of a time change.

FALSE - You do not want Palm OS to notify the application of a time change.

<ALPF_FLAG_NOTIFY_INSTALL> *boolean*

</ALPF_FLAG_NOTIFY_INSTALL>

TRUE - You want Palm OS to notify the application of a new installation.

FALSE - You do not want Palm OS to notify the application of a new installation.

<ALPF_FLAG_NOTIFY_DB_CHANGE> *boolean*

</ALPF_FLAG_NOTIFY_DB_CHANGE>

TRUE - You want Palm OS to notify the application of a database change.

FALSE - You do not want Palm OS to notify the application of a database change.

<ALPF_FLAG_NO_OVERLAY> *boolean*

</ALPF_FLAG_NO_OVERLAY>

TRUE - This value tells Overlay Manager to ignore any defined overlays for this application when the application is launched.

FALSE - This value tells Overlay Manager to look for the defined overlays for this application when the application is launched.

End Tag </APP_LAUNCH_PREFS_RESOURCE>

Example <APP_LAUNCH_PREFS_RESOURCE RESOURCE_ID="1000">
 <ALPF_VERSION> 1 </ALPF_VERSION>
 <ALPF_BUILD_OS_VERSION>
 '6.0.0.RELEASE.0'
 </ALPF_BUILD_OS_VERSION>
 <ALPF_MIN_OS_VERSION>
 '6.0.0.RELEASE.0'
 </ALPF_MIN_OS_VERSION>
 <ALPF_STACK_SIZE> 0 </ALPF_STACK_SIZE>
 <ALPF_FLAGS>
 <ALPF_FLAG_NOTIFY_RESET>
 FALSE
 </ALPF_FLAG_NOTIFY_RESET>
 <ALPF_FLAG_NOTIFY_FIND>
 FALSE
 </ALPF_FLAG_NOTIFY_FIND>
 <ALPF_FLAG_NOTIFY_TIME_CHANGE>
 FALSE
 </ALPF_FLAG_NOTIFY_TIME_CHANGE>
 <ALPF_FLAG_NOTIFY_INSTALL>
 FALSE
 </ALPF_FLAG_NOTIFY_INSTALL>
 <ALPF_FLAG_NOTIFY_DB_CHANGE>
 FALSE
 </ALPF_FLAG_NOTIFY_DB_CHANGE>
 <ALPF_FLAG_NO_OVERLAY>
 FALSE

XRD Resource Element Reference

Application Preferences Resource

```
</ALPF_FLAG_NO_OVERLAY>
</ALPF_FLAGS>
</APP_LAUNCH_PREFS_RESOURCE>
```

Application Preferences Resource

Purpose The application preferences resource stores information used by the Palm OS. Use this resource if you are compiling for an OS target prior to Palm OS Cobalt. However, for Palm OS Cobalt and later, use the application launch preferences resource instead. See "[Application Launch Preferences Resource](#)" on page 34.

Target Format Format generated by PalmRC:

- Palm OS 4: 'pref'
- Palm OS 5: 'aprf'
- Palm OS 6: 'aprf'

Start Tag <APP_PREFS_RESOURCE RESOURCE_ID="*resource_id*">

Child Elements <PRIORITY> *integer* </PRIORITY>
integer - Specifies the priority for the application.

<STACK_SIZE> *integer* </STACK_SIZE>
integer - Specifies the stack space for the application.

<MIN_HEAP_SPACE> *integer* </MIN_HEAP_SPACE>
integer - Specifies the minimum heap space for the application.

End Tag </APP_PREFS_RESOURCE>

Example <APP_PREFS_RESOURCE RESOURCE_ID="1000">
 <PRIORITY> 30 </PRIORITY>
 <STACK_SIZE> 4096 </STACK_SIZE>
 <MIN_HEAP_SPACE> 4096 </MIN_HEAP_SPACE>
</APP_PREFS_RESOURCE>

Application Version Resource

Purpose	The application version resource stores a single string. Palm OS uses this string to display version information about the application.
Target Format	Format generated by PalmRC: <ul style="list-style-type: none">• All OS targets: 'tver'
Start Tag	<code><APP_VERSION_RESOURCE RESOURCE_ID="resource_id"></code>
Child Elements	<code><TEXT> quoted_text </TEXT></code> <i>quoted_text</i> - Specifies the text that Palm OS displays as the version information for the application.
End Tag	<code></APP_VERSION_RESOURCE></code>
Example	<pre><APP_VERSION_RESOURCE RESOURCE_ID="1000"> <TEXT> "1.0" </TEXT> </APP_VERSION_RESOURCE></pre>

Bitmap Resource

Purpose	<p>The bitmap resource may be used as a form bitmap object, or may be loaded and used with the Palm OS bitmap API.</p> <p>The bitmap stores a set of images which represent the same logical image in one or more image formats. For example, it may provide a monochrome (black-and-white) image, a grayscale image, and/or color versions of the image.</p> <p>The XML format is the same as the application icon bitmap resource.</p>
Target Format	Format generated by PalmRC: <ul style="list-style-type: none">• Palm OS 4: 'Tbmp'• Palm OS 5: 'abmp'• Palm OS 6: 'Tbmp'
Start Tag	<code><BITMAP_RESOURCE RESOURCE_ID="resource_id"></code>
Child Elements	<code><BITMAPS> one or more BITMAP elements </BITMAPS></code> The BITMAPS element consists of a collection of BITMAP elements.

XRD Resource Element Reference

Bitmap Resource

<BITMAP> *bitmap_defn* </BITMAP>

bitmap_defn consists of the following elements:

<BITMAP_DENSITY> *enum* </BITMAP_DENSITY>

This optional element allows you to specify multiple density bitmaps in a bitmap resource. Bitmap elements must be defined in ascending order of density, then in increasing order of bit depth for a given density.

72 - Normal (1X) density (also referred to as single density). This is the default, if this element is omitted.

108 - 1.5X density (also referred to as one-and-one half density).

144 - Double (2X) density.

216 - Triple (3X) density.

The width and height values defined for each bitmap element must match the physical dimensions.

The normalized dimensions for all bitmap elements in a set must match. For example, if the width of a normal density bitmap element is 10, then the one-and-one-half density element(s) must have width 15, and the double density element(s) must have width 20.

<WIDTH> *integer* </WIDTH>

integer - Specifies the width of the bitmap in standard coordinates.

<HEIGHT> *integer* </HEIGHT>

integer - Specifies the height of the bitmap in standard coordinates.

<BIT_DEPTH> *integer* </BIT_DEPTH>

integer - Valid values: 1, 2, 4, 8, and 16.

<ROW_BYTES> *integer* </ROW_BYTES>

This element is optional. Normally, this should be omitted, in which case the correct bytes-per-row of image data is calculated based on the image depth and width. However, to accurately reproduce the data

stored in Palm OS bitmap structures, this field may be explicitly defined.

`<BITMAP_COMPRESSION> enum`

`</BITMAP_COMPRESSION>`

May be specified as enum values NONE, SCANLINE, RLE, PACKBITS, or BEST.

`<HAS_TRANSPARENCY> boolean`

`</HAS_TRANSPARENCY>`

TRUE - Indicates that the bitmap has a transparent color, specified using the TRANSPARENT_INDEX or TRANSPARENT_COLOR element.

FALSE - Indicates that the bitmap does not have a transparent color.

`<TRANSPARENT_INDEX> integer`

`</TRANSPARENT_INDEX>`

Specifies an index number for a color.

`<TRANSPARENT_COLOR> color_value`

`</TRANSPARENT_COLOR>`

Where *color_value* is either an index or a specific red/green/blue color value. To specify an index, use the following:

`<INDEX> integer </INDEX>`

Specifies an index number for a color.

To specify a color value, use the following:

`<RED> integer </RED>`

Specifies the amount of red in the color, in a range from 0x00 (no red) to 0xFF (maximum amount of red).

`<GREEN> integer </GREEN>`

Specifies the amount of green in the color, in a range from 0x00 (no green) to 0xFF (maximum amount of green).

`<BLUE> integer </BLUE>`

Specifies the amount of blue in the color, in a range from 0x00 (no blue) to 0xFF (maximum amount of blue).

XRD Resource Element Reference

Bitmap Resource

`<HAS_COLOR_TABLE> boolean </HAS_COLOR_TABLE>`

TRUE - Indicates that the bitmap has a color table, specified by the COLOR_TABLE element. See "[Color Table Resource](#)" on page 45 for more information.

FALSE - Indicates that the bitmap does not have a color table.

`<BITMAP_DATA> binary_data </BITMAP_DATA>`

The image data of the bitmap may be specified using either a BITMAP_DATA element or a BITMAP_FILE element.

A BITMAP_DATA element specifies the raster image data inline in the XRD file. The image data is specified as binary data. The image data is the uncompressed raster data specified as `rowBytes` times `numScanLines` (`rowBytes * numScanLines`) bytes of data, where the scan lines are stored top to bottom.

`<BITMAP_FILE> ext_file </BITMAP_FILE>`

The image data of the bitmap may be specified using either a BITMAP_DATA element or a BITMAP_FILE element.

A BITMAP_FILE element specifies the image data as the quoted string path of an external image file, which must be in a supported image format such as BMP or PICT. The dimensions of the image in the image file must match exactly the dimensions specified by the BITMAP element.

The image in the image file is automatically bit-depth converted to the bit depth specified by the BITMAP element. For example, if you reference a 24-bit red-green-blue BMP file from a 1-bit bitmap, the image is automatically converted to black and white. Bit depth conversion maps each source pixel to the closest color available in the destination bit depth.

The actual algorithm used for color matching is subject to change. If your application is dependent on specific color use, then make sure you use all of the

specific bit depths required rather than relying on automatic conversion.

When using external image files, you can store the image as a BMP file. For simplicity, you can use BITMAP_FILE paths that specify a path relative to the XRD file.

End Tag `</BITMAP_RESOURCE>`

Example `<BITMAP_RESOURCE RESOURCE_ID="2000">
 <BITMAPS>
 <BITMAP>
 <WIDTH> 148 </WIDTH>
 <HEIGHT> 145 </HEIGHT>
 <BIT_DEPTH> 1 </BIT_DEPTH>
 <BITMAP_COMPRESSION> BEST </BITMAP_COMPRESSION>
 <HAS_TRANSPARENCY> FALSE </HAS_TRANSPARENCY>
 <HAS_COLOR_TABLE> FALSE </HAS_COLOR_TABLE>
 <BITMAP_FILE> ../TestPicts/TestPict1.pict
 </BITMAP>
 </BITMAPS>
</BITMAP_RESOURCE>`

Byte Integer List Resource

Purpose The byte integer list stores zero or more 8-bit unsigned integer numbers.

Target Format Format generated by PalmRC:

- Palm OS 4: 'BLST'
- Palm OS 5: 'abyt'
- Palm OS 6: 'abyt'

Start Tag `<BYTE_LIST_RESOURCE RESOURCE_ID="resource_id">`

Child Elements `<DEFAULT_ITEM> 0 </DEFAULT_ITEM>`
Specifies the value of the default item.

`<VALUES> value_elements </VALUES>`
value_elements consists of one or more VALUE element descriptions.

XRD Resource Element Reference

Code Signing Security Certificate Resource

`<VALUE> integer </VALUE>`

End Tag `</BYTE_LIST_RESOURCE>`

Example `<BYTE_LIST_RESOURCE RESOURCE_ID="1000">
<DEFAULT_ITEM> 0 </DEFAULT_ITEM>
<VALUES>
 <VALUE> 0 </VALUE>
 <VALUE> 0xFF </VALUE>
</VALUES>
</BYTE_LIST_RESOURCE>`

Code Signing Security Certificate Resource

Purpose This code signing security certificate resource represents a certificate to an application.

NOTE: This resource type is used for viewing decompiled resources only. It is not used for creating certificate resources. To create certificate resources, use the tool PRCCert.

Target Format Format generated by PalmRC:

- Palm OS 4: Not applicable.
- Palm OS 5: Not applicable.
- Palm OS 6: 'cert'

Start Tag `<CODE_CERT_RESOURCE RESOURCE_ID="resource_id">`

End Tag `</CODE_CERT_RESOURCE>`

Code Signing Security Signature Resource

Purpose This code signing security signature resource represents a signature to an application.

NOTE: This resource type is used for viewing decompiled resources only. It is not used for creating certificate resources. To create signature resources, use the tool PRCSign.

Target Format	Format generated by PalmRC: <ul style="list-style-type: none"> • Palm OS 4: Not applicable. • Palm OS 5: Not applicable. • Palm OS 6: 'sign'
Start Tag	<code><CODE_SIGN_RESOURCE RESOURCE_ID="resource_id"></code>
End Tag	<code></CODE_SIGN_RESOURCE></code>

Color Table Resource

Purpose	The color table is a list of RGB (red/green/blue) colors.
Target Format	Format generated by PalmRC: <ul style="list-style-type: none"> • Palm OS 4: 'tclt' • Palm OS 5: 'aclt' • Palm OS 6: 'tclt'
Start Tag	<code><COLOR_TABLE_RESOURCE RESOURCE_ID="resource_id"></code>
Child Elements	<code><COLOR_TABLE> color_elements </COLOR_TABLE></code> <i>color_elements</i> consists of one or more COLOR element descriptions. <code><COLOR> color_def </COLOR></code> <i>color_def</i> consists of the following sub-elements: <code><INDEX> integer </INDEX></code> Specifies an index number for a color. <code><RED> integer </RED></code> Specifies the amount of red in the color, in a range from 0x00 (no red) to 0xFF (maximum amount of red). <code><GREEN> integer </GREEN></code> Specifies the amount of green in the color, in a range from 0x00 (no green) to 0xFF (maximum amount of green).

XRD Resource Element Reference

DWord Integer List Resource

`<BLUE> integer </BLUE>`

Specifies the amount of blue in the color, in a range from 0x00 (no blue) to 0xFF (maximum amount of blue).

End Tag `</COLOR_TABLE_RESOURCE>`

Example `<COLOR_TABLE_RESOURCE RESOURCE_ID="10002">
<COLOR_TABLE>
 <COLOR>
 <INDEX> 0x00 </INDEX>
 <RED> 0xFF </RED>
 <GREEN> 0xFF </GREEN>
 <BLUE> 0xFF </BLUE>
 </COLOR>
 <COLOR>
 <INDEX> 0x01 </INDEX>
 <RED> 0xAA </RED>
 <GREEN> 0xAA </GREEN>
 <BLUE> 0xAA </BLUE>
 </COLOR>
 <COLOR>
 <INDEX> 0x02 </INDEX>
 <RED> 0x55 </RED>
 <GREEN> 0x55 </GREEN>
 <BLUE> 0x55 </BLUE>
 </COLOR>
 <COLOR>
 <INDEX> 0x03 </INDEX>
 <RED> 0x00 </RED>
 <GREEN> 0x00 </GREEN>
 <BLUE> 0x00 </BLUE>
 </COLOR>
</COLOR_TABLE>
</COLOR_TABLE_RESOURCE>`

DWord Integer List Resource

Purpose The dword integer list stores zero or more 32-bit unsigned integer numbers.

Target Format Format generated by PalmRC:

- Palm OS 4: 'DLST'
- Palm OS 5: 'adwd'

	<ul style="list-style-type: none"> • Palm OS 6: 'adwd'
Start Tag	<code><DWORD_LIST_RESOURCE RESOURCE_ID="resource_id"></code>
Child Elements	<code><DEFAULT_ITEM> integer </DEFAULT_ITEM></code> <i>Specifies the value of the default item.</i> <code><VALUES> value_elements </VALUES></code> <i>value_elements consists of one or more VALUE element descriptions.</i> <code><VALUE> integer </VALUE></code>
End Tag	<code></DWORD_LIST_RESOURCE></code>
Example	<pre> <DWORD_LIST_RESOURCE RESOURCE_ID="1000"> <DEFAULT_ITEM> 0 </DEFAULT_ITEM> <VALUES> <VALUE> 0 </VALUE> <VALUE> 0xFFFFFFFF </VALUE> </VALUES> </DWORD_LIST_RESOURCE> </pre>

Extended Font Resource

Purpose	The extended font resource.
Target Format	Format generated by PalmRC: <ul style="list-style-type: none"> • Palm OS 4: 'nfnt' • Palm OS 5: 'afnx' • Palm OS 6: 'nfnt'
Start Tag	<code><FONT_EXTENDED_RESOURCE RESOURCE_ID="resource_id"></code>
Child Elements	<code><FONT_TYPE> integer </FONT_TYPE></code> <i>integer - A hex value that identifies this font type.</i> <code><FONT_ASCENT> integer </FONT_ASCENT></code> <i>integer - A nonnegative integer specifying the pixel amount that this font extends above the baseline. (The font ascent plus the font descent equals the font height.)</i> <code><FONT_DESCENT> integer </FONT_DESCENT></code> <i>integer - A nonnegative integer specifying the pixel amount that this font extends below the baseline. (The font ascent plus the font descent equals the font height.)</i>

XRD Resource Element Reference

Extended Font Resource

```
<FONT_EXTENDED_ITEMS> font_extended_items
  </FONT_EXTENDED_ITEMS>
    where font_extended_items consists of one or more
    <FONT_EXTENDED_ITEM> elements.

<FONT_EXTENDED_ITEM> font_extened_item_defn
  </FONT_EXTENDED_ITEM>
    where each font_extened_item_defn consists of the
    following child elements:

    <FONT_DENSITY> integer </FONT_DENSITY>
      integer - The font density. Use 72 for a low density
      font, use 108 for one-and-one-half density font, and
      use 144 for a double density font.

    <FONT_GLYPHS> font_glyph_items </FONT_GLYPHS>
      where each font_glyph_item consists of the
      following child elements:

      <FONT_GLYPH>

        <FONT_GLYPH_CODE> integer
        </FONT_GLYPH_CODE>
          integer - A hex value that identifies this font glyph.

        <FONT_GLYPH_IMAGE> glyph_image
        </FONT_GLYPH_IMAGE>
          glyph_image - Image data. Each row of the bitmap is
          specified as a quoted string. (The number of rows in
          the glyph must equal the font height.) See the example
          below for a sample glyph_image.

      </FONT_GLYPH>

      <FONT_MISSING_GLYPH>
        <FONT_GLYPH_IMAGE> glyph_image
        </FONT_GLYPH_IMAGE>
          glyph_image - Image data for font items not
          specified by an integer hex value. Each row of the
          bitmap is specified as a quoted string. (The number of
          rows in the glyph must equal the font height.) See the
          example below for a sample glyph_image.

      </FONT_MISSING_GLYPH>

  End Tag </FONT_EXTENDED_RESOURCE>
```

Example

```

<FONT_EXTENDED_RESOURCE RESOURCE_ID="1000">
  <FONT_TYPE> 0x9200 </FONT_TYPE>
  <FONT_ASCENT> 8 </FONT_ASCENT>
  <FONT_DESCENT> 0 </FONT_DESCENT>
  <FONT_EXTENDED_ITEMS>
    <FONT_EXTENDED_ITEM>
      <FONT_DENSITY> 72 </FONT_DENSITY>
      <FONT_GLYPHS>
        <FONT_GLYPH>
          <FONT_GLYPH_CODE> 0x0001 </FONT_GLYPH_CODE>
          <FONT_GLYPH_IMAGE>
            " ..... "
            " .....#..... "
            " .....###..... "
            " ...#####... "
            " ..#####.. "
            " .#####. "
            " ##### "
            " ..... "
          </FONT_GLYPH_IMAGE>
        </FONT_GLYPH>
        <FONT_GLYPH>
          <FONT_GLYPH_CODE> 0x0002 </FONT_GLYPH_CODE>
          <FONT_GLYPH_IMAGE>
            " ..... "
            " ##### "
            " .#####. "
            " ..#####.. "
            " ...#####... "
            " .....###..... "
            " .....#..... "
            " ..... "
          </FONT_GLYPH_IMAGE>
        </FONT_GLYPH>
      </FONT_GLYPHS>
      <FONT_MISSING_GLYPH>
        <FONT_GLYPH_IMAGE>
          " ..... "
          " ##### "
          " #...# "
          " #...# "
          " #...# "
          " #...# "
          " ##### "
          " ..... "
        </FONT_GLYPH_IMAGE>
      </FONT_MISSING_GLYPH>
    </FONT_EXTENDED_ITEM>
  </FONT_EXTENDED_ITEMS>
</FONT_EXTENDED_RESOURCE>

```

Extended Font Resource

50 *Palm OS Resource File Formats*

```

" ..... "
" ..... "
"#####"
"#####"
"##.....#"
"##.....#"
"##.....#"
"##.....#"
"##.....#"
"##.....#"
"##.....#"
"##.....#"
"#####"
"#####"
" ..... "
" ..... "
</FONT_GLYPH_IMAGE>
</FONT_MISSING_GLYPH>
</FONT_EXTENDED_ITEM>
</FONT_EXTENDED_ITEMS>
</FONT_EXTENDED_RESOURCE>

```

Font Resource

Purpose	The font resource represents a bitmap font, specifying each glyph in the font as a bitmap.
Target Format	Format generated by PalmRC: <ul style="list-style-type: none"> • Palm OS 4: 'NFNT' • Palm OS 5: 'afnt ' • Palm OS 6: 'NFNT '
Start Tag	<FONT_RESOURCE RESOURCE_ID=" <i>resource_id</i> ">
Child Elements	<FONT_TYPE> <i>integer</i> </FONT_TYPE> <i>integer</i> - A hex value that identifies this font type. <FONT_ASCENT> <i>integer</i> </FONT_ASCENT> <i>integer</i> - A nonnegative integer specifying the pixel amount that this font extends above the baseline. (The font ascent plus the font descent equals the font height.)

XRD Resource Element Reference

Font Resource

`<FONT_DESCENT> integer </FONT_DESCENT>`
integer - A nonnegative integer specifying the pixel amount that this font extends below the baseline. (The font ascent plus the font descent equals the font height.)

`<FONT_GLYPHS> font_glyph_items </FONT_GLYPHS>`
where each *font_glyph_item* consists of the following child elements:

`<FONT_GLYPH>`
`<FONT_GLYPH_CODE> integer`
`</FONT_GLYPH_CODE>`
integer - A hex value that identifies this font glyph.
`<FONT_GLYPH_IMAGE> glyph_image`
`</FONT_GLYPH_IMAGE>`
glyph_image - Image data. Each row of the bitmap is specified as a quoted string. (The number of rows in the glyph must equal the font height.) See the example below for a sample *glyph_image*.

`</FONT_GLYPH>`

`<FONT_MISSING_GLYPH> font_missing_glyph_items`
`</FONT_MISSING_GLYPH>`
where each *font_missing_glyph_item* consists of the following child elements:

`<FONT_GLYPH_IMAGE> glyph_image`
`</FONT_GLYPH_IMAGE>`
glyph_image - Image data for font items not specified by an integer hex value. Each row of the bitmap is specified as a quoted string. (The number of rows in the glyph must equal the font height.) See the example below for a sample *glyph_image*.

End Tag `</FONT_RESOURCE>`

Example

```
<FONT_RESOURCE RESOURCE_ID="1000">
  <FONT_TYPE> 0x9000 </FONT_TYPE>
  <FONT_ASCENT> 9 </FONT_ASCENT>
  <FONT_DESCENT> 1 </FONT_DESCENT>
  <FONT_GLYPHS>
    <FONT_GLYPH>
      <FONT_GLYPH_CODE> 0x0003 </FONT_GLYPH_CODE>
      <FONT_GLYPH_IMAGE>
```

```

        " . . . . ## . . . . "
        " . . . . ### . . . . "
        " . . ##### . # . "
        " . ##### . # . "
        " ##### . # . "
        " ##### . # . "
        " . ##### . # . "
        " . . ##### . # . "
        " . . . . ### . . . . "
        " . . . . ## . . . . "
        </FONT_GLYPH_IMAGE>
    </FONT_GLYPH>
<FONT_GLYPH>
    <FONT_GLYPH_CODE> 0x0004 </FONT_GLYPH_CODE>
    <FONT_GLYPH_IMAGE>
        " . . . . ## . . . . "
        " . . . . ### . . . . "
        " # . ##### . . "
        " # . ##### . "
        " # . ##### "
        " # . ##### "
        " # . ##### "
        " # . ##### . "
        " # . ##### . . "
        " . . . . ### . . . . "
        " . . . . ## . . . . "
        </FONT_GLYPH_IMAGE>
    </FONT_GLYPH>
</FONT_GLYPHS>
<FONT_MISSING_GLYPH>
    <FONT_GLYPH_IMAGE>
        " . . . . . "
        " ##### "
        " # . . . # "
        " # . . . # "
        " # . . . # "
        " # . . . # "
        " # . . . # "
        " # . . . # "
        " ##### "
        " . . . . . "
        </FONT_GLYPH_IMAGE>
    </FONT_MISSING_GLYPH>
</FONT_RESOURCE>

```

Form Resource

Purpose	The form resource stores a form description which contains a window description and a list of form object descriptions (for example, user interface controls such as buttons and lists).
Target Format	Format generated by PalmRC: <ul style="list-style-type: none">• Palm OS 4: 'tFRM'• Palm OS 5: 'afrm'• Palm OS 6: 'tFRM'
Start Tag	<code><FORM_RESOURCE RESOURCE_ID="resource_id"></code>
Child Elements	<code><FORM_ID> resource_id </FORM_ID></code> Specifies the identifier for the form. This must match the resource ID attribute on the FORM_RESOURCE tag. <code><BOUNDS> bounds_defn </BOUNDS></code> <i>bounds_defn</i> consists of the following child elements: <code><LEFT> integer </LEFT></code> Specifies, in standard coordinates, the left-most edge of the form. <code><TOP> integer </TOP></code> Specifies, in standard coordinates, the top-most edge of the form. <code><WIDTH> integer </WIDTH></code> Specifies, in standard coordinates, the width of the form. <code><HEIGHT> integer </HEIGHT></code> Specifies, in standard coordinates, the height of the form. <code><USABLE> boolean </USABLE></code> Specifies the visibility of the form, either visible or invisible. TRUE - the form is displayed in the Palm OS application FALSE - the form is not displayed.

`<MODAL> boolean </MODAL>`

Specifies how events outside of the form boundary are interpreted; that is, if a user must interact with the form before continuing with other tasks.

TRUE - the form does not respond to taps outside the form.

FALSE - the form does respond to taps outside the form.

`<SAVE_BEHIND> boolean </SAVE_BEHIND>`

Specifies how the area behind the form appears on the screen.

TRUE - Palm OS saves the area of the screen the form obscures; then restores the area when the form is hidden.

FALSE - Palm OS does not save the area of the screen the form obscures.

NOTE: The `SAVE_BEHIND` element is included for 68K application compatibility use. For ARM-native applications, you should never set the `SAVE_BEHIND` element's value to TRUE.

`<HELP_ID> integer </HELP_ID>`

Specifies the help text resource ID that appears when the user taps the Help icon in a modal form. (The help ID is only applicable if you specify TRUE for the `MODAL` element.)

`<MENU_ID> integer </MENU_ID>`

Specifies the resource ID of the menu bar associated with the form.

`<DEFAULT_BUTTON> integer </DEFAULT_BUTTON>`

Specifies the button index of the form's default button. Palm OS software simulates the tapping of this button when the form is dismissed.

`<TITLE_ICON> boolean </TITLE_ICON>`

Indicates whether this form has a title icon.

`<TITLE_BAR_FOCUSABLE> boolean`

`</TITLE_BAR_FOCUSABLE>`

Indicates whether the title bar can be used as a tab stop in the tab order for the Form Navigation resource. For more information, see "[Form Navigation Resource](#)" on page 58.

XRD Resource Element Reference

Form Resource

<FORM_OBJECTS> *form_objects* </FORM_OBJECTS>
form_objects - One or more of the following form object descriptions.

FORM_TITLE

See “[Form Title Object](#)” on page 96.

FORM_BUTTON

See “[Form Button Object](#)” on page 63.

FORM_LABEL

See “[Form Label Object](#)” on page 80.

FORM_CHECKBOX

See “[Form Check Box Object](#)” on page 65.

FORM_FIELD

See “[Form Field Object](#)” on page 69.

FORM_GRAPHIC_BUTTON

See “[Form Graphic Button Object](#)” on page 74.

FORM_PUSH_BUTTON

See “[Form Push Button Object](#)” on page 85.

FORM_GRAPHIC_PUSH_BUTTON

See “[Form Graphic Push Button Object](#)” on page 76.

FORM_POPUP_TRIGGER

See “[Form Pop-Up Trigger Object](#)” on page 83.

FORM_SELECTOR_TRIGGER

See “[Form Selector Trigger Object](#)” on page 91.

FORM_REPEATING_BUTTON

See “[Form Repeating Button Object](#)” on page 87.

FORM_GRAPHIC_REPEATING_BUTTON

See “[Form Graphic Repeating Button Object](#)” on page 78.

FORM_SLIDER

See “[Form Slider Object](#)” on page 93.

FORM_FEEDBACK_SLIDER

See “[Form Feedback Slider Object](#)” on page 67.

FORM_LIST

See “[Form List Object](#)” on page 81.

FORM_TABLE

See “[Form Table Object](#)” on page 95.

FORM_BITMAP

See “[Form Bitmap Object](#)” on page 62.

FORM_POPUP

See “[Form Pop-Up Object](#)” on page 83.

FORM_GRAFFITI_STATE

See “[Form Graffiti State Object](#)” on page 74.

FORM_GADGET

See “[Form Gadget Object](#)” on page 72.

FORM_SCROLLBAR

See “[Form Scroll Bar Object](#)” on page 89.

End Tag </FORM_RESOURCE>

Example

```
<FORM_RESOURCE RESOURCE_ID="1100">
  <FORM_ID> 1100 </FORM_ID>
  <BOUNDS>
    <LEFT> 2 </LEFT>
    <TOP> 2 </TOP>
    <WIDTH> 156 </WIDTH>
    <HEIGHT> 156 </HEIGHT>
  </BOUNDS>
  <USABLE> TRUE </USABLE>
  <MODAL> TRUE </MODAL>
  <SAVE_BEHIND> TRUE </SAVE_BEHIND>
  <HELP_ID> 0 </HELP_ID>
  <MENU_ID> 0 </MENU_ID>
  <DEFAULT_BUTTON> 0 </DEFAULT_BUTTON>
  <FORM_OBJECTS>
    <FORM_TITLE>
      <TEXT> "About Starter" </TEXT>
    </FORM_TITLE>
    <FORM_LABEL>
      <ID> 1104 </ID>
      <LOCATION>
        <X> 50 </X>
        <Y> 104 </Y>
      </LOCATION>
      <USABLE> TRUE </USABLE>
      <FONT_ID> BOLD_FONT </FONT_ID>
      <TEXT> "Version 1.0" </TEXT>
    </FORM_LABEL>
    <FORM_BUTTON>
```

XRD Resource Element Reference

Form Navigation Resource

```
<ID> 1105 </ID>
<BOUNDS>
  <LEFT> 58 </LEFT>
  <TOP> 133 </TOP>
  <WIDTH> 40 </WIDTH>
  <HEIGHT> 12 </HEIGHT>
</BOUNDS>
<USABLE> TRUE </USABLE>
<ENABLED> TRUE </ENABLED>
<TEXT> "OK" </TEXT>
<LEFT_ANCHOR> TRUE </LEFT_ANCHOR>
<FONT_ID> STD_FONT </FONT_ID>
<BUTTON_FRAME> STANDARD_BUTTON_FRAME </BUTTON_FRAME>
</FORM_BUTTON>
</FORM_OBJECTS>
</FORM_RESOURCE>
```

Form Navigation Resource

Purpose Used in one-handed navigation, the form navigation resource specifies relative position of the elements in a FORM_RESOURCE. To enable left-right navigation through the objects on a form, this resource indicates the tab order. To enable up-down navigation, this resource specifies, for each object, the object lying directly above and the object lying directly below.

You can create a form navigation resource for each form in the application.

Target Format Format generated by PalmRC:

- Palm OS 4: Not applicable.
- Palm OS 5: 'fnav'
- Palm OS 6: 'fnav'

Start Tag <FORM_NAVIGATION_RESOURCE
RESOURCE_ID="resource_id">
resource_id- The resource ID value must match the Form Resource's ID value.

Child Elements <VERSION>1</VERSION>
The only valid value for this element is the number 1.

`<OBJECT_FOCUS> boolean </OBJECT_FOCUS>`

Mutually exclusive value with the APP_FOCUS element. (If OBJECT_FOCUS is TRUE, then APP_FOCUS must be FALSE.)

If true, the form should open initially in object focus mode—that is, with the navigation highlight on one of its elements. Specify which element to highlight using the INITIAL_OBJECT_ID element.

`<APP_FOCUS> boolean </APP_FOCUS>`

Mutually exclusive value with the OBJECT_FOCUS element. (If APP_FOCUS is TRUE, then OBJECT_FOCUS must be FALSE.)

If true, the form should initially open in application focus mode—that is, with no navigation highlight.

`<INITIAL_OBJECT_ID> resource_id`

`</INITIAL_OBJECT_ID>`

resource_id - Must be either 0 or the ID of one of the objects in the form.

`<JUMP_OBJECT_ID> resource_id </JUMP_OBJECT_ID>`

resource_id - Must be either 0 or the ID of one of the objects in the form.

`<BOTTOMLEFT_OBJECT_ID> resource_id`

`</BOTTOMLEFT_OBJECT_ID>`

resource_id - Must be either 0 or the ID of one of the objects in the form.

`<TITLE_BAR_FOCUSABLE> boolean`

`</TITLE_BAR_FOCUSABLE>`

Indicates whether to allow the tab order to focus on the title bar for menu bar access.

`<NAVIGATION_FORM_OBJECTS> nav_form_objects`

`</NAVIGATION_FORM_OBJECTS>`

nav_form_objects consists of one or more NAVIGATION_FORM_OBJECT elements.

The order of the elements within the NAVIGATION_FORM_OBJECTS element dictates the tab order of the form.

XRD Resource Element Reference

Form Navigation Resource

<NAVIGATION_FORM_OBJECT>

Information about each navigatable element within a form. If there are no navigatable elements, the NAVIGATION_FORM_OBJECTS element can be empty.

<OBJECT_ID> *integer* </OBJECT_ID>

The ID of the form object for this position in the tab order. The ID should match an object in the form resource.

<ABOVE_OBJECT_ID> *integer* </ABOVE_OBJECT_ID>

The ID of the form object when the user presses the UP button. Valid values are any object ID in the form or 0.

<BELOW_OBJECT_ID> *integer* </BELOW_OBJECT_ID>

The ID of the form object when the user presses the DOWN button. Valid values are any object ID in the form or 0.

<SKIP_OBJECT> *boolean* </SKIP_OBJECT>

Indicates whether to skip the object in the tab order.

TRUE - Skip this object in the tab order.

FALSE - Include this object in the tab order.

<FORCE_INTERACTION> *boolean*

</FORCE_INTERACTION>

Used for multi-line edit fields. Sets the focus to the field.

<BIG_BUTTON> *boolean* </BIG_BUTTON>

Optional value for applications targeted for Handspring devices. Sets big mode for button controls.

</NAVIGATION_FORM_OBJECT>

End Tag

</FORM_NAVIGATION_RESOURCE>

Example

```
<FORM_NAVIGATION_RESOURCE RESOURCE_ID="1000"
  COMMENT="An example of a form navigation resource">
```

```
  <VERSION>1</VERSION>
```

```
  <OBJECT_FOCUS> TRUE </OBJECT_FOCUS>
```

```
  <APP_FOCUS> FALSE </APP_FOCUS>
```

```
  <INITIAL_OBJECT_ID> 0 </INITIAL_OBJECT_ID>
```

```
  <JUMP_OBJECT_ID> 0 </JUMP_OBJECT_ID>
```



```
<BOTTOMLEFT_OBJECT_ID> 0 </BOTTOMLEFT_OBJECT_ID>
<TITLE_BAR_FOCUSABLE>TRUE</TITLE_BAR_FOCUSABLE>

<NAVIGATION_FORM_OBJECTS>

  <NAVIGATION_FORM_OBJECT COMMENT="First object">
    <OBJECT_ID> 1001 </OBJECT_ID>
    <ABOVE_OBJECT_ID> 0 </ABOVE_OBJECT_ID>
    <BELOW_OBJECT_ID> 1002 </BELOW_OBJECT_ID>
    <SKIP_OBJECT> TRUE </SKIP_OBJECT>
    <FORCE_INTERACTION> TRUE </FORCE_INTERACTION>
  </NAVIGATION_FORM_OBJECT>

  <NAVIGATION_FORM_OBJECT COMMENT="Second object">
    <OBJECT_ID> 1002 </OBJECT_ID>
    <ABOVE_OBJECT_ID> 1001 </ABOVE_OBJECT_ID>
    <BELOW_OBJECT_ID> 0 </BELOW_OBJECT_ID>
    <SKIP_OBJECT> TRUE </SKIP_OBJECT>
    <FORCE_INTERACTION> TRUE </FORCE_INTERACTION>
    <BIG_BUTTON> TRUE </BIG_BUTTON>
  </NAVIGATION_FORM_OBJECT>

</NAVIGATION_FORM_OBJECTS>

</FORM_NAVIGATION_RESOURCE>
```

Form Object Descriptions

Form resources can use any of the following form object descriptions, listed here in alphabetical order.

Form Bitmap Object 62
Form Button Object 63
Form Check Box Object 65
Form Feedback Slider Object 67
Form Field Object 69
Form Gadget Object 72
Form Graffiti State Object 74
Form Graphic Button Object 74
Form Graphic Push Button Object 76
Form Graphic Repeating Button Object 78

XRD Resource Element Reference

Form Navigation Resource

Form Label Object	80
Form List Object	81
Form Pop-Up Object	83
Form Pop-Up Trigger Object	83
Form Push Button Object	85
Form Repeating Button Object	87
Form Scroll Bar Object	89
Form Selector Trigger Object	91
Form Slider Object	93
Form Table Object	95
Form Title Object	96

Form Bitmap Object

Purpose	The form resource may have one or more bitmap objects as items in the FORM_OBJECTS element.
Start Tag	<FORM_BITMAP>
Child Elements	<p><USABLE> <i>boolean</i> </USABLE> Specifies the visibility of the object, either visible or invisible. TRUE - the object is displayed in the Palm OS application FALSE - the object is not displayed.</p> <p><LOCATION> <i>location_defn</i> </LOCATION> <i>location_defn</i> consists of the following child elements:</p> <p><X> <i>integer</i> </X> Specifies the left origin (form-relative position of the left side of the object).</p> <p><Y> <i>integer</i> </Y> Specifies the top origin (form-relative position of the top of the object).</p> <p><BITMAP_ID> <i>integer</i> </BITMAP_ID></p>
End Tag	</FORM_BITMAP>
Example	<pre><FORM_BITMAP> <USABLE> TRUE </USABLE> <LOCATION></pre>

```
<X> 23 </X>
<Y> 16 </Y>
</LOCATION>
<BITMAP_ID> 1000 </BITMAP_ID>
</FORM_BITMAP>
```

Form Button Object

Purpose	The form resource may have one or more button objects as items in the FORM_OBJECTS element.
Start Tag	<FORM_BUTTON>
Child Elements	<p><ID> <i>integer</i> </ID> Specifies the object identifier.</p> <p><BOUNDS> <i>bounds_defn</i> </BOUNDS> <i>bounds_defn</i> consists of the following child elements:</p> <p><LEFT> <i>integer</i> </LEFT> Specifies, in standard coordinates, the left-most edge of the form object.</p> <p><TOP> <i>integer</i> </TOP> Specifies, in standard coordinates, the top-most edge of the form object.</p> <p><WIDTH> <i>integer</i> </WIDTH> Specifies, in standard coordinates, the width of the form object.</p> <p><HEIGHT> <i>integer</i> </HEIGHT> Specifies, in standard coordinates, the height of the form object.</p> <p><USABLE> <i>boolean</i> </USABLE> Specifies the visibility of the object, either visible or invisible. TRUE - the object is displayed in the Palm OS application FALSE - the object is not displayed.</p> <p><ENABLED> <i>boolean</i> </ENABLED> Specifies whether the object responds to user interface events. TRUE - the object responds to user interface events. FALSE - the object does not respond to user interface events.</p>

XRD Resource Element Reference

Form Navigation Resource

`<TEXT> quoted_text </TEXT>`

Specifies the text displayed inside the button.

`<LEFT_ANCHOR> boolean </LEFT_ANCHOR>`

Controls how the object resizes itself.

TRUE - the object's left bound is fixed.

FALSE - the object's right bound is fixed.

`<FONT_ID> enum </FONT_ID>`

The FONT_ID element is used to specify the font used by the form object. The element may be specified as an integer value between 0 and 255, or as enum values listed below:

STD_FONT

BOLD_FONT

LARGE_FONT

SYMBOL_FONT

SYMBOL_11_FONT

SYMBOL_7_FONT

LED_FONT

LARGE_BOLD_FONT

`<BUTTON_FRAME> enum </BUTTON_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

End Tag `</FORM_BUTTON>`

Example

```
<FORM_BUTTON>
  <ID> 1002 </ID>
  <BOUNDS>
    <LEFT> 117 </LEFT>
    <TOP> 80 </TOP>
    <WIDTH> 36 </WIDTH>
    <HEIGHT> 12 </HEIGHT>
```

```

</BOUNDS>
<USABLE> TRUE </USABLE>
<ENABLED> TRUE </ENABLED>
<TEXT> "Go" </TEXT>
<LEFT_ANCHOR> TRUE </LEFT_ANCHOR>
<FONT_ID> STD_FONT </FONT_ID>
<BUTTON_FRAME> STANDARD_BUTTON_FRAME </BUTTON_FRAME>
</FORM_BUTTON>

```

Form Check Box Object

Purpose	The form resource may have one or more check box objects as items in the FORM_OBJECTS element.
Start Tag	<FORM_CHECKBOX>
Child Elements	<p><ID> <i>integer</i> </ID> Specifies the object identifier.</p> <p><BOUNDS> <i>bounds_defn</i> </BOUNDS> <i>bounds_defn</i> consists of the following child elements:</p> <p><LEFT> <i>integer</i> </LEFT> Specifies, in standard coordinates, the left-most edge of the form object.</p> <p><TOP> <i>integer</i> </TOP> Specifies, in standard coordinates, the top-most edge of the form object.</p> <p><WIDTH> <i>integer</i> </WIDTH> Specifies, in standard coordinates, the width of the form object.</p> <p><HEIGHT> <i>integer</i> </HEIGHT> Specifies, in standard coordinates, the height of the form object.</p> <p><USABLE> <i>boolean</i> </USABLE> Specifies the visibility of the object, either visible or invisible. TRUE - the object is displayed in the Palm OS application FALSE - the object is not displayed.</p>

XRD Resource Element Reference

Form Navigation Resource

`<ENABLED> boolean </ENABLED>`

Specifies whether the object responds to user interface events.

TRUE - the object responds to user interface events.

FALSE - the object does not respond to user interface events.

`<TEXT> quoted_text </TEXT>`

Specifies the text displayed to the right of the check box.

`<FONT_ID> enum </FONT_ID>`

The FONT_ID element is used to specify the font used by the form object. The element may be specified as an integer value between 0 and 255, or as enum values listed below:

STD_FONT

BOLD_FONT

LARGE_FONT

SYMBOL_FONT

SYMBOL_11_FONT

SYMBOL_7_FONT

LED_FONT

LARGE_BOLD_FONT

`<GROUP_ID> integer </GROUP_ID>`

Specifies a value that identifies a group of controls. To indicate that the object is not part of a group, specify the value 0.

Note: For check boxes, you should set this value to 0.

`<SELECTED> boolean </SELECTED>`

Specifies the initial selection state of the object.

TRUE - The object is initially selected.

FALSE - The object is initially not selected.

End Tag `</FORM_CHECKBOX>`

Example

```
<FORM_CHECKBOX>
  <ID> 1002 </ID>
  <BOUNDS>
    <LEFT> 56 </LEFT>
    <TOP> 19 </TOP>
```

```
<WIDTH> 65 </WIDTH>
<HEIGHT> 12 </HEIGHT>
</BOUNDS>
<USABLE> TRUE </USABLE>
<ENABLED> TRUE </ENABLED>
<TEXT> "checkbox" </TEXT>
<FONT_ID> BOLD_FONT </FONT_ID>
<GROUP_ID> 0 </GROUP_ID>
<SELECTED> TRUE </SELECTED>
</FORM_CHECKBOX>
```

Form Feedback Slider Object

Purpose	The form resource may have one or more feedback slider objects as items in the FORM_OBJECTS element.
Start Tag	<FORM_FEEDBACK_SLIDER>
Child Elements	<p><ID> <i>integer</i> </ID> Specifies the object identifier.</p> <p><BOUNDS> <i>bounds_defn</i> </BOUNDS> <i>bounds_defn</i> consists of the following child elements:</p> <p><LEFT> <i>integer</i> </LEFT> Specifies, in standard coordinates, the left-most edge of the form object.</p> <p><TOP> <i>integer</i> </TOP> Specifies, in standard coordinates, the top-most edge of the form object.</p> <p><WIDTH> <i>integer</i> </WIDTH> Specifies, in standard coordinates, the width of the form object.</p> <p><HEIGHT> <i>integer</i> </HEIGHT> Specifies, in standard coordinates, the height of the form object.</p> <p><USABLE> <i>boolean</i> </USABLE> Specifies the visibility of the object, either visible or invisible. TRUE - the object is displayed in the Palm OS application FALSE - the object is not displayed.</p>

XRD Resource Element Reference

Form Navigation Resource

`<ENABLED> boolean </ENABLED>`

Specifies whether the object responds to user interface events.

TRUE - the object responds to user interface events.

FALSE - the object does not respond to user interface events.

`<VALUE> integer </VALUE>`

The initial value for the slider.

`<MIN_VALUE> integer </MIN_VALUE>`

The minimum value the slider can represent.

`<MAX_VALUE> integer </MAX_VALUE>`

The maximum value the slider can represent.

`<PAGE_SIZE> integer </PAGE_SIZE>`

The amount by which the slider thumb moves when the user taps to the thumb's right or left.

`<THUMB_BITMAP_ID> integer </THUMB_BITMAP_ID>`

The resource ID of the bitmap resource or bitmap family resource used to create the slider thumb. Use 0 to specify the default thumb bitmap.

`<BACKGROUND_BITMAP_ID> integer`

`</BACKGROUND_BITMAP_ID>`

The resource ID of the bitmap resource or bitmap family resource used to create the background for the slider. Use 0 to specify the default background bitmap.

End Tag `</FORM_FEEDBACK_SLIDER>`

Example

```
<FORM_FEEDBACK_SLIDER>
  <ID> 1802 </ID>
  <BOUNDS>
    <LEFT> 21 </LEFT>
    <TOP> 14 </TOP>
    <WIDTH> 114 </WIDTH>
    <HEIGHT> 15 </HEIGHT>
  </BOUNDS>
  <USABLE> TRUE </USABLE>
  <ENABLED> TRUE </ENABLED>
  <VALUE> 0 </VALUE>
  <MIN_VALUE> 0 </MIN_VALUE>
  <MAX_VALUE> 150 </MAX_VALUE>
  <PAGE_SIZE> 10 </PAGE_SIZE>
  <THUMB_BITMAP_ID> 4000 </THUMB_BITMAP_ID>
```

```
<BACKGROUND_BITMAP_ID> 4001 </BACKGROUND_BITMAP_ID>
</FORM_FEEDBACK_SLIDER>
```

Form Field Object

Purpose	The form resource may have one or more field objects as items in the FORM_OBJECTS element.
Start Tag	<FORM_FIELD>
Child Elements	<p><ID> <i>integer</i> </ID> Specifies the object identifier.</p> <p><BOUNDS> <i>bounds_defn</i> </BOUNDS> <i>bounds_defn</i> consists of the following child elements:</p> <p><LEFT> <i>integer</i> </LEFT> Specifies, in standard coordinates, the left-most edge of the form object.</p> <p><TOP> <i>integer</i> </TOP> Specifies, in standard coordinates, the top-most edge of the form object.</p> <p><WIDTH> <i>integer</i> </WIDTH> Specifies, in standard coordinates, the width of the form object.</p> <p><HEIGHT> <i>integer</i> </HEIGHT> Specifies, in standard coordinates, the height of the form object.</p> <p><USABLE> <i>boolean</i> </USABLE> Specifies the visibility of the object, either visible or invisible. TRUE - the object is displayed in the Palm OS application. FALSE - the object is not displayed.</p> <p><EDITABLE> <i>boolean</i> </EDITABLE> Specifies whether the field can be changed by the user. TRUE - the field is editable. FALSE - the field is not editable. Note that noneditable fields don't accept user input, but they can be changed programmatically.</p>

XRD Resource Element Reference

Form Navigation Resource

`<SINGLE_LINE> boolean </SINGLE_LINE>`

Specifies whether the field scrolls horizontally and accepts RETURN or TAB characters.

TRUE - the field does not scroll horizontally and it does not accept RETURN or TAB characters.

FALSE - the field does scroll horizontally and it does accept RETURN or TAB characters.

`<DYNAMIC_SIZE> boolean </DYNAMIC_SIZE>`

Specifies whether the field grows as the user enters more text.

TRUE - the field grows as the user enters more text.

FALSE - the field remains the same size as when it was originally created.

`<UNDERLINE> enum </UNDERLINE>`

The UNDERLINE 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_UNDERLINE

GRAY_UNDERLINE

SOLID_UNDERLINE

COLOR_UNDERLINE

`<JUSTIFICATION> enum </JUSTIFICATION>`

Specifies the justification setting for the field.

LEFT_ALIGN

Text is aligned on the left. You should use LEFT_ALIGN for editable single-line text fields.

RIGHT_ALIGN

Text is aligned on the right. You should use RIGHT_ALIGN for single-line numeric fields or noneditable fields used as labels.

`<AUTO_SHIFT> boolean </AUTO_SHIFT>`

Specifies whether the auto-shift rules apply for this field.

TRUE - the auto-shift rules apply. You should use this setting for most editable fields.

FALSE - the auto-shift rules do not apply.

`<HAS_SCROLLBAR> boolean </HAS_SCROLLBAR>`

Specifies whether you want to associate a scroll bar with a multi-line field.

TRUE - you want a scroll bar for this field.

FALSE - you don't want a scroll bar for this field.

`<NUMERIC> boolean </NUMERIC>`

Specifies whether this field is limited to numeric input.

TRUE - the field will allow the user to enter numbers only (the characters 0 through 9). 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. Note that numeric fields do not allow plus signs.

FALSE - the field allows all character data.

`<MAX_CHARS> integer </MAX_CHARS>`

Specifies the 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. This attribute has no effect on noneditable fields.

`<FONT_ID> enum </FONT_ID>`

The FONT_ID element is used to specify the font used by the form object. The element may be specified as an integer value between 0 and 255, or as enum values listed below:

STD_FONT

BOLD_FONT

LARGE_FONT

SYMBOL_FONT

SYMBOL_11_FONT

SYMBOL_7_FONT

LED_FONT

LARGE_BOLD_FONT

`<MAX_VISIBLE_LINES> integer </MAX_VISIBLE_LINES>`

Specifies the maximum number of lines visible for a multi-line entry field.

XRD Resource Element Reference

Form Navigation Resource

End Tag `</FORM_FIELD>`

Example

```
<FORM_FIELD>
  <ID> 1004 </ID>
  <BOUNDS>
    <LEFT> 4 </LEFT>
    <TOP> 53 </TOP>
    <WIDTH> 50 </WIDTH>
    <HEIGHT> 12 </HEIGHT>
  </BOUNDS>
  <USABLE> TRUE </USABLE>
  <EDITABLE> TRUE </EDITABLE>
  <SINGLE_LINE> FALSE </SINGLE_LINE>
  <DYNAMIC_SIZE> FALSE </DYNAMIC_SIZE>
  <UNDERLINE> GRAY_UNDERLINE </UNDERLINE>
  <JUSTIFICATION> LEFT_ALIGN </JUSTIFICATION>
  <AUTO_SHIFT> TRUE </AUTO_SHIFT>
  <HAS_SCROLLBAR> FALSE </HAS_SCROLLBAR>
  <NUMERIC> FALSE </NUMERIC>
  <MAX_CHARS> 80 </MAX_CHARS>
  <FONT_ID> LARGE_FONT </FONT_ID>
  <MAX_VISIBLE_LINES> 0 </MAX_VISIBLE_LINES>
</FORM_FIELD>
```

Form Gadget Object

Purpose The form resource may have one or more gadget objects as items in the FORM_OBJECTS element.

Start Tag `<FORM_GADGET>`

Child Elements

```
<ID> integer </ID>
  Specifies the object identifier.

<BOUNDS> bounds_defn </BOUNDS>
  bounds_defn consists of the following child elements:

    <LEFT> integer </LEFT>
      Specifies, in standard coordinates, the left-most edge
      of the form object.

    <TOP> integer </TOP>
      Specifies, in standard coordinates, the top-most edge
      of the form object.

    <WIDTH> integer </WIDTH>
      Specifies, in standard coordinates, the width of the
      form object.
```

`<HEIGHT> integer </HEIGHT>`

Specifies, in standard coordinates, the height of the form object.

`<USABLE> boolean </USABLE>`

Specifies the visibility of the object, either visible or invisible.

TRUE - The object is displayed in the Palm OS application.

FALSE - The object is not displayed.

`<EXTENDED> boolean </EXTENDED>`

Specifies whether the gadget is an extended gadget. If set, the gadget is an extended gadget.

TRUE - The gadget is an extended gadget. An extended gadget has the handler field defined in its `FormGadgetType`. For more information about extended gadgets, see the book *Exploring Palm OS: User Interface*.

FALSE - The gadget is a standard gadget compatible with all releases of Palm OS.

`<VISIBLE> boolean </VISIBLE>`

This value is set to TRUE when the gadget is drawn, and set to FALSE when the gadget is erased. `FrmHideObject()` 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.

End Tag

`</FORM_GADGET>`

Example

```
<FORM_GADGET>
  <ID> 1207 </ID>
  <BOUNDS>
    <LEFT> 6 </LEFT>
    <TOP> 70 </TOP>
    <WIDTH> 16 </WIDTH>
    <HEIGHT> 16 </HEIGHT>
  </BOUNDS>
  <USABLE> FALSE </USABLE>
  <EXTENDED> TRUE </EXTENDED>
  <VISIBLE> FALSE </VISIBLE>
</FORM_GADGET>
```

XRD Resource Element Reference

Form Navigation Resource

Form Graffiti State Object

Purpose	The form resource may have one graffiti state shift indicator object as items in the FORM_OBJECTS element.
Start Tag	<FORM_GRAFFITI_STATE>
Child Elements	<p><LOCATION> <i>location_defn</i> </LOCATION> <i>location_defn</i> consists of the following child elements:</p> <p><X> <i>integer</i> </X> Specifies the left origin (form-relative position of the left side of the object).</p> <p><Y> <i>integer</i> </Y> Specifies the top origin (form-relative position of the top of the object).</p>
End Tag	</FORM_GRAFFITI_STATE>
Example	<pre><FORM_GRAFFITI_STATE> <LOCATION> <X> 146 </X> <Y> 23 </Y> </LOCATION> </FORM_GRAFFITI_STATE></pre>

Form Graphic Button Object

Purpose	The form resource may have one or more graphic button objects as items in the FORM_OBJECTS element.
Start Tag	<FORM_GRAPHIC_BUTTON>
Child Elements	<p><ID> <i>integer</i> </ID> Specifies the object identifier.</p> <p><BOUNDS> <i>bounds_defn</i> </BOUNDS> <i>bounds_defn</i> consists of the following child elements:</p> <p><LEFT> <i>integer</i> </LEFT> Specifies, in standard coordinates, the left-most edge of the form object.</p> <p><TOP> <i>integer</i> </TOP> Specifies, in standard coordinates, the top-most edge of the form object.</p>

`<WIDTH> integer </WIDTH>`

Specifies, in standard coordinates, the width of the form object.

`<HEIGHT> integer </HEIGHT>`

Specifies, in standard coordinates, the height of the form object.

`<USABLE> boolean </USABLE>`

Specifies the visibility of the object, either visible or invisible.

TRUE - the object is displayed in the Palm OS application

FALSE - the object is not displayed.

`<ENABLED> boolean </ENABLED>`

Specifies whether the object responds to user interface events.

TRUE - the object responds to user interface events.

FALSE - the object does not respond to user interface events.

`<LEFT_ANCHOR> boolean </LEFT_ANCHOR>`

Controls how the object resizes itself.

TRUE - the object's left bound is fixed.

FALSE - the object's right bound is fixed.

`<BUTTON_FRAME> enum </BUTTON_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

`<BITMAP_ID> integer </BITMAP_ID>`

Specifies the resource ID of the bitmap or bitmap family that draws the graphic on this button.

`<SELECTED_BITMAP_ID> integer </SELECTED_BITMAP_ID>`

Resource ID of the bitmap or bitmap family that draws the graphic when the button is selected. It is important to

XRD Resource Element Reference

Form Navigation Resource

provide a different graphic for the selected bitmap on a color screen.

End Tag `</FORM_GRAPHIC_BUTTON>`

Example `<FORM_GRAPHIC_BUTTON>
<ID> 1202 </ID>
<BOUNDS>
 <LEFT> 8 </LEFT>
 <TOP> 8 </TOP>
 <WIDTH> 40 </WIDTH>
 <HEIGHT> 40 </HEIGHT>
</BOUNDS>
<USABLE> TRUE </USABLE>
<ENABLED> TRUE </ENABLED>
<LEFT_ANCHOR> TRUE </LEFT_ANCHOR>
<BUTTON_FRAME> NO_BUTTON_FRAME </BUTTON_FRAME>
<BITMAP_ID> 1000 </BITMAP_ID>
<SELECTED_BITMAP_ID> 1001 </SELECTED_BITMAP_ID>
</FORM_GRAPHIC_BUTTON>`

Form Graphic Push Button Object

Purpose The form resource may have one or more graphic push button objects as items in the FORM_OBJECTS element.

Start Tag `<FORM_GRAPHIC_PUSH_BUTTON>`

Child Elements `<ID> integer </ID>`
Specifies the object identifier.

`<BOUNDS> bounds_defn </BOUNDS>`
bounds_defn consists of the following child elements:

`<LEFT> integer </LEFT>`
Specifies, in standard coordinates, the left-most edge of the form object.

`<TOP> integer </TOP>`
Specifies, in standard coordinates, the top-most edge of the form object.

`<WIDTH> integer </WIDTH>`
Specifies, in standard coordinates, the width of the form object.

`<HEIGHT> integer </HEIGHT>`

Specifies, in standard coordinates, the height of the form object.

`<USABLE> boolean </USABLE>`

Specifies the visibility of the object, either visible or invisible.

TRUE - the object is displayed in the Palm OS application

FALSE - the object is not displayed.

`<ENABLED> boolean </ENABLED>`

Specifies whether the object responds to user interface events.

TRUE - the object responds to user interface events.

FALSE - the object does not respond to user interface events.

`<BITMAP_ID> integer </BITMAP_ID>`

Specifies the resource ID of the bitmap or bitmap family that draws the graphic on this button.

`<SELECTED_BITMAP_ID> integer </SELECTED_BITMAP_ID>`

Resource ID of the bitmap or bitmap family that draws the graphic when the button is selected. It is important to provide a different graphic for the selected bitmap on a color screen.

`<GROUP_ID> integer </GROUP_ID>`

Specifies a value that identifies a group of controls. To indicate that the object is not part of a group, specify the value 0.

End Tag

`</FORM_GRAPHIC_PUSH_BUTTON>`

Example

`<FORM_GRAPHIC_PUSH_BUTTON>`

`<ID> 1302 </ID>`

`<BOUNDS>`

`<LEFT> 17 </LEFT>`

`<TOP> 14 </TOP>`

`<WIDTH> 20 </WIDTH>`

`<HEIGHT> 20 </HEIGHT>`

`</BOUNDS>`

`<USABLE> TRUE </USABLE>`

`<ENABLED> TRUE </ENABLED>`

`<BITMAP_ID> 2000 </BITMAP_ID>`

`<SELECTED_BITMAP_ID> 2001 </SELECTED_BITMAP_ID>`

XRD Resource Element Reference

Form Navigation Resource

```
<GROUP_ID> 0 </GROUP_ID>  
</FORM_GRAPHIC_PUSH_BUTTON>
```

Form Graphic Repeating Button Object

Purpose The form resource may have one or more graphic repeating button objects as items in the FORM_OBJECTS element.

Start Tag <FORM_GRAPHIC_REPEATING_BUTTON>

Child Elements

```
<ID> integer </ID>  
    Specifies the object identifier.
```

```
<BOUNDS> bounds_defn </BOUNDS>  
    bounds_defn consists of the following child elements:
```

```
<LEFT> integer </LEFT>  
    Specifies, in standard coordinates, the left-most edge  
    of the form object.
```

```
<TOP> integer </TOP>  
    Specifies, in standard coordinates, the top-most edge  
    of the form object.
```

```
<WIDTH> integer </WIDTH>  
    Specifies, in standard coordinates, the width of the  
    form object.
```

```
<HEIGHT> integer </HEIGHT>  
    Specifies, in standard coordinates, the height of the  
    form object.
```

```
<USABLE> boolean </USABLE>  
    Specifies the visibility of the object, either visible or invisible.  
    TRUE - the object is displayed in the Palm OS application  
    FALSE - the object is not displayed.
```

```
<ENABLED> boolean </ENABLED>  
    Specifies whether the object responds to user interface  
    events.  
    TRUE - the object responds to user interface events.  
    FALSE - the object does not respond to user interface events.
```

`<LEFT_ANCHOR> boolean </LEFT_ANCHOR>`

Controls how the object resizes itself.

TRUE - the object's left bound is fixed.

FALSE - the object's right bound is fixed.

`<BUTTON_FRAME> enum </BUTTON_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

`<BITMAP_ID> integer </BITMAP_ID>`

Specifies the resource ID of the bitmap or bitmap family that draws the graphic on this button.

`<SELECTED_BITMAP_ID> integer </SELECTED_BITMAP_ID>`

Resource ID of the bitmap or bitmap family that draws the graphic when the button is selected. It is important to provide a different graphic for the selected bitmap on a color screen.

End Tag

`</FORM_GRAPHIC_REPEATING_BUTTON>`

Example

```
<FORM_GRAPHIC_REPEATING_BUTTON>
  <ID> 1402 </ID>
  <BOUNDS>
    <LEFT> 6 </LEFT>
    <TOP> 7 </TOP>
    <WIDTH> 30 </WIDTH>
    <HEIGHT> 30 </HEIGHT>
  </BOUNDS>
  <USABLE> TRUE </USABLE>
  <ENABLED> TRUE </ENABLED>
  <LEFT_ANCHOR> TRUE </LEFT_ANCHOR>
  <BUTTON_FRAME> NO_BUTTON_FRAME </BUTTON_FRAME>
  <BITMAP_ID> 1000 </BITMAP_ID>
  <SELECTED_BITMAP_ID> 1001 </SELECTED_BITMAP_ID>
</FORM_GRAPHIC_REPEATING_BUTTON>
```

XRD Resource Element Reference

Form Navigation Resource

Form Label Object

Purpose	The form resource may have one or more label objects as items in the FORM_OBJECTS element.
Start Tag	<FORM_LABEL>
Child Elements	<p><ID> <i>integer</i> </ID> Specifies the object identifier.</p> <p><LOCATION> <i>location_defn</i> </LOCATION> <i>location_defn</i> consists of the following child elements:</p> <p><X> <i>integer</i> </X> Specifies the left origin (form-relative position of the left side of the object).</p> <p><Y> <i>integer</i> </Y> Specifies the top origin (form-relative position of the top of the object).</p> <p><USABLE> <i>boolean</i> </USABLE> Specifies the visibility of the object, either visible or invisible. TRUE - The object is displayed in the Palm OS application. FALSE - The object is not displayed.</p> <p><FONT_ID> <i>enum</i> </FONT_ID> The FONT_ID element is used to specify the font used by the form object. The element may be specified as an integer value between 0 and 255, or as enum values listed below:</p> <p>STD_FONT</p> <p>BOLD_FONT</p> <p>LARGE_FONT</p> <p>SYMBOL_FONT</p> <p>SYMBOL_11_FONT</p> <p>SYMBOL_7_FONT</p> <p>LED_FONT</p> <p>LARGE_BOLD_FONT</p> <p><TEXT> <i>quoted_text</i> </TEXT> Specifies the text of the label.</p>

End Tag `</FORM_LABEL>`

Example

```
<FORM_LABEL>
  <ID> 1104 </ID>
  <LOCATION>
    <X> 50 </X>
    <Y> 104 </Y>
  </LOCATION>
  <USABLE> TRUE </USABLE>
  <FONT_ID> BOLD_FONT </FONT_ID>
  <TEXT> "Version 1.0" </TEXT>
</FORM_LABEL>
```

Form List Object

Purpose The form resource may have one or more list objects as items in the FORM_OBJECTS element.

Start Tag `<FORM_LIST>`

Child Elements

```
<ID> integer </ID>
  Specifies the object identifier.

<BOUNDS> bounds_defn </BOUNDS>
  bounds_defn consists of the following child elements:

    <LEFT> integer </LEFT>
      Specifies, in standard coordinates, the left-most edge
      of the form object.

    <TOP> integer </TOP>
      Specifies, in standard coordinates, the top-most edge
      of the form object.

    <WIDTH> integer </WIDTH>
      Specifies, in standard coordinates, the width of the
      form object.

    <HEIGHT> integer </HEIGHT>
      Specifies, in standard coordinates, the height of the
      form object.

<USABLE> boolean </USABLE>
  Specifies the visibility of the object, either visible or invisible.
  TRUE - the object is displayed in the Palm OS application
  FALSE - the object is not displayed.
```

XRD Resource Element Reference

Form Navigation Resource

`<FONT_ID> enum </FONT_ID>`

The FONT_ID element is used to specify the font used by the form object. The element may be specified as an integer value between 0 and 255, or as enum values listed below:

STD_FONT

BOLD_FONT

LARGE_FONT

SYMBOL_FONT

SYMBOL_11_FONT

SYMBOL_7_FONT

LED_FONT

LARGE_BOLD_FONT

`<NUM_VIS_ITEMS> integer </NUM_VIS_ITEMS>`

Specifies the number of items the list displays, generally one of these values:

13 - For a full screen pop-up list.

11 - For a full screen stand-alone list (modeless form).

10 - For a full screen stand-alone list (modal form).

0 - For the Categories pop-up list (the actual size is determined at runtime).

Set to the total number of items in your list if less than the specific numbers listed, or if you don't want the **Categories** pop-up list.

`<LIST_ITEMS> text_elements </LIST_ITEMS>`

Specifies the text for the items you want in the list. Leave this element out if you want to add list items programmatically.

End Tag `</FORM_LIST>`

Example

```
<FORM_LIST>
  <ID> 1013 </ID>
  <BOUNDS>
    <LEFT> 29 </LEFT>
    <TOP> 96 </TOP>
    <WIDTH> 30 </WIDTH>
    <HEIGHT> 33 </HEIGHT>
  </BOUNDS>
```

```
<USABLE> TRUE </USABLE>
<FONT_ID> STD_FONT </FONT_ID>
<NUM_VIS_ITEMS> 3 </NUM_VIS_ITEMS>
<LIST_ITEMS>
  <TEXT> "Sun" </TEXT>
  <TEXT> "Mon" </TEXT>
  <TEXT> "Tue" </TEXT>
</LIST_ITEMS>
</FORM_LIST>
```

Form Pop-Up Object

Purpose	The form resource may have one or more pop-up objects as items in the FORM_OBJECTS element.
Start Tag	<FORM_POPUP>
Child Elements	<p><CONTROL_ID> <i>integer</i> </CONTROL_ID> The pop-up object's CONTROL_ID value matches the corresponding pop-up trigger's ID value. See "Form Pop-Up Trigger Object."</p> <p><LIST_ID> <i>integer</i> </LIST_ID> The pop-up object's LIST_ID value matches the ID value for the list object that you want to pop-up. See "Form List Object" on page 81.</p>
End Tag	</FORM_POPUP>
Example	<pre><FORM_POPUP> <CONTROL_ID> 1314 </CONTROL_ID> <LIST_ID> 1013 </LIST_ID> </FORM_POPUP></pre>

Form Pop-Up Trigger Object

Purpose	The form resource may have one or more pop-up trigger objects as items in the FORM_OBJECTS element.
Start Tag	<FORM_POPUP_TRIGGER>
Child Elements	<p><ID> <i>integer</i> </ID> Specifies the object identifier. The pop-up trigger's ID value matches the corresponding pop-up object's CONTROL_ID value. See "Form Pop-Up Object" on page 83.</p>

XRD Resource Element Reference

Form Navigation Resource

<BOUNDS> *bounds_defn* </BOUNDS>

bounds_defn consists of the following child elements:

<LEFT> *integer* </LEFT>

Specifies, in standard coordinates, the left-most edge of the form object.

<TOP> *integer* </TOP>

Specifies, in standard coordinates, the top-most edge of the form object.

<WIDTH> *integer* </WIDTH>

Specifies, in standard coordinates, the width of the form object.

<HEIGHT> *integer* </HEIGHT>

Specifies, in standard coordinates, the height of the form object.

<USABLE> *boolean* </USABLE>

Specifies the visibility of the object, either visible or invisible.

TRUE - the object is displayed in the Palm OS application

FALSE - the object is not displayed.

<ENABLED> *boolean* </ENABLED>

Specifies whether the object responds to user interface events.

TRUE - the object responds to user interface events.

FALSE - the object does not respond to user interface events.

<LEFT_ANCHOR> *boolean* </LEFT_ANCHOR>

Controls how the object resizes itself.

TRUE - the object's left bound is fixed.

FALSE - the object's right bound is fixed.

<TEXT> *quoted_text* </TEXT>

Specifies the text for the pop-up label. Usually, you leave this field blank and set the label programmatically.

<FONT_ID> *enum* </FONT_ID>

The FONT_ID element is used to specify the font used by the form object. The element may be specified as an integer value between 0 and 255, or as enum values listed below:

STD_FONT
 BOLD_FONT
 LARGE_FONT
 SYMBOL_FONT
 SYMBOL_11_FONT
 SYMBOL_7_FONT
 LED_FONT
 LARGE_BOLD_FONT

End Tag </FORM_POPUP_TRIGGER>

Example <FORM_POPUP_TRIGGER>
 <ID> 1314 </ID>
 <BOUNDS>
 <LEFT> 111 </LEFT>
 <TOP> 61 </TOP>
 <WIDTH> 42 </WIDTH>
 <HEIGHT> 12 </HEIGHT>
 </BOUNDS>
 <USABLE> FALSE </USABLE>
 <ENABLED> TRUE </ENABLED>
 <LEFT_ANCHOR> TRUE </LEFT_ANCHOR>
 <TEXT> "Pop-up" </TEXT>
 <FONT_ID> STD_FONT </FONT_ID>
 </FORM_POPUP_TRIGGER>

Form Push Button Object

Purpose The Form resource may have one or more push button objects as items in the FORM_OBJECTS element.

Start Tag <FORM_PUSH_BUTTON>

Child Elements <ID> *integer* </ID>
 Specifies the object identifier.

<BOUNDS> *bounds_defn* </BOUNDS>
 bounds_defn consists of the following child elements:

<LEFT> *integer* </LEFT>
 Specifies, in standard coordinates, the left-most edge of the form object.

XRD Resource Element Reference

Form Navigation Resource

`<TOP> integer </TOP>`

Specifies, in standard coordinates, the top-most edge of the form object.

`<WIDTH> integer </WIDTH>`

Specifies, in standard coordinates, the width of the form object.

`<HEIGHT> integer </HEIGHT>`

Specifies, in standard coordinates, the height of the form object.

`<USABLE> boolean </USABLE>`

Specifies the visibility of the object, either visible or invisible.

TRUE - the object is displayed in the Palm OS application

FALSE - the object is not displayed.

`<ENABLED> boolean </ENABLED>`

Specifies whether the object responds to user interface events.

TRUE - the object responds to user interface events.

FALSE - the object does not respond to user interface events.

`<TEXT> quoted_text </TEXT>`

Specifies the text displayed inside the push button.

`<FONT_ID> enum </FONT_ID>`

Specifies the font used by the form object. The element may be specified as an integer value between 0 and 255, or as enum values listed below:

STD_FONT

BOLD_FONT

LARGE_FONT

SYMBOL_FONT

SYMBOL_11_FONT

SYMBOL_7_FONT

LED_FONT

LARGE_BOLD_FONT

`<GROUP_ID> integer </GROUP_ID>`

Specifies a value that identifies a group of controls. To indicate that the object is not part of a group, specify the value 0.

End Tag `</FORM_PUSH_BUTTON>`

Example

```
<FORM_PUSH_BUTTON>
  <ID> 1015 </ID>
  <BOUNDS>
    <LEFT> 71 </LEFT>
    <TOP> 101 </TOP>
    <WIDTH> 12 </WIDTH>
    <HEIGHT> 12 </HEIGHT>
  </BOUNDS>
  <USABLE> TRUE </USABLE>
  <ENABLED> TRUE </ENABLED>
  <TEXT> "A" </TEXT>
  <FONT_ID> STD_FONT </FONT_ID>
  <GROUP_ID> 0 </GROUP_ID>
</FORM_PUSH_BUTTON>
```

Form Repeating Button Object

Purpose The form resource may have one or more repeating button objects as items in the FORM_OBJECTS element.

Start Tag `<FORM_REPEATING_BUTTON>`

Child Elements `<ID> integer </ID>`
Specifies the object identifier.

`<BOUNDS> bounds_defn </BOUNDS>`

bounds_defn consists of the following child elements:

`<LEFT> integer </LEFT>`

Specifies, in standard coordinates, the left-most edge of the form object.

`<TOP> integer </TOP>`

Specifies, in standard coordinates, the top-most edge of the form object.

`<WIDTH> integer </WIDTH>`

Specifies, in standard coordinates, the width of the form object.

XRD Resource Element Reference

Form Navigation Resource

`<HEIGHT> integer </HEIGHT>`

Specifies, in standard coordinates, the height of the form object.

`<USABLE> boolean </USABLE>`

Specifies the visibility of the object, either visible or invisible.

TRUE - the object is displayed in the Palm OS application

FALSE - the object is not displayed.

`<ENABLED> boolean </ENABLED>`

Specifies whether the object responds to user interface events.

TRUE - the object responds to user interface events.

FALSE - the object does not respond to user interface events.

`<LEFT_ANCHOR> boolean </LEFT_ANCHOR>`

Controls how the object resizes itself.

TRUE - the object's left bound is fixed.

FALSE - the object's right bound is fixed.

`<TEXT> quoted_text </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_ID> enum </FONT_ID>`

The FONT_ID element is used to specify the font used by the form object. The element may be specified as an integer value between 0 and 255, or as enum values listed below:

STD_FONT

BOLD_FONT

LARGE_FONT

SYMBOL_FONT

SYMBOL_11_FONT

SYMBOL_7_FONT

LED_FONT

LARGE_BOLD_FONT

<BUTTON_FRAME> *enum* </BUTTON_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

End Tag </FORM_REPEATING_BUTTON>

Example

```
<FORM_REPEATING_BUTTON>
  <ID> 1316 </ID>
  <BOUNDS>
    <LEFT> 118 </LEFT>
    <TOP> 78 </TOP>
    <WIDTH> 36 </WIDTH>
    <HEIGHT> 12 </HEIGHT>
  </BOUNDS>
  <USABLE> FALSE </USABLE>
  <ENABLED> TRUE </ENABLED>
  <TEXT> "Rep" </TEXT>
  <LEFT_ANCHOR> TRUE </LEFT_ANCHOR>
  <FONT_ID> STD_FONT </FONT_ID>
  <BUTTON_FRAME> NO_BUTTON_FRAME </BUTTON_FRAME>
</FORM_REPEATING_BUTTON>
```

Form Scroll Bar Object

Purpose The form resource may have one or more scroll bar objects as items in the FORM_OBJECTS element.

Start Tag <FORM_SCROLLBAR>

Child Elements

<ID> *integer* </ID>

Specifies the object identifier.

<BOUNDS> *bounds_defn* </BOUNDS>

bounds_defn consists of the following child elements:

<LEFT> *integer* </LEFT>

Specifies, in standard coordinates, the left-most edge of the form object.

XRD Resource Element Reference

Form Navigation Resource

`<TOP> integer </TOP>`

Specifies, in standard coordinates, the top-most edge of the form object.

`<WIDTH> integer </WIDTH>`

Specifies, in standard coordinates, the width of the form object.

`<HEIGHT> integer </HEIGHT>`

Specifies, in standard coordinates, the height of the form object.

`<USABLE> boolean </USABLE>`

Specifies the visibility of the object, either visible or invisible.

TRUE - the object is displayed in the Palm OS application

FALSE - the object is not displayed.

`<VALUE> integer </VALUE>`

The initial setting for the scroll bar.

`<MIN_VALUE> integer </MIN_VALUE>`

The minimum value the scroll bar can represent.

`<MAX_VALUE> integer </MAX_VALUE>`

The maximum value the scroll bar can represent.

`<PAGE_SIZE> integer </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.

End Tag

`</FORM_SCROLLBAR>`

Example

```
<FORM_SCROLLBAR>
  <ID> 1217 </ID>
  <BOUNDS>
    <LEFT> 147 </LEFT>
    <TOP> 115 </TOP>
    <WIDTH> 7 </WIDTH>
    <HEIGHT> 40 </HEIGHT>
  </BOUNDS>
  <USABLE> FALSE </USABLE>
  <VALUE> 33 </VALUE>
  <MIN_VALUE> 0 </MIN_VALUE>
  <MAX_VALUE> 100 </MAX_VALUE>
```

```
<PAGE_SIZE> 11 </PAGE_SIZE>
</FORM_SCROLLBAR>
```

Form Selector Trigger Object

Purpose	The form resource may have one or more selector trigger objects as items in the FORM_OBJECTS element.
Start Tag	<code><FORM_SELECTOR_TRIGGER></code>
Child Elements	<p><code><ID> <i>integer</i> </ID></code> Specifies the object identifier.</p> <p><code><BOUNDS> <i>bounds_defn</i> </BOUNDS></code> <i>bounds_defn</i> consists of the following child elements:</p> <p><code><LEFT> <i>integer</i> </LEFT></code> Specifies, in standard coordinates, the left-most edge of the form object.</p> <p><code><TOP> <i>integer</i> </TOP></code> Specifies, in standard coordinates, the top-most edge of the form object.</p> <p><code><WIDTH> <i>integer</i> </WIDTH></code> Specifies, in standard coordinates, the width of the form object.</p> <p><code><HEIGHT> <i>integer</i> </HEIGHT></code> Specifies, in standard coordinates, the height of the form object.</p> <p><code><USABLE> <i>boolean</i> </USABLE></code> Specifies the visibility of the object, either visible or invisible. TRUE - the object is displayed in the Palm OS application FALSE - the object is not displayed.</p> <p><code><ENABLED> <i>boolean</i> </ENABLED></code> Specifies whether the object responds to user interface events. TRUE - the object responds to user interface events. FALSE - the object does not respond to user interface events.</p>

XRD Resource Element Reference

Form Navigation Resource

`<LEFT_ANCHOR> boolean </LEFT_ANCHOR>`

Controls how the object resizes itself.

TRUE - the object's left bound is fixed.

FALSE - the object's right bound is fixed.

`<TEXT> quoted_text </TEXT>`

Specifies the text for the initial value of the pop-up trigger. You can leave this element blank and specify the text at runtime.

`<FONT_ID> enum </FONT_ID>`

The FONT_ID element is used to specify the font used by the form object. The element may be specified as an integer value between 0 and 255, or as enum values listed below:

STD_FONT

BOLD_FONT

LARGE_FONT

SYMBOL_FONT

SYMBOL_11_FONT

SYMBOL_7_FONT

LED_FONT

LARGE_BOLD_FONT

End Tag `</FORM_SELECTOR_TRIGGER>`

Example

```
<FORM_SELECTOR_TRIGGER>
  <ID> 1018 </ID>
  <BOUNDS>
    <LEFT> 11 </LEFT>
    <TOP> 135 </TOP>
    <WIDTH> 39 </WIDTH>
    <HEIGHT> 12 </HEIGHT>
  </BOUNDS>
  <USABLE> TRUE </USABLE>
  <ENABLED> TRUE </ENABLED>
  <LEFT_ANCHOR> TRUE </LEFT_ANCHOR>
  <TEXT> "Selector" </TEXT>
  <FONT_ID> STD_FONT </FONT_ID>
</FORM_SELECTOR_TRIGGER>
```

Form Slider Object

Purpose	The form resource may have one or more slider objects as items in the FORM_OBJECTS element.
Start Tag	<FORM_SLIDER>
Child Elements	<p><ID> <i>integer</i> </ID> Specifies the object identifier.</p> <p><BOUNDS> <i>bounds_defn</i> </BOUNDS> <i>bounds_defn</i> consists of the following child elements:</p> <p><LEFT> <i>integer</i> </LEFT> Specifies, in standard coordinates, the left-most edge of the form object.</p> <p><TOP> <i>integer</i> </TOP> Specifies, in standard coordinates, the top-most edge of the form object.</p> <p><WIDTH> <i>integer</i> </WIDTH> Specifies, in standard coordinates, the width of the form object.</p> <p><HEIGHT> <i>integer</i> </HEIGHT> Specifies, in standard coordinates, the height of the form object.</p> <p><USABLE> <i>boolean</i> </USABLE> Specifies the visibility of the object, either visible or invisible. TRUE - the object is displayed in the Palm OS application FALSE - the object is not displayed.</p> <p><ENABLED> <i>boolean</i> </ENABLED> Specifies whether the object responds to user interface events. TRUE - the object responds to user interface events. FALSE - the object does not respond to user interface events.</p> <p><VALUE> <i>integer</i> </VALUE> The initial setting for the slider.</p> <p><MIN_VALUE> <i>integer</i> </MIN_VALUE> The minimum value the slider can represent.</p>

XRD Resource Element Reference

Form Navigation Resource

`<MAX_VALUE> integer </MAX_VALUE>`

The maximum value the slider can represent.

`<PAGE_SIZE> integer </PAGE_SIZE>`

The amount by which the slider thumb moves when the user taps to the thumb's right or left.

`<THUMB_BITMAP_ID> integer </THUMB_BITMAP_ID>`

The resource ID of the bitmap resource or bitmap family resource used to create the slider thumb. Use 0 to specify the default thumb bitmap.

`<BACKGROUND_BITMAP_ID> integer
</BACKGROUND_BITMAP_ID>`

The resource ID of the bitmap resource or bitmap family resource used to create the background for the slider. Use 0 to specify the default background bitmap.

End Tag

`</FORM_SLIDER>`

Example

```
<FORM_SLIDER>
  <ID> 1702 </ID>
  <BOUNDS>
    <LEFT> 24 </LEFT>
    <TOP> 12 </TOP>
    <WIDTH> 114 </WIDTH>
    <HEIGHT> 15 </HEIGHT>
  </BOUNDS>
  <USABLE> TRUE </USABLE>
  <ENABLED> TRUE </ENABLED>
  <VALUE> 0 </VALUE>
  <MIN_VALUE> 0 </MIN_VALUE>
  <MAX_VALUE> 150 </MAX_VALUE>
  <PAGE_SIZE> 10 </PAGE_SIZE>
  <THUMB_BITMAP_ID> 4000 </THUMB_BITMAP_ID>
  <BACKGROUND_BITMAP_ID> 4001 </BACKGROUND_BITMAP_ID>
</FORM_SLIDER>
```

Form Table Object

Purpose	The form resource may have one or more table objects as items in the FORM_OBJECTS element.
Start Tag	<FORM_TABLE>
Child Elements	<p><ID> <i>integer</i> </ID> Specifies the object identifier.</p> <p><BOUNDS> <i>bounds_defn</i> </BOUNDS> <i>bounds_defn</i> consists of the following child elements:</p> <p><LEFT> <i>integer</i> </LEFT> Specifies, in standard coordinates, the left-most edge of the form object.</p> <p><TOP> <i>integer</i> </TOP> Specifies, in standard coordinates, the top-most edge of the form object.</p> <p><WIDTH> <i>integer</i> </WIDTH> Specifies, in standard coordinates, the width of the form object.</p> <p><HEIGHT> <i>integer</i> </HEIGHT> Specifies, in standard coordinates, the height of the form object.</p> <p><EDITABLE> <i>boolean</i> </EDITABLE> Specifies whether the user can modify the table. TRUE - the user can modify the table. FALSE - the user cannot modify the table.</p> <p><TABLE_COLUMNS> <i>column_entries</i> </TABLE_COLUMNS> <i>column_entries</i> consists of a <COLUMN_WIDTH> element for each column in the table, which specify each column's width in standard coordinates. The sum of the widths must equal the table width. The number of <COLUMN_WIDTH> elements determines the number of columns in the table.</p>

XRD Resource Element Reference

Form Navigation Resource

`<TABLE_ROWS> row_entries </TABLE_ROWS>`

row_entries consists of a `<ROW_HEIGHT>` element for each row in the table, which specify the height in standard coordinates for each row.

The number of `<ROW_HEIGHT>` elements determines the number of rows in the table.

End Tag `</FORM_TABLE>`

Example

```
<FORM_TABLE>
  <ID> 1020 </ID>
  <BOUNDS>
    <LEFT> 99 </LEFT>
    <TOP> 114 </TOP>
    <WIDTH> 40 </WIDTH>
    <HEIGHT> 40 </HEIGHT>
  </BOUNDS>
  <EDITABLE> TRUE </EDITABLE>
  <TABLE_COLUMNS>
    <COLUMN_WIDTH> 20 </COLUMN_WIDTH>
    <COLUMN_WIDTH> 20 </COLUMN_WIDTH>
  </TABLE_COLUMNS>
  <TABLE_ROWS>
    <ROW_HEIGHT> 11 </ROW_HEIGHT>
    <ROW_HEIGHT> 11 </ROW_HEIGHT>
    <ROW_HEIGHT> 11 </ROW_HEIGHT>
  </TABLE_ROWS>
</FORM_TABLE>
```

Form Title Object

Purpose The form resource's first item is usually a form title object in the `FORM_OBJECTS` element.

Start Tag `<FORM_TITLE>`

Child Elements `<TEXT> quoted_text </TEXT>`
Specifies the text for the form's title.

End Tag `</FORM_TITLE>`

Example

```
<FORM_TITLE>
  <TEXT> "Address Entry Details" </TEXT>
</FORM_TITLE>
```

Graphic Family Resource

Purpose	The Graphic Family resource concatenates image files in PNG format into one resource.
Target Format	Format generated by PalmRC: <ul style="list-style-type: none">• Palm OS 4: Not applicable.• Palm OS 5: Not applicable.• Palm OS 6: 'Tbmp'
Start Tag	<code><GRAPHIC_FAMILY_RESOURCE RESOURCE_ID="resource_id" LOCALE="resource_id" OVERLAY_STATUS="resource_id"></code>
Child Elements	<div><code><WIDTH> integer </WIDTH></code> <code><WIDTH> integer </WIDTH></code> <i>integer</i> - Specifies the width of the image in standard coordinates.</div> <div><code><HEIGHT> integer </HEIGHT></code> <i>integer</i> - Specifies the height of the image in standard coordinates.</div> <div><code><IMAGES> one or more IMAGE elements </IMAGES></code> The IMAGES element consists of a collection of IMAGE elements.</div> <div><code><IMAGE> image_defn </IMAGE></code> <i>image_defn</i> consists of the following elements:<div><code><DENSITY> enum </DENSITY></code> This optional element allows you to specify multiple density images in a graphic family resource. 72 - Normal (1X) density (also referred to single density). This is the default, if this element is omitted. 108 - 1.5X density (also referred to as one-and-one half density). 144 - Double (2X) density. 216 - Triple (3X) density.</div><code><GRAYSCALE> boolean </GRAYSCALE></code> Indicates whether this is a grayscale image.</div>

XRD Resource Element Reference

Graphic Family Resource

<DITHER> *enum* </DITHER>

OFF - Dithering is not used for this image.

ONLOAD - The image is dithered when loaded.

ONDRAW - The image is dithered when drawn.

<COLORSPACE> *enum* </COLORSPACE>

Specifies the colorspace used for this image. It is used only in conjunction with a GRAPHIC_FAMILY_DATA element.

<GRAPHIC_FAMILY_DATA> *binary_data*

</GRAPHIC_FAMILY_DATA>

The image data may be specified using either a GRAPHIC_FAMILY_DATA element or a GRAPHIC_FAMILY_FILE element.

A GRAPHIC_FAMILY_DATA element specifies the raster image data inline in the XRD file. The image data is specified as binary data. The image data is the uncompressed raster data specified as rowBytes times numScanLines (rowBytes * numScanLines) bytes of data, where the scan lines are stored top to bottom.

<GRAPHIC_FAMILY_FILE> *ext_file*

</GRAPHIC_FAMILY_FILE>

The image data may be specified using either a GRAPHIC_FAMILY_DATA element or a GRAPHIC_FAMILY_FILE element.

A GRAPHIC_FAMILY_FILE element specifies the image data as the quoted string path of an external image file, which must be in a supported image format such as PNG. The dimensions of the image in the image file must match exactly the dimensions specified by the IMAGE element.

When using external image files, you can store the image as a PNG file, which is the only supported format at this time. For simplicity, you can use GRAPHIC_FAMILY_FILE paths that specify a path relative to the XRD file.

End Tag </GRAPHIC_FAMILY_RESOURCE>

Example

```
<GRAPHIC_FAMILY_RESOURCE RESOURCE_ID="1000" COMMENT="January
or All">

  <WIDTH> 225 </WIDTH>
  <HEIGHT> 34 </HEIGHT>

  <IMAGES>

    <IMAGE>
      <DENSITY>108</DENSITY>
      <GRAPHIC_FAMILY_FILE>"QVGA_Cal_January.png"
      </GRAPHIC_FAMILY_FILE>
    </IMAGE>

    <IMAGE>
      <DENSITY>144</DENSITY>
      <GRAPHIC_FAMILY_FILE>"HVGA_Cal_January.png"
      </GRAPHIC_FAMILY_FILE>
    </IMAGE>

    <IMAGE>
      <DENSITY>216</DENSITY>
      <GRAPHIC_FAMILY_FILE>"VGA_Cal_January.png"
      </GRAPHIC_FAMILY_FILE>
    </IMAGE>

  </IMAGES>

</GRAPHIC_FAMILY_RESOURCE>
```

Menu Bar Resource

Purpose The menu bar stores a list of menus where each menu has a title and a list of menu items.

Note that the calculated bounds of the menu and menu title are explicitly specified in the `BOUNDS` and `TITLE_BOUNDS` elements. These elements are included to accurately reproduce the data stored in Palm OS menu structures.

Because PalmRC calculates the correct values for these elements when an XRD file is compiled, you should not specify these values, especially when working with localized text string. When creating a new resource, you may set these bounds to zero. Future revisions of

XRD Resource Element Reference

Menu Bar Resource

the resource tools will treat these elements as optional, and they should not normally be used.

Target Format Format generated by PalmRC:

- Palm OS 4: 'MBAR'
- Palm OS 5: 'amnu'
- Palm OS 6: 'MBAR'

Start Tag <MENU_BAR_RESOURCE RESOURCE_ID="*resource_id*">

Child Elements <VISIBLE> *boolean* </VISIBLE>
 TRUE - Indicates that the menu bar is visible.
 FALSE - Indicates that the menu bar is not visible.

 <MENUS> *menu_elements* </MENUS>
 menu_elements
 One or more MENU element descriptions.

 <MENU> *menu_description* </MENU>
 menu_description consists of the following sub-elements:
 <TITLE> *quoted_text* </TITLE>
 Defines the menu text for the menu command.

 <HIDDEN> *boolean* </HIDDEN>
 TRUE - the menu is hidden.
 FALSE - the menu is displayed.

`<BOUNDS> bounds_defn </BOUNDS>`

Because PalmRC calculates the correct values for these elements when an XRD file is compiled, you should not specify these values.

`<LEFT> integer </LEFT>` - Specifies, in standard coordinates, the left-most edge of the menu bounds (the “x” value).

`<TOP> integer </TOP>` - Specifies, in standard coordinates, the top-most edge of the menu bounds (the “y” value).

`<WIDTH> integer </WIDTH>` - Specifies, in standard coordinates, the width (the horizontal extent) for the menu bounds.

`<HEIGHT> integer </HEIGHT>` - Specifies, in standard coordinates, the height (the vertical extent) for the menu bounds.

`<TITLE_BOUNDS> bounds_description`

`</TITLE_BOUNDS>`

Because PalmRC calculates the correct values for these elements when an XRD file is compiled, you should not specify these values.

`<LEFT> integer </LEFT>` - Sets the horizontal origin for the menu title bounds (the “x” value).

`<TOP> integer </TOP>` - Sets the vertical origin for the menu title bounds (the “y” value).

`<WIDTH> integer </WIDTH>` - Sets the horizontal extent for the menu title bounds.

`<HEIGHT> integer </HEIGHT>` - Sets the vertical extent for the menu title bounds.

`<MENU_ITEMS> menu_item_elements </MENU_ITEMS>`

`menu_item_elements` - One or more MENU_ITEM element descriptions.

XRD Resource Element Reference

Menu Bar Resource

`<MENU_ITEM> menu_item_description`
`</MENU_ITEM>`
menu_item_description - Consists of the following sub-elements:

`<ID> integer </ID>` - The identifier for the menu item.

`<TITLE> quoted_text </TITLE>` - The menu item text.

`<COMMAND> quoted_text </COMMAND>` - The command shortcut for the menu item.

The menu command shortcut must be a single uppercase letter ('A' through 'Z').

`<HIDDEN> boolean </HIDDEN>`
TRUE - The menu item is hidden.
FALSE - The menu item is displayed.

End Tag `</MENU_BAR_RESOURCE>`

Example The following example shows how to specify the standard Edit menu in your Palm OS application. Note the menu IDs for the specific menu items.

Undo : 10000

Cut : 10001

Copy: 10002

Paste: 10003

Select All: 10004

Separator Line: 10005

Keyboard: 10006

Graffiti Help: 10007

Listing 3.1 Palm OS standard Edit menu items

```
<MENU_BAR_RESOURCE RESOURCE_ID="1000">
  <VISIBLE> TRUE </VISIBLE>
  <MENUS>
    <MENU>
      <TITLE> "Edit" </TITLE>
```

```
<HIDDEN> FALSE </HIDDEN>
<BOUNDS>
  <LEFT> 6 </LEFT>
  <TOP> 14 </TOP>
  <WIDTH> 100 </WIDTH>
  <HEIGHT> 82 </HEIGHT>
</BOUNDS>
<TITLE_BOUNDS>
  <LEFT> 4 </LEFT>
  <TOP> 0 </TOP>
  <WIDTH> 27 </WIDTH>
  <HEIGHT> 12 </HEIGHT>
</TITLE_BOUNDS>
<MENU_ITEMS>
  <MENU_ITEM>
    <ID> 10000 </ID>
    <TITLE> "Undo" </TITLE>
    <COMMAND> "U" </COMMAND>
    <HIDDEN> FALSE </HIDDEN>
  </MENU_ITEM>
  <MENU_ITEM>
    <ID> 10001 </ID>
    <TITLE> "Cut" </TITLE>
    <COMMAND> "X" </COMMAND>
    <HIDDEN> FALSE </HIDDEN>
  </MENU_ITEM>
  <MENU_ITEM>
    <ID> 10002 </ID>
    <TITLE> "Copy" </TITLE>
    <COMMAND> "C" </COMMAND>
    <HIDDEN> FALSE </HIDDEN>
  </MENU_ITEM>
  <MENU_ITEM>
    <ID> 10003 </ID>
    <TITLE> "Paste" </TITLE>
    <COMMAND> "P" </COMMAND>
    <HIDDEN> FALSE </HIDDEN>
  </MENU_ITEM>
  <MENU_ITEM>
    <ID> 10004 </ID>
    <TITLE> "Select All" </TITLE>
    <COMMAND> "S" </COMMAND>
    <HIDDEN> FALSE </HIDDEN>
  </MENU_ITEM>
  <MENU_ITEM>
    <ID> 10005 </ID>
    <TITLE> "-" </TITLE>
    <COMMAND> " " </COMMAND>
```

XRD Resource Element Reference

MIDI Resource

```
        <HIDDEN> FALSE </HIDDEN>
    </MENU_ITEM>
    <MENU_ITEM>
        <ID> 10006 </ID>
        <TITLE> "Keyboard" </TITLE>
        <COMMAND> "K" </COMMAND>
        <HIDDEN> FALSE </HIDDEN>
    </MENU_ITEM>
    <MENU_ITEM>
        <ID> 10007 </ID>
        <TITLE> "Graffiti Help" </TITLE>
        <COMMAND> "G" </COMMAND>
        <HIDDEN> FALSE </HIDDEN>
    </MENU_ITEM>
</MENU_ITEMS>
</MENU>
</MENUS>
</MENU_BAR_RESOURCE>
```

MIDI Resource

Purpose	<p>The MIDI resource is used to define MIDI (Musical Instrument Digital Interface) data for your application.</p> <p>A MIDI resource may be defined as binary data or external file data.</p>
Target Format	<p>Format generated by PalmRC:</p> <ul style="list-style-type: none">• All OS targets: 'MIDI '
Start Tag	<code><MIDI_RESOURCE RESOURCE_ID="resource_id"></code>
Child Elements	<p><code><MIDI_RAW_DATA> <i>binary_data</i> </MIDI_RAW_DATA></code> Use this format to specify inline binary data.</p> <p><code><MIDI_FILE> <i>ext_file</i> </MIDI_FILE></code> Use this format to specify external data contained in a file.</p>
End Tag	<code></MIDI_RESOURCE></code>
Example	<i>Inline Data</i>

```
<MIDI_RESOURCE RESOURCE_ID="999">
  <MIDI_RAW_DATA>
    4D 54 68 64 00 00 00 06 00 00 00 01 01 90 4D 54
    72 6B 00 00 00 32 00 90 62 7F 4A 62 00 00 5F 7F
    82 28 5F 00 81 14 62 7F 4A 62 00 00 5F 7F 82 28
```

```
5F 00 81 14 62 7F 4A 62 00 00 5F 7F 82 28 5F 00
85 1A 5F 01 00 FF 2F 00
</MIDI_RAW_DATA>
</MIDI_RESOURCE>
```

External File Data

```
<MIDI_RESOURCE RESOURCE_ID="1999">
  <MIDI_FILE> "../TestMidi/twilight.mid" </MIDI_FILE>
</MIDI_RESOURCE>
```

Overlay Resource

Purpose The overlay resource is used to provide localized strings which are used by the Palm OS Overlay Manager.

NOTE: This resource type is used for viewing decompiled resources only. It is not used for creating overlay resources. To create overlay resources, use the tool hOverlay.

Target Format Format generated by PalmRC:

- All OS targets: 'ovly'

Start Tag <OVERLAY_RESOURCE RESOURCE_ID="resource_id">

End Tag </OVERLAY_RESOURCE>

XRD Resource Element Reference

Raw Resource

Raw Resource

Purpose The raw resource is used to represent any resource for which there is no logical resource description available, but the binary data is explicitly specified.

It is also used when extracting Palm OS code resources such as 'code' and 'data' to be stored in an XRD file.

The resource data may be specified inline as binary data using the RES_DATA element, or it may be specified in an external file using the DATA_FILE element.

Target Format The source type for raw resources is specified with the RES_TYPE element, as described below.

Start Tag <RAW_RESOURCE RESOURCE_ID="*resource_id*">

Child Elements <RES_TYPE> *four_character_code* </RES_TYPE>
Use this value to specify the resource type for the raw resource.

<RES_DATA> *binary_data* </RES_DATA>
Use this format to specify inline binary data.

<DATA_FILE> *ext_file* </DATA_FILE>
Use this format to specify external data contained in a file.

End Tag </RAW_RESOURCE>

Example *Inline Data*

```
<RAW_RESOURCE RESOURCE_ID="0">
  <RES_TYPE> 'code' </RES_TYPE>
  <RES_DATA>
    00 00 00 30 00 00 00 08 00 00 00 08 00 00 00 20
    00 00 3F 3C 00 01 A9 F0
  </RES_DATA>
</RAW_RESOURCE>
```

External Data

```
<RAW_RESOURCE RESOURCE_ID="65535">
  <RES_TYPE> 'XXXX' </RES_TYPE>
  <DATA_FILE> "../TestData/xxxx.bin" </DATA_FILE>
</RAW_RESOURCE>
```

Schema Database Resource

Purpose	The schema database resource represents a “flattened” Palm OS schema database. For more information about defining and using schema databases, see <i>Exploring Palm OS: Memory, Databases, and Files</i> .
Target Format	Format generated by PalmRC: <ul style="list-style-type: none">• Palm OS 4: Not applicable.• Palm OS 5: Not applicable.• Palm OS 6: 'scdb'
Start Tag	<SCHEMA_DATABASE_RESOURCE RESOURCE_ID="resource_id">
Child Elements	<div><SDB_HEADER> <i>schema_header</i> </SDB_HEADER></div> <div>The <i>schema_header</i> contains the following elements:</div> <div><SDB_DB_NAME> <i>quoted_text</i> </SDB_DB_NAME> <i>quoted_text</i> - The name for this database.</div> <div>Databases in Palm OS Cobalt are uniquely identified by a combination of the database’s name and its creator ID. Thus, database names need only be unique for a single creator ID: two databases with the same name can reside on a single handheld as long as their creator IDs differ.</div> <div><SDB_DB_USER_NAME> <i>quoted_text</i> </SDB_DB_USER_NAME> <i>quoted_text</i> - The database’s display name.</div> <div><SDB_DB_TYPE> <i>four_character_code</i> </SDB_DB_TYPE> <i>four_character_code</i> - The database’s type, specified when the database is created.</div> <div><SDB_DB_CREATOR> <i>quoted_text</i> </SDB_DB_CREATOR> <i>quoted_text</i> - The database’s creator, specified when the database is created.</div> <div><SDB_DB_VERSION> <i>integer</i> </SDB_DB_VERSION> <i>integer</i> - The application-specific version number. The default version number is 0.</div>

XRD Resource Element Reference

Schema Database Resource

<SDB_SCHEMAS> *schema_def* </SDB_SCHEMAS>

The *schema_def* contains the following elements:

<SDB_SCHEMA_ID> *resource_id* </SDB_SCHEMA_ID>
resource_id - A 32-bit user-defined identifier. This ID must be unique for a given table.

<SDB_SCHEMA_NAME> *quoted_text*
</SDB_SCHEMA_NAME>
quoted_text - A user-defined name for the schema. This name can be up to 32 bytes in length, including the terminating null character.

This name must be a valid SQL identifier, as defined by Palm OS. Specifically, identifiers must start with either an uppercase letter ('A' through 'Z'), a lowercase letter ('a' through 'z'), or the underscore character ('_'). The remaining characters can be letters, numbers ('0' through '9'), or the underscore character.

<SDB_COLUMNS> one or more SDB_COLUMN elements
</SDB_COLUMNS>

<SDB_COLUMN> *column_def* <SDB_COLUMN>
The *column_def* contains the following elements:

<SDB_COLUMN_ID> *resource_id*
</SDB_COLUMN_ID>
resource_id - A 32-bit user-defined identifier. This ID must be unique for a given table.

<SDB_COLUMN_NAME> *quoted_text*
</SDB_COLUMN_NAME>
quoted_text - A user-defined name for the column. This name can be up to 32 bytes in length, including the terminating null character. The column name is stored in a single user-defined language encoding.

This name must be a valid SQL identifier, as defined by Palm OS. Specifically, identifiers must start with either an uppercase letter ('A' through 'Z'), a lowercase letter ('a' through 'z'), or the underscore character ('_'). The remaining characters can be letters, numbers ('0' through '9'), or the underscore character.

<SDB_COLUMN_ATTRIBUTES> *column_attr_defn*

</SDB_COLUMN_ATTRIBUTES>

where *column_attr_defn* consists of the following elements:

<SDB_COLUMN_ATTRIBUTE_NONSYNCABLE>

boolean

</SDB_COLUMN_ATTRIBUTE_NONSYNCABLE>

boolean - Indicates whether the column data is to be synchronized.

<SDB_COLUMN_ATTRIBUTE_WRITEABLE> *boolean*

</SDB_COLUMN_ATTRIBUTE_WRITEABLE>

boolean - Indicates whether the column data can be modified.

<SDB_COLUMN_TYPE> *column_type*

</SDB_COLUMN_TYPE>

column_type - The type of data contained within the database column, which is one of the following types:

BOOLEAN, BLOB, CHAR, VARCHAR, INT8, INT16, INT32, UINT8, UINT16, UINT32.

<SDB_COLUMN_SIZE> *integer*

</SDB_COLUMN_SIZE>

integer - The maximum size, in bytes, for columns that contain variable-length strings, blobs, and vectors.

<SDB_COLUMN_PROPERTIES>

<SDB_COLUMN_PROPERTY> *property_def*

</SDB_COLUMN_PROPERTY>

</SDB_COLUMN_PROPERTIES>

property_def allows you to define custom properties for a column, and contains the following elements:

<SDB_COLUMN_PROPERTY_ID> *integer*

</SDB_COLUMN_PROPERTY_ID>

integer - The identifier for the custom property. This identifier must be greater than 10, the value assigned

XRD Resource Element Reference

Schema Database Resource

for the schema database constant
dbColumnPropertyUpperBound.

```
<SDB_COLUMN_PROPERTY_DATA> column_data  
</SDB_COLUMN_PROPERTY_DATA>
```

column_data can be defined as either *binary_data* or as *quoted_text*. If you specify the column data as *quoted_text*, then the specified string is transformed into binary data and a terminating null character is appended.

```
<SDB_CATEGORIES> one or more SDB_CATEGORY elements  
</SDB_CATEGORIES>
```

```
<SDB_CATEGORY> category_def  
</SDB_CATEGORY>
```

where *category_def* contains the following elements:

```
<SDB_CATEGORY_ID> integer  
</SDB_CATEGORY_ID>
```

integer - The category ID value.

```
<SDB_CATEGORY_NAME> quoted_text  
</SDB_CATEGORY_NAME>
```

quoted_text - The category name.

```
<SDB_CATEGORY_EDITABLE> boolean  
</SDB_CATEGORY_EDITABLE>
```

boolean - Indicates whether the category information can be changed.

Note: If you specify categories using the SDB_CATEGORY elements, the first category must be defined with SDB_CATEGORY_ID set to 0, and SDB_CATEGORY_NAME set to "Unfiled", as shown in the example below.

```
<SDB_CATEGORY>  
  <SDB_CATEGORY_ID> 0 </SDB_CATEGORY_ID>  
  <SDB_CATEGORY_NAME> "Unfiled" </SDB_CATEGORY_NAME>  
  <SDB_CATEGORY_EDITABLE> TRUE </SDB_CATEGORY_EDITABLE>  
</SDB_CATEGORY>
```

<SDB_RECORDS> one or more SDB_RECORD elements

</SDB_RECORDS>

<SDB_RECORD> *record_def* </SDB_RECORD>

where *record_def* contains the following elements:

<SDB_RECORD_ID> *integer* </SDB_RECORD_ID>

integer - A 32-bit user-defined identifier. This record ID must be an integer in the range 2048 through 8191, in increasing order (smallest first, largest last).

<SDB_SCHEMA_ID> *integer* </SDB_SCHEMA_ID>

integer - A 32-bit user-defined identifier.

<SDB_RECORD_ATTRIBUTES> *attrib_def*

</SDB_RECORD_ATTRIBUTES>

where *attrib_def* contains the following elements:

<SDB_RECORD_ATTRIBUTE_READONLY> *boolean*

</SDB_RECORD_ATTRIBUTE_READONLY>

boolean - Indicates whether this attribute can be changed.

<SDB_RECORD_ATTRIBUTE_SECRET> *boolean*

</SDB_RECORD_ATTRIBUTE_SECRET>

boolean - Indicates whether this attribute is hidden.

<SDB_RECORD_DATA> one or more
SDB_RECORD_COLUMN_DATA elements
</SDB_RECORD_DATA>

<SDB_RECORD_COLUMN_DATA> *column_data_def*

</SDB_RECORD_COLUMN_DATA>

where *column_data_def* contains the following elements:

<SDB_COLUMN_ID> *integer* </SDB_COLUMN_ID>

integer - The column ID.

<SDB_RECORD_DATA_VALUE> *quoted_text*

</SDB_RECORD_DATA_VALUE>

quoted_text - The column data.

<SDB_RECORD_CATEGORIES> one or more
SDB_CATEGORY_ID elements

</SDB_RECORD_CATEGORIES>

XRD Resource Element Reference

Schema Database Resource

<SDB_CATEGORY_ID> *integer* </SDB_CATEGORY_ID>
integer - The category ID value.

End Tag </SCHEMA_DATABASE_RESOURCE>

Example <SCHEMA_DATABASE_RESOURCE RESOURCE_ID="1000">
 <SDB_HEADER>
 <SDB_DB_NAME> "Resources" </SDB_DB_NAME>
 <SDB_DB_USER_NAME> "Manager" </SDB_DB_USER_NAME>
 <SDB_DB_TYPE> 'appl' </SDB_DB_TYPE>
 <SDB_DB_CREATOR> 'mgra' </SDB_DB_CREATOR>
 <SDB_DB_VERSION> 3 </SDB_DB_VERSION>
 </SDB_HEADER>
 <SDB_SCHEMAS>
 <SDB_SCHEMA>
 <SDB_SCHEMA_ID> 1 </SDB_SCHEMA_ID>
 <SDB_SCHEMA_NAME> "X" </SDB_SCHEMA_NAME>
 <SDB_COLUMNS>
 <SDB_COLUMN>
 <SDB_COLUMN_ID> 2102 </SDB_COLUMN_ID>
 <SDB_COLUMN_NAME> "Contact" </SDB_COLUMN_NAME>
 <SDB_COLUMN_TYPE> VAR_CHAR </SDB_COLUMN_TYPE>
 <SDB_COLUMN_SIZE> 32 </SDB_COLUMN_SIZE>
 </SDB_COLUMN>
 </SDB_COLUMNS>
 </SDB_SCHEMA>
 </SDB_SCHEMAS>
 <SDB_CATEGORIES>
 <SDB_CATEGORY>
 <SDB_CATEGORY_ID> 0 </SDB_CATEGORY_ID>
 <SDB_CATEGORY_NAME> "Unfiled" </SDB_CATEGORY_NAME>
 <SDB_CATEGORY_EDITABLE> TRUE </SDB_CATEGORY_EDITABLE>
 </SDB_CATEGORY>
 </SDB_CATEGORIES>
</SCHEMA_DATABASE_RESOURCE>

Soft Constant Resource

Purpose	The soft constant stores a single 32-bit unsigned integer number.
Target Format	Format generated by PalmRC: <ul style="list-style-type: none">• Palm OS 4: 'tint'• Palm OS 5: 'aint'• Palm OS 6: 'tint'
Start Tag	<SOFT_CONSTANT_RESOURCE RESOURCE_ID=" <i>resource_id</i> ">
Child Elements	<VALUE> <i>integer</i> </VALUE> The initial setting for the soft constant.
End Tag	</SOFT_CONSTANT_RESOURCE>
Example	<pre><SOFT_CONSTANT_RESOURCE RESOURCE_ID="1000"> <VALUE> 1000 </VALUE> </SOFT_CONSTANT_RESOURCE> <SOFT_CONSTANT_RESOURCE RESOURCE_ID="1001"> <VALUE> 0x3E8 </VALUE> </SOFT_CONSTANT_RESOURCE></pre>

String List Resource

Purpose	The string list stores a prefix string, and zero or more additional strings.
Target Format	Format generated by PalmRC: <ul style="list-style-type: none">• All OS targets: 'tSTL'
Start Tag	<STRING_LIST_RESOURCE RESOURCE_ID=" <i>resource_id</i> ">
Child Elements	<PREFIX> <i>quoted_text</i> </PREFIX> The PREFIX element defines the beginning string for the string list. <STRINGS> <i>text_elements</i> </STRINGS> The STRINGS elements define the alternative ending strings for the string list.
End Tag	</STRING_LIST_RESOURCE>

XRD Resource Element Reference

String Resource

Example	<pre><STRING_LIST_RESOURCE RESOURCE_ID="1000"> <PREFIX> "The show " </PREFIX> <STRINGS> <TEXT> "never ends." </TEXT> <TEXT> "is over." </TEXT> </STRINGS > </STRING_LIST_RESOURCE></pre>
----------------	--

String Resource

Purpose	The string resource stores a single string.
Target Format	Format generated by PalmRC: <ul style="list-style-type: none">• All OS targets: ' tSTR '
Start Tag	<code><STRING_RESOURCE RESOURCE_ID="resource_id"></code>
Child Elements	<code><TEXT> <i>quoted_text</i> </TEXT></code>
End Tag	<code></STRING_RESOURCE></code>
Example	<pre><STRING_RESOURCE RESOURCE_ID="1000"> <TEXT> "Your string here." </TEXT> </STRING_RESOURCE></pre>

TrueType Font Resource

Purpose	The TrueType font resource defines a TrueType font. The TrueType font resource may be defined as binary data or external file data.
Target Format	Format generated by PalmRC: <ul style="list-style-type: none">• Palm OS 4: Not applicable.• Palm OS 5: Not applicable.• Palm OS 6: ' fttf '
Start Tag	<code><TRUETYPE_FONT_RESOURCE RESOURCE_ID="resource_id"></code>
Child Elements	<code><TTFONT_RAW_DATA> <i>binary_data</i> </TTFONT_RAW_DATA></code> <p>Use this format to specify inline binary data. The binary data must be the same format stored in a TrueType font file (TTF file), as used in Microsoft Windows.</p>

End Tag `<TTFONT_FILE> ext_file </TTFONT_FILE>`
ext_file - Use this format to specify the name of a TrueType font file (TTF file), as used in Microsoft Windows.

Example *Inline Data*

```
<TRUETYPE_FONT_RESOURCE RESOURCE_ID="4142">
  <TTFONT_RAW_DATA>
    00 01 00 00 00 11 01 00 00 04 00 10 4F 53 2F 32
    ...
    B8 00 0A 2B B8 00 00 2B
  </TTFONT_RAW_DATA>
</TRUETYPE_FONT_RESOURCE>
```

External File Data

```
<TRUETYPE_FONT_RESOURCE RESOURCE_ID="1000">
  <TTFONT_FILE> "../TestFonts/hotpizza.ttf" </TTFONT_FILE>
</TRUETYPE_FONT_RESOURCE>
```

Wave Sound Resource

Purpose The wave sound resource is used to define wave sound data for your application. The wave sound resource may be defined as binary data or external file data.

Target Format Format generated by PalmRC:

- Palm OS 4: Not applicable.
- Palm OS 5: 'wave'
- Palm OS 6: 'wave'

Start Tag `<WAVE_SOUND_RESOURCE RESOURCE_ID="resource_id">`

Child Elements `<WAVE_RAW_DATA> binary_data </WAVE_RAW_DATA>`

Use this format to specify inline binary data.

`<WAVE_FILE> ext_file </WAVE_FILE>`

Use this format to specify external data contained in a file.

End Tag `</WAVE_SOUND_RESOURCE>`

XRD Resource Element Reference

Window Constraints Resource

Example *Inline Data*

```
<WAVE_SOUND_RESOURCE RESOURCE_ID="3000">
  <WAVE_RAW_DATA>
    52 49 46 46 22 27 00 00 57 41 56 45 66 6D 74 20
    ...
    80 80 80 80 80 7B 79 81 7B 00
  </WAVE_RAW_DATA>
</WAVE_SOUND_RESOURCE>
```

External File Data

```
<WAVE_SOUND_RESOURCE RESOURCE_ID="2000">
  <WAVE_FILE> " ./TestWav/meow.wav" </WAVE_FILE>
</WAVE_SOUND_RESOURCE>
```

Window Constraints Resource

Purpose The window constraints resource allows you to specify the creation size and resize constraints for your application window.

You can specify window constraints for update-based windows and transitional windows. For more information about window types and window constraints, see *Exploring Palm OS: User Interface*.

Target Format Format generated by PalmRC:

- Palm OS 4: Not applicable.
- Palm OS 5: Not applicable.
- Palm OS 6: ' frmb '

Start Tag <WINDOW_CONSTRAINTS_RESOURCE
RESOURCE_ID="resource_id">

Child Elements <VERSION> *integer* </VERSION>
integer - Indicates the window constraints version. For Palm OS Cobalt, use the integer value 1.

<WINDOW_CREATE_FLAGS> *integer*
</WINDOW_CREATE_FLAGS>
integer - Uses the values defined by the WinFlagsType enumeration. For example, to specify normal application window flags, set this value to 0x00000000. For more

information, see the WinFlagsType description in the book *Exploring Palm OS: User Interface*.

<WINDOW_X_POS> *integer* </WINDOW_X_POS>

The position, given in standard coordinates, of the left side of the window.

<WINDOW_Y_POS> *integer* </WINDOW_Y_POS>

The position of the top of the window in standard coordinates.

<WINDOW_WIDTH_MIN> *integer* </WINDOW_WIDTH_MIN>

The minimum possible width of the window in standard coordinates.

<WINDOW_WIDTH_MAX> *integer* </WINDOW_WIDTH_MAX>

The maximum width of the window in standard coordinates.

<WINDOW_WIDTH_PREF> *integer* </WINDOW_WIDTH_PREF>

The preferred width of the window in standard coordinates.

<WINDOW_HEIGHT_MIN> *integer* </WINDOW_HEIGHT_MIN>

The minimum height that the window should ever be made in standard coordinates.

<WINDOW_HEIGHT_MAX> *integer* </WINDOW_HEIGHT_MAX>

The maximum height for the window in standard coordinates.

<WINDOW_HEIGHT_PREF> *integer* </WINDOW_HEIGHT_PREF>

The preferred height for this window in standard coordinates.

End Tag </WINDOW_CONSTRAINTS_RESOURCE>

Comments The special values SHRT_MIN and SHRT_MAX, which are defined in the Window.h header file, are not recognized by PalmRC. If you want to use these values in your window constraints resource, you need to specify the raw values rather than the constant names.

SHRT_MIN -32768

The system decides the value of this constraint.

SHRT_MAX 32767

Specifies a value that means "as large as possible."

Example <WINDOW_CONSTRAINTS_RESOURCE RESOURCE_ID="2222">

<VERSION> 1 </VERSION>

<WINDOW_CREATE_FLAGS> 0x00000007 </WINDOW_CREATE_FLAGS>

XRD Resource Element Reference

Word Integer List Resource

```
<WINDOW_X_POS> 10 </WINDOW_X_POS>
<WINDOW_Y_POS> 20 </WINDOW_Y_POS>
<WINDOW_WIDTH_MIN> 100 </WINDOW_WIDTH_MIN>
<WINDOW_WIDTH_MAX> 200 </WINDOW_WIDTH_MAX>
<WINDOW_WIDTH_PREF> 140 </WINDOW_WIDTH_PREF>
<WINDOW_HEIGHT_MIN> 110 </WINDOW_HEIGHT_MIN>
<WINDOW_HEIGHT_MAX> 180 </WINDOW_HEIGHT_MAX>
<WINDOW_HEIGHT_PREF> 120 </WINDOW_HEIGHT_PREF>
</WINDOW_CONSTRAINTS_RESOURCE>
```

Word Integer List Resource

Purpose The word integer list stores 0 or more 16-bit unsigned integer numbers.

Note that unlike the byte integer list and dword integer list, the word integer list does not have a default item property.

Target Format Format generated by PalmRC:

- Palm OS 4: 'wrdl '
- Palm OS 5: 'awrd '
- Palm OS 6: 'awrd '

Start Tag <WORD_LIST_RESOURCE RESOURCE_ID="*resource_id*">

Child Elements <VALUES> *value_elements* </VALUES>
value_elements consists of one or more VALUE element descriptions.
<VALUE> *integer* </VALUE>

End Tag </WORD_LIST_RESOURCE>

Example <WORD_LIST_RESOURCE RESOURCE_ID="1000">
 <VALUES>
 <VALUE> 0 </VALUE>
 <VALUE> 0xFFFF </VALUE>
 </VALUES>
</WORD_LIST_RESOURCE>

Custom Resources

This chapter provides the reference information for defining your own XML-based format for resources that are not already defined. This custom resource defines both the XML representation of the logical resource data as well as the physical resource format of the resulting binary data.

You can use the `PalmRC` resource compiler to compile your custom resource data descriptions just as easily as using the pre-defined resource formats. In addition, the custom resource description can be used by other tools such as the Generate XRD resource decompiler.

This chapter consists of the following sections:

- “[Reference Syntax](#)” on page 120 describes the XML format you use to define and refer to custom resources.
- “[Custom Enumeration Description](#)” on page 124 describes the reference format for custom enum types.
- “[Custom Structure Description](#)” on page 126 describes the reference format for custom struct types.
- “[Custom Resource Description](#)” on page 127 describes the reference format for custom resource types.
- “[Name Format Requirements](#)” on page 129 describes format requirements for names used in custom resources.
- “[Data Model Objects](#)” on page 130 describes the additional data model objects you can use in your custom resource descriptions.

Reference Syntax

This section describes the XML syntax that you use to define custom types and that you use to refer to a custom type description.

Palm OS Custom Resource Type Description

Purpose Use the Palm OS custom resource type element to define a custom resource type. There are three types that you can define:

- A custom enumeration description (see “[Custom Enumeration Description](#)” on page 124 for more information)
- A custom structure description (see “[Custom Structure Description](#)” on page 126 for more information)
- A custom resource description (see “[Custom Resource Description](#)” on page 127 for more information)

Start Tag <PALMOS_CUSTOM_RESOURCE_TYPES>

Child Elements <DEFINE_ENUM_TYPE NAME=*quoted_text*> *enum_defn*
</DEFINE_ENUM_TYPE>

Defines a custom enumeration description, as described in “[Custom Enumeration Description](#)” on page 124.

<DEFINE_STRUCT_TYPE NAME=*quoted_text*> *struct_defn*
</DEFINE_STRUCT_TYPE>

Defines a custom structure description, as described in “[Custom Structure Description](#)” on page 126.

<DEFINE_RESOURCE_TYPE NAME=*quoted_text*>
resource_defn </DEFINE_RESOURCE_TYPE>

Defines a custom resource description, as described in “[Custom Resource Description](#)” on page 127.

End Tag </PALMOS_CUSTOM_RESOURCE_TYPES>

Example <PALMOS_CUSTOM_RESOURCE_TYPES>

```

<DEFINE_STRUCT_TYPE NAME="POINT_TYPE">
  <STRUCT_ITEMS>
    <FIELD_SINT16>
      <FIELD_NAME> "X" </FIELD_NAME>
      <DISPLAY_FORMAT> DECIMAL </DISPLAY_FORMAT>
    </FIELD_SINT16>
    <FIELD_SINT16>

```

```
        <FIELD_NAME> "Y" </FIELD_NAME>
        <DISPLAY_FORMAT> DECIMAL </DISPLAY_FORMAT>
    </FIELD_SINT16>
</STRUCT_ITEMS>
</DEFINE_STRUCT_TYPE>

</PALMOS_CUSTOM_RESOURCE_TYPES>

<PALMOS_CUSTOM_RESOURCE_TYPES>

    <DEFINE_RESOURCE_TYPE NAME="POINT_LIST">
        <RES_TYPE> 'PLst' </RES_TYPE>
        <ENDIANNESS> BIG_ENDIAN </ENDIANNESS>
        <STRUCT_ITEMS>
            <ARRAY_COUNT_UINT16>
                <ARRAY_FIELD> POINTS </ARRAY_FIELD>
            </ARRAY_COUNT_UINT16>
            <FIELD_ARRAY>
                <FIELD_NAME> "POINTS" </FIELD_NAME>
                <ARRAY_ITEM>
                    <FIELD_STRUCT>
                        <FIELD_NAME> "POINT" </FIELD_NAME>
                        <STRUCT_TYPE> POINT_TYPE </STRUCT_TYPE>
                    </FIELD_STRUCT>
                </ARRAY_ITEM>
            </FIELD_ARRAY>
            <STRING_TABLE>
            </STRING_TABLE>
        </STRUCT_ITEMS>
    </DEFINE_RESOURCE_TYPE>

</PALMOS_CUSTOM_RESOURCE_TYPES>
```

Palm OS Custom Resource Type Use

There are two elements that allow you to refer to a Palm OS custom resource type:

- [“Custom Resource Element”](#)
- [“Custom Types Files Element”](#)

Custom Resources

Custom Resource Element

Custom Resource Element

Purpose Use the custom resource element to refer to a previously defined custom resource by name.

Start Tag `<CUSTOM_RESOURCE RESOURCE_ID="resource_id">`

Child Elements `<CUSTOM_RES_TYPE> resource_name </CUSTOM_RES_TYPE>`
Specifies the name of the custom resource type, which must conform to the rules described in “[Name Format Requirements](#)” on page 129.

`<CUSTOM_DATA> data_desc </CUSTOM_DATA>`
Specifies data in the format you defined by using the custom resource type’s description.

End Tag `</CUSTOM_RESOURCE>`

Example

```
<CUSTOM_RESOURCE RESOURCE_ID="1000">
  <CUSTOM_RES_TYPE> POINT_LIST </CUSTOM_RES_TYPE>
  <CUSTOM_DATA>
    <POINTS>
      <POINT>
        <X> 10 </X>
        <Y> 10 </Y>
      </POINT>
      <POINT>
        <X> 50 </X>
        <Y> 10 </Y>
      </POINT>
      <POINT>
        <X> 50 </X>
        <Y> 50 </Y>
      </POINT>
      <POINT>
        <X> 10 </X>
        <Y> 50 </Y>
      </POINT>
      <POINT>
        <X> 10 </X>
        <Y> 10 </Y>
      </POINT>
    </POINTS>
  </CUSTOM_DATA>
</CUSTOM_RESOURCE>
```

Custom Types Files Element

Purpose Use the custom types files element to refer to custom types that are contained in a separate custom resource type descriptions file.

NOTE: Custom resource type descriptions files must have an XRT file extension.

Start Tag `<CUSTOM_TYPES_FILES>`

Child Elements `<CUSTOM_TYPES_FILE> ext_file </CUSTOM_TYPES_FILE>`
Specifies the name of the XRT file, the file that contains custom resource type descriptions.

End Tag `</CUSTOM_TYPES_FILES>`

Example

```
<CUSTOM_TYPES_FILES>
  <CUSTOM_TYPES_FILE>
    "SimpleExample.xrt"
  </CUSTOM_TYPES_FILE>
</CUSTOM_TYPES_FILES>
```

Comments You can define one or more XRT files containing custom resource type descriptions. These XRT files are XML files that follow the Palm OS Custom Resource Types format, using the PALMOS_CUSTOM_RESOURCE_TYPES document type.

You can reference one or more XRT files from XRD files by using the optional CUSTOM_TYPES_FILES header block in the XRD file.

Each custom types file is specified in the same manner as other XRD external file references:

- If a custom types file is specified as a filename only, then the XRT file is expected to be in the same directory as the referencing XRD.
- If a custom types file is specified with a relative path URL syntax, then the XRT file is located using this path relative to the referencing XRD file

Example of an XRD file reference

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<PALMOS_RESOURCE_FILE>
  <COMMENT_TEXT> "CustomRes.xrd" </COMMENT_TEXT>
  <CUSTOM_TYPES_FILES>
```

Custom Resources

Custom Enumeration Description

```
<CUSTOM_TYPES_FILE>
  "CustomResTypes.xrt"
</CUSTOM_TYPES_FILE>
<CUSTOM_TYPES_FILE>
  "CustomResTypes2.xrt"
</CUSTOM_TYPES_FILE>
</CUSTOM_TYPES_FILES>

<CUSTOM_RESOURCE RESOURCE_ID="1000">
  <CUSTOM_RES_TYPE> MY_RES_TYPE </CUSTOM_RES_TYPE>
  <CUSTOM_DATA>
    <X> 1941 </X>
    <Y> -235 </Y>
  </CUSTOM_DATA>
</CUSTOM_RESOURCE>
</PALMOS_RESOURCE_FILE>
```

Custom resource types must be defined before being used by CUSTOM_RESOURCE descriptions.

Custom Enumeration Description

An enum description defines an integer field with a fixed set of labeled values. Subsequently a field of the enum type can be defined in a custom resource data structure. The field data in the custom resource XML must then be specified as one of the defined enum labels.

Define Enum Type Element

Purpose Use the define enum type element to define a custom enumeration.

Start Tag `<DEFINE_ENUM_TYPE NAME=quoted_text>`
Specifies the name of the custom enum, which must conform to the rules described in "[Name Format Requirements](#)" on page 129.

Child Elements `<ENUM_DATA_TYPE> data_type </ENUM_DATA_TYPE>`
Specifies the data type of the enum: either UINT8, UINT16, or UINT32.

`<ENUM_ITEMS> one or more ENUM_ITEM element`
`</ENUM_ITEMS>`

`<ENUM_ITEM> enum_defn <ENUM_ITEM>`
where *enum_defn* consists of the following elements:

`<ENUM_ITEM_VALUE> integer`
`</ENUM_ITEM_VALUE>`

integer is a unique integer value that fits in the enum data type.

`<ENUM_ITEM_NAME> quoted_text`
`</ENUM_ITEM_NAME>`

quoted_text is a unique label, which must conform to the rules described in "[Name Format Requirements](#)" on page 129.

End Tag `</DEFINE_ENUM_TYPE>`

Example Given an enum defined in C language as follows:

```
enum {
    opCode_MoveTo = 1,
    opCode_LineTo = 2,
    opCode_Text = 3
};

typedef uint16_t DrawingOpcode;
```

A corresponding custom types description might be:

```
<DEFINE_ENUM_TYPE NAME="DRAWING_OPCODE">
  <COMMENT_TEXT>
    "DRAWING_OPCODE is an example of an \n"
    "enumerated value description."
  </COMMENT_TEXT>
  <ENUM_DATA_TYPE> UINT16 </ENUM_DATA_TYPE>
  <ENUM_ITEMS>
    <ENUM_ITEM>
      <ENUM_ITEM_VALUE> 1 </ENUM_ITEM_VALUE>
      <ENUM_ITEM_NAME> "OPCODE_MOVE_TO" </ENUM_ITEM_NAME>
    </ENUM_ITEM>
    <ENUM_ITEM>
      <ENUM_ITEM_VALUE> 2 </ENUM_ITEM_VALUE>
      <ENUM_ITEM_NAME> "OPCODE_LINE_TO" </ENUM_ITEM_NAME>
    </ENUM_ITEM>
    <ENUM_ITEM>
      <ENUM_ITEM_VALUE> 3 </ENUM_ITEM_VALUE>
      <ENUM_ITEM_NAME> "OPCODE_TEXT" </ENUM_ITEM_NAME>
  </ENUM_ITEMS>
</DEFINE_ENUM_TYPE>
```

```
</ENUM_ITEM>
</ENUM_ITEMS>
</DEFINE_ENUM_TYPE>
```

Custom Structure Description

A struct description defines a named list of fields. Subsequently a field of the struct type can be defined in other custom resource data structures.

Define Struct Type Element

Purpose Use the define struct type element to define a custom structure.

Start Tag `<DEFINE_STRUCT_TYPE NAME=quoted_text>`
Specifies the name of the custom structure, which must conform to the rules described in “[Name Format Requirements](#)” on page 129.

Child Elements `<STRUCT_ITEMS> data_defs </STRUCT_ITEMS>`
Defines the data items in the structure, using data model objects described in “[Data Model Objects](#)” on page 130.

End Tag `</DEFINE_STRUCT_TYPE>`

Example Given a structure defined in C language as follows:

```
struct Rect {
    sint16_t top;
    sint16_t left;
    sint16_t bottom;
    sint16_t right;
};
```

A corresponding custom types description might be:

```
<DEFINE_STRUCT_TYPE NAME="RECT">

    <COMMENT_TEXT>
        "RECT is an example of a custom struct description.\n"
        "This example only uses simple field types."
    </COMMENT_TEXT>

    <STRUCT_ITEMS>
```

```
<FIELD_SINT16>
  <FIELD_NAME> "TOP" </FIELD_NAME>
  <DISPLAY_FORMAT> DECIMAL </DISPLAY_FORMAT>
</FIELD_SINT16>

<FIELD_SINT16>
  <FIELD_NAME> "LEFT" </FIELD_NAME>
  <DISPLAY_FORMAT> DECIMAL </DISPLAY_FORMAT>
</FIELD_SINT16>

<FIELD_SINT16>
  <FIELD_NAME> "BOTTOM" </FIELD_NAME>
  <DISPLAY_FORMAT> DECIMAL </DISPLAY_FORMAT>
</FIELD_SINT16>

<FIELD_SINT16>
  <FIELD_NAME> "RIGHT" </FIELD_NAME>
  <DISPLAY_FORMAT> DECIMAL </DISPLAY_FORMAT>
</FIELD_SINT16>

</STRUCT_ITEMS>

</DEFINE_STRUCT_TYPE>
```

Custom Resource Description

A resource type description defines a PRC resource type code, a default endianness, and the corresponding data structure. The data structure is defined the same way as for a struct type, namely as a named list of fields.

Define Resource Type Element

- Purpose** Use the define resource type element to define a custom resource.
- Start Tag** `<DEFINE_RESOURCE_TYPE NAME=quoted_text>`
Specifies name of the custom resource, which must conform to the rules described in “[Name Format Requirements](#)” on page 129.

Custom Resources

Define Resource Type Element

Child Elements	<pre><RES_TYPE> four_character_code </RES_TYPE></pre> <p>Specifies a four-character code representing the resource type for the custom resource.</p> <pre><ENDIANNESS> enum </ENDIANNESS></pre> <p>Specifies the endianness of the custom resource, either <code>BIG_ENDIAN</code> or <code>LITTLE_ENDIAN</code>.</p> <pre><STRUCT_ITEMS> data_defns </STRUCT_ITEMS></pre> <p>Defines the data items in the structure, using data model objects described in “Data Model Objects” on page 130.</p>
End Tag	<pre></DEFINE_RESOURCE_TYPE></pre>
Example	Given a resource type defined in C language as follows:

```
typedef struct
{
    UInt16 count; // number of items
    UInt16 item[]; // variable length
}
WordListRscType;
```

A corresponding custom resource type description might be:

```
<DEFINE_RESOURCE_TYPE NAME="WORD_LIST">

    <RES_TYPE> 'awrd' </RES_TYPE>

    <ENDIANNESS> BIG_ENDIAN </ENDIANNESS>

    <STRUCT_ITEMS>

        <ARRAY_COUNT_UINT16>
            <ARRAY_FIELD> THE_ARRAY </ARRAY_FIELD>
        </ARRAY_COUNT_UINT16>

        <FIELD_ARRAY>
            <FIELD_NAME> "THE_ARRAY" </FIELD_NAME>
            <ARRAY_ITEM>
                <FIELD_UINT16>
                    <FIELD_NAME> "ITEM" </FIELD_NAME>
                    <DISPLAY_FORMAT> DECIMAL </DISPLAY_FORMAT>
                </FIELD_UINT16>
            </ARRAY_ITEM>
        </FIELD_ARRAY>

    </STRUCT_ITEMS>
```

</DEFINE_RESOURCE_TYPE>

Name Format Requirements

The names for enum types, enum value labels, struct types, and resource types are required to be a certain format:

- The names are allowed to use alphanumeric characters (lower case letters 'a' through 'z', uppercase letters 'A' through 'Z', and numbers '0' through '9') and the underscore character ('_').
- The names must begin with a letter (lower case letters 'a' through 'z', uppercase letters 'A' through 'Z') or with the underscore character ('_'). A name cannot have a number as the first character.
- Names are case-sensitive.
- Enum type names, struct type names, and resource type names must be globally unique. Field names must be unique within the defining data scope.
- For consistency with Palm OS Resource Type descriptions, it is recommended that all uppercase names be used, as this convention results in consistent appearance in the XML file. However this convention is purely a matter of style.

Data Model Objects

The following data model objects can be used for resource type and struct descriptions. There are three types of data model objects:

Data field items: Data field items define a named data field of a specific data type. When the resource data is declared, it must include a corresponding XML element with a data value of the corresponding data type. For example, if a `FIELD_UINT16` named "X" is declared, the resource data must use a value declared similarly to this: `<X> 42 </X>`

For more information about data field items, see "[Data Field Items](#)" on page 131.

Meta-data field items : Meta-data fields are data field whose values are automatically calculated by the resource compiler and thus do not appear explicitly in the resource data. For example, if you define an `ARRAY_COUNT_UINT16` item, then the resource compiler automatically calculates the number of elements in the array field referenced.

For more information about meta-data field items, see "[Meta-Data Field Items](#)" on page 143.

Format control items : Format control items are used to control the binary or compiler state when compiling and decompiling the binary data. For example, the `SET_ENDIANNESS` item is used to set whether integer data is output as big-endian or little-endian. The effect of format control items (items that affect the compiler state) is limited to the local resource or struct description and to any child field descriptions.

For more information about format control items, see "[Format Control Items](#)" on page 145.

Data Field Items

A data field description consists of the element identifying the field data type, optionally followed by a COMMENT_TEXT element, followed by a FIELD_NAME element, followed by any elements specific to the data type.

You can use the following data field items in your custom resource descriptions:

- [“Boolean Field”](#) on page 131
- [“Integer Fields”](#) on page 132
- [“Floating Point Fields”](#) on page 133
- [“Enum Fields”](#) on page 133
- [“String Fields”](#) on page 133
- [“String Table And Table String Fields”](#) on page 134
- [“Bitfield Fields”](#) on page 136
- [“Binary Fields”](#) on page 137
- [“Array Fields”](#) on page 138
- [“Struct Fields”](#) on page 139
- [“Variant Fields”](#) on page 139

Boolean Field

A boolean field is stored in binary as a single byte with values 0 or 1. In the resource data XML, the value must be specified as FALSE or TRUE.

Example field description

```
<FIELD_BOOLEAN>
  <FIELD_NAME> "MY_BOOLEAN" </FIELD_NAME>
</FIELD_BOOLEAN>
```

Example field data

```
<MY_BOOLEAN> TRUE </MY_BOOLEAN>
```

Integer Fields

An integer field specifies whether it is a signed or unsigned value and how many bits are used to store the value. The supported integer field types are:

- SINT8
- UINT8
- SINT16
- UINT16
- SINT32
- UINT32
- SINT64
- UINT64

Integer values are stored in the binary with the endianness specified by the resource type and compiler context.

You may optionally define the `DISPLAY_FORMAT` as `DECIMAL`, `HEXADECIMAL`, or `CHAR`. This element is used to format the data values for display when resource data is decompiled or viewed by tools such as `GenerateXRD`.

In the resource data XML, you may specify integer values as decimal, hexadecimal, or character constant values (regardless of the display format).

Example field description

```
<FIELD_SINT32>
  <FIELD_NAME> "MY_INTEGER" </FIELD_NAME>
  <DISPLAY_FORMAT> DECIMAL </DISPLAY_FORMAT>
</FIELD_SINT32>
```

Example field data

```
<MY_INTEGER> 3 </MY_INTEGER>
```

Floating Point Fields

A floating point field is specified including how many bits are used to store the value. The supported floating point field types are FIELD_FLOAT32 and FIELD_FLOAT64.

Floating point values are stored in the binary with the endianness specified by the resource type and compiler context.

Example field description

```
<FIELD_FLOAT32>  
  <FIELD_NAME> "MY_FLOAT" </FIELD_NAME>  
</FIELD_FLOAT32>
```

Example field data

```
<MY_FLOAT> 3.14 </MY_FLOAT>
```

Enum Fields

An enum field is specified as a field using an enum type previously defined. The integer binary size is specified by the enum type description. The value is stored in the binary with the endianness specified by the resource type and compiler context.

Example enum field description

```
<FIELD_ENUM>  
  <FIELD_NAME> "THE_DRAWING_OP" </FIELD_NAME>  
  <ENUM_TYPE> DRAWING_OPCODE </ENUM_TYPE>  
</FIELD_ENUM>
```

Example field data

```
<THE_DRAWING_OP> OPCODE_LINE_TO </THE_DRAWING_OP>
```

String Fields

A string field stored a text value. Text is stored in resource binary as a null-terminated string in the text encoding specified by the compilation target. There are two string formats supported: FIELD_STRING and FIELD_FIXED_STRING.

Custom Resources

Data Model Objects

FIELD_STRING strings are stored with variable length in the binary data. For example, the string "CAT" would be stored as 4 bytes (0x43, 0x41, 0x53, 0x00).

FIELD_FIXED_STRING values are stored as a predefined fixed size in the binary data. The size is specified with BUFFER_SIZE as the number of bytes for the string buffer size. This size must include 1 for the terminating null character. Any unused bytes in the string buffer are filled to 0.

For example, if the buffer size were specified as 6, the string "CAT" would be stored as 6 bytes (0x43, 0x41, 0x53, 0x00, 0x00, 0x00). If the string, including the terminating null character, does not fit in the buffer at compile time, then resource compiler gives an error message.

Example string field description

```
<FIELD_STRING>
  <FIELD_NAME> "MY_STRING" </FIELD_NAME>
</FIELD_STRING>
```

Example string field data

```
<MY_STRING> "Hello" </MY_STRING>
```

Example fixed string field description

```
<FIELD_FIXED_STRING>
  <FIELD_NAME> "MY_STRING2" </FIELD_NAME>
  <BUFFER_SIZE> 32 </BUFFER_SIZE>
</FIELD_FIXED_STRING>
```

Example fixed string field data

```
<MY_STRING2> "Hello" </MY_STRING2>
```

String Table And Table String Fields

Regular string fields are stored as inline data with variable length. The array data structures can not be randomly accessed without iterating over the data at run-time, which can complicate working resource types that use arrays of records.

To help with this common problem, the custom resource types data model provides table string fields and a string table data model object.

When you use table string fields, the offset of the string is output to the binary data rather than having the string data being placed inline at the field. The offset is relative to the beginning of the resource and is stored as a fixed size integer as specified by the table string field (either FIELD_TABLE_STRING_UINT16 or FIELD_TABLE_STRING_UINT32).

When you use table string fields, the resource type description also includes a string table data model object. The string table object is defined after all the table string fields in the resource. Typically the string table object will be defined as the last item in the custom resource type description.

NOTE: If the same string is defined by multiple table string fields, the string is only stored once in the string table and both table string fields will have the same offset.

Example custom resource type description using string table

```
<DEFINE_RESOURCE_TYPE NAME="MY_STRING_LIST">
  <RES_TYPE> 'Strs' </RES_TYPE>
  <ENDIANNESS> BIG_ENDIAN </ENDIANNESS>
  <STRUCT_ITEMS>
    <ARRAY_COUNT_UINT16>
      <ARRAY_FIELD> STRINGS </ARRAY_FIELD>
    </ARRAY_COUNT_UINT16>
    <FIELD_ARRAY>
      <FIELD_NAME> "STRINGS" </FIELD_NAME>
      <ARRAY_ITEM>
        <FIELD_TABLE_STRING_UINT16>
          <FIELD_NAME> "STRING" </FIELD_NAME>
        </FIELD_TABLE_STRING_UINT16>
      </ARRAY_ITEM>
    </FIELD_ARRAY>
    <STRING_TABLE>
  </STRING_TABLE>
  </STRUCT_ITEMS>
</DEFINE_RESOURCE_TYPE>
```

Custom Resources

Data Model Objects

Example custom resource data

```
<CUSTOM_RESOURCE RESOURCE_ID="1000">
  <CUSTOM_RES_TYPE> MY_STRING_LIST </CUSTOM_RES_TYPE>
  <CUSTOM_DATA>
    <STRINGS>
      <STRING> "CAT" </STRING>
      <STRING> "DOG" </STRING>
    </STRINGS>
  </CUSTOM_DATA>
</CUSTOM_RESOURCE>
```

Resulting resource binary data

```
00: 00 02          // count of strings
02: 00 0A          // offset of first string = "CAT"
04: 00 0E          // offset of second string = "DOG"
06: 00 00 00 08    // string table starts with size
0A: 43 41 54 00    // "CAT"
0E: 44 4F 47 00    // "DOG"
12:
```

Bitfield Fields

A bitfield field is an integer field where individual bits or sequences of the bits are treated as separate logical values. A bitfield field is specified including how many bits are used to store the value. The supported bitfield field types are:

- FIELD_BITFIELD_UINT8
- FIELD_BITFIELD_UINT16
- FIELD_BITFIELD_UINT32

The bitfield field is defined as a sequence of bitfield items. Bitfield items may be defined as BITFIELD_BOOL or BITFIELD_UINT n where n is a number from 1 to 31. Bitfield items are defined from most significant bit down to least significant bit.

Note that the conglomerate integer value is stored in the binary with the endianness specified by the resource type and compiler context. Bitfield items must be defined to use exactly the number of bits stored in the bitfield field.

Example field description

```
<FIELD_BITFIELD_UINT8>
  <FIELD_NAME> "ATTRIBUTES" </FIELD_NAME>
  <BITFIELDS>
    <BITFIELD_BOOL>
      <FIELD_NAME> "HI_BIT" </FIELD_NAME>
    </BITFIELD_BOOL>
    <BITFIELD_UINT3>
      <FIELD_NAME> "RESERVED" </FIELD_NAME>
    </BITFIELD_UINT3>
    <BITFIELD_UINT4>
      <FIELD_NAME> "CATEGORY" </FIELD_NAME>
    </BITFIELD_UINT4>
  </BITFIELDS>
</FIELD_BITFIELD_UINT8>
```

Example resource data

```
<ATTRIBUTES>
  <HI_BIT> TRUE </HI_BIT>
  <RESERVED> 0 </RESERVED>
  <CATEGORY> 2 </CATEGORY>
</ATTRIBUTES>
```

Binary Fields

Binary fields are used to store arbitrary raw binary data. There are two formats of binary supported: `FIELD_BINARY` and `FIELD_FIXED_BINARY`.

`FIELD_BINARY` fields are stored with variable length in the binary data. That is, you can specify any amount of data in the resource data description.

`FIELD_FIXED_BINARY` fields are stored with a specific length in the binary data. You specify the size in bytes with the `BUFFER_SIZE` element. In this case, the amount of data you specify in the resource data description must match exactly the expected size.

Binary data is specified in the custom resource data as hexadecimal encoded binary.

Example binary field description

```
<FIELD_BINARY>
  <FIELD_NAME> "A_BLOB" </FIELD_NAME>
</FIELD_BINARY>
```

Example binary field data

```
<A_BLOB>
  80 FF 00 01
</A_BLOB>
```

Example fixed-size binary field description

```
<FIELD_FIXED_BINARY>
  <FIELD_NAME> "B_BLOB" </FIELD_NAME>
  <BUFFER_SIZE> 8 </BUFFER_SIZE>
</FIELD_FIXED_BINARY>
```

Example field data

```
<B_BLOB>
  01 02 03 04 05 06 07 08
</B_BLOB>
```

Note that there is a related meta-data field object class named `BINARY_SIZE` which can be used to store the size of a binary data field in a separate integer field.

Array Fields

Array fields are used to store homogeneous arrays of data fields. The array item type is defined in the `ARRAY_ITEM` element. This element must contain a single data field specifier, which may be either a simple field type or a conglomerate field type such as a `FIELD_STRUCT`. Array items are not required to be homogeneous size in the binary data.

Example array field description

```
<FIELD_ARRAY>
  <FIELD_NAME> "MY_ARRAY" </FIELD_NAME>
  <ARRAY_ITEM>
    <FIELD_UINT16>
```

```
<FIELD_NAME> "CHAR_CODE" </FIELD_NAME>
<DISPLAY_FORMAT>
    HEXADECIMAL
</DISPLAY_FORMAT>
</FIELD_UINT16>
</ARRAY_ITEM>
</FIELD_ARRAY>
```

Example array data description

```
<MY_ARRAY>
  <CHAR_CODE> 0x1122 </CHAR_CODE>
  <CHAR_CODE> 0x3344 </CHAR_CODE>
</MY_ARRAY>
```

Note that there is a related meta-data field object class named `ARRAY_COUNT` which can be used to store the number of items in an array in a separate integer field.

Struct Fields

A struct field is used to define a field with data type corresponding to a previously defined struct type. The struct type is specified with the `STRUCT_TYPE` element.

Example struct field description

```
<FIELD_STRUCT>
  <FIELD_NAME> "POINT" </FIELD_NAME>
  <STRUCT_TYPE> POINT_TYPE </STRUCT_TYPE>
</FIELD_STRUCT>
```

Example struct data description

```
<POINT>
  <X> 20 </X>
  <Y> 30 </Y>
</POINT>
```

Variant Fields

A variant field is used to define a data description which varies depending on the value of a key field. The variant field specifies the key field (which must have been previously defined in the same

Custom Resources

Data Model Objects

scope) and a list of key values and corresponding data descriptions. You define the data description for each variant by using a `STRUCT_ITEMS` element in the same fashion as a `DEFINE_STRUCT_TYPE` description. Note that the data description for a variant may be empty.

Example of custom resource type using inline descriptions for field variants

```
<DEFINE_RESOURCE_TYPE NAME="TEST_VARIANT">
  <RES_TYPE> 'TVar' </RES_TYPE>
  <ENDIANNESS> BIG_ENDIAN </ENDIANNESS>
  <STRUCT_ITEMS>

    <FIELD_UINT32>
      <FIELD_NAME> "FORMAT" </FIELD_NAME>
      <DISPLAY_FORMAT> CHAR </DISPLAY_FORMAT>
    </FIELD_UINT32>

    <FIELD_UINT16>
      <FIELD_NAME> "VERSION" </FIELD_NAME>
      <DISPLAY_FORMAT> DECIMAL </DISPLAY_FORMAT>
    </FIELD_UINT16>

    <FIELD_UINT16>
      <FIELD_NAME> "RESERVED" </FIELD_NAME>
      <DISPLAY_FORMAT> DECIMAL </DISPLAY_FORMAT>
    </FIELD_UINT16>

    <FIELD_VARIANT>
      <FIELD_NAME> "DATA" </FIELD_NAME>
      <KEY_FIELD> VERSION </KEY_FIELD>
      <VARIANTS>

        <VARIANT>
          <KEY_VALUE> 1 </KEY_VALUE>
          <STRUCT_ITEMS>
            </STRUCT_ITEMS>
          </VARIANT>

        <VARIANT>
          <KEY_VALUE> 2 </KEY_VALUE>
          <STRUCT_ITEMS>
            <FIELD_UINT32>
              <FIELD_NAME> "X" </FIELD_NAME>
            </FIELD_UINT32>
          </STRUCT_ITEMS>
        </VARIANT>
      </VARIANTS>
    </FIELD_VARIANT>
  </STRUCT_ITEMS>
</DEFINE_RESOURCE_TYPE>
```



```
        </VARIANT>

    </VARIANTS>
</FIELD_VARIANT>

</STRUCT_ITEMS>
</DEFINE_RESOURCE_TYPE>
```

Example of corresponding resource data with 'VERSION = 1'

```
<CUSTOM_RESOURCE RESOURCE_ID="4101">
  <CUSTOM_RES_TYPE> TEST_VARIANT </CUSTOM_RES_TYPE>
  <CUSTOM_DATA>
    <FORMAT> 'TEST' </FORMAT>
    <VERSION> 1 </VERSION>
    <RESERVED> 0 </RESERVED>
    <DATA>
      </DATA>
    </CUSTOM_DATA>
  </CUSTOM_RESOURCE>
```

Example of corresponding resource data with 'VERSION = 2'

```
<CUSTOM_RESOURCE RESOURCE_ID="4102">
  <CUSTOM_RES_TYPE> TEST_VARIANT </CUSTOM_RES_TYPE>
  <CUSTOM_DATA>
    <FORMAT> 'test' </FORMAT>
    <VERSION> 2 </VERSION>
    <RESERVED> 0 </RESERVED>
    <DATA>
      <X> 0x12345678 </X>
    </DATA>
  </CUSTOM_DATA>
</CUSTOM_RESOURCE>
```

Example of custom resource type using separate structs for field variants

```
<DEFINE_RESOURCE_TYPE NAME="TEST_SILK_SCREEN">
  <RES_TYPE> 'silk' </RES_TYPE>
  <ENDIANNESS> BIG_ENDIAN </ENDIANNESS>
  <STRUCT_ITEMS>
    <FIELD_UINT16>
      <FIELD_NAME> "VERSION" </FIELD_NAME>
      <DISPLAY_FORMAT> DECIMAL </DISPLAY_FORMAT>
    </FIELD_UINT16>
```

Custom Resources

Data Model Objects

```
<FIELD_VARIANT>
  <FIELD_NAME> "VERSIONED_DATA" </FIELD_NAME>
  <KEY_FIELD> VERSION </KEY_FIELD>
  <VARIANTS>

    <VARIANT>
      <KEY_VALUE> 1 </KEY_VALUE>
      <STRUCT_ITEMS>
        <FIELD_STRUCT>
          <FIELD_NAME>
            "SILK_SCREEN_V1 "
          </FIELD_NAME>
          <STRUCT_TYPE>
            SILK_SCREEN_V1
          </STRUCT_TYPE>
        </FIELD_STRUCT>
      </STRUCT_ITEMS>
    </VARIANT>

    <VARIANT>
      <KEY_VALUE> 2 </KEY_VALUE>
      <STRUCT_ITEMS>
        <FIELD_STRUCT>
          <FIELD_NAME>
            "SILK_SCREEN_V2 "
          </FIELD_NAME>
          <STRUCT_TYPE>
            SILK_SCREEN_V2
          </STRUCT_TYPE>
        </FIELD_STRUCT>
      </STRUCT_ITEMS>
    </VARIANT>

  </VARIANTS>
</FIELD_VARIANT>

</STRUCT_ITEMS>
</DEFINE_RESOURCE_TYPE>
```

Meta-Data Field Items

You can use the following meta-data field items in your custom resource descriptions:

- “[Array Count Fields](#)” on page 143
- “[Binary Size Fields](#)” on page 144

Array Count Fields

An array count field is used to define an integer field containing the number of elements in the referenced array field. An array count field is specified including how many bits are used to store the value. The supported array count field types are:

- ARRAY_COUNT_UINT8
- ARRAY_COUNT_UINT16
- ARRAY_COUNT_UINT32

Array count values are stored in the binary with the endianness specified by the resource type and compiler context.

Note that array count values are calculated by the resource compiler at compile time and thus do not appear in the resource data explicitly.

Note also that only one array count object is allowed per array field.

For the purposes of viewing and decompiling data, the array count object must occur before the corresponding array. If an array count object is not specified, arrays are assumed to use all subsequent data in the resource binary.

Example array count field description

```
<ARRAY_COUNT_UINT16>
  <ARRAY_FIELD> MY_ARRAY </ARRAY_FIELD>
</ARRAY_COUNT_UINT16>
```

Binary Size Fields

A binary size field is used to define an integer field containing the size in bytes of the referenced binary field. A binary size field is specified including how many bits are used to store the value. The supported binary size field types are:

- BINARY_SIZE_UINT8
- BINARY_SIZE_UINT16
- BINARY_SIZE_UINT32

Binary size values are stored in the resource binary with the endianness specified by the resource type and compiler context.

Note that binary size values are calculated by the resource compiler at compile time and thus do not appear in the resource data explicitly.

Note also that only one binary size object is allowed per binary field.

For the purposes of viewing and decompiling data, the binary size object must occur before the corresponding binary field. If a binary size object is not specified, binary fields are assumed to use all subsequent data in the resource binary.

Example binary size field description

```
<BINARY_SIZE_UINT16>  
  <BINARY_FIELD> A_BLOB </BINARY_FIELD>  
</BINARY_SIZE_UINT16>
```

Format Control Items

You can use the following format control items in your custom resource descriptions:

- “[Align](#)” on page 145
- “[Set Endianness](#)” on page 146

Align

The `ALIGN` object is used to pad the binary data stream to a certain alignment. This is done by outputting bytes with value 0 if necessary until the binary data offset is a multiple of the specified value.

Example alignment field description

```
<ALIGN> 2 </ALIGN>
```

Example of ALIGN in a resource type description

```
<DEFINE_RESOURCE_TYPE NAME="TEST_ALIGN">
  <RES_TYPE> 'TAIn' </RES_TYPE>
  <ENDIANNESS> BIG_ENDIAN </ENDIANNESS>
  <STRUCT_ITEMS>

    <FIELD_SINT8>
      <FIELD_NAME> "X" </FIELD_NAME>
    </FIELD_SINT8>

    <ALIGN> 2 </ALIGN>

    <FIELD_UINT16>
      <FIELD_NAME> "COUNT" </FIELD_NAME>
    </FIELD_UINT16>

  </STRUCT_ITEMS>
</DEFINE_RESOURCE_TYPE>
```

In this example, field X is a single byte at offset 0, thus advancing the binary data offset to 1. Because this is not a multiple of 2, the `ALIGN` description then emits a pad byte to advance the binary offset to 2. The field `COUNT` is then a 2-byte integer at offset 0, thus advancing the binary data offset and total resource length to 4.

Set Endianness

The SET_ENDIANNES element is used to specify the format of multi-byte integer and floating point values in the resource binary data. The value may be specified as BIG_ENDIAN or LITTLE_ENDIAN. The endianness format is normally specified by the resource type and used consistently for the entire resource, but can be overridden for individual fields or structs if desired.

The endianness change is restricted to the scope of the resource type or struct type.

Example endianness field description

```
<SET_ENDIANNES> LITTLE_ENDIAN </SET_ENDIANNES>
```

Example of SET_ENDIANNES in a resource type description

```
<DEFINE_RESOURCE_TYPE NAME="TEST_ENDIANNES">
  <RES_TYPE> 'TEnd' </RES_TYPE>
  <ENDIANNES> BIG_ENDIAN </ENDIANNES>
  <STRUCT_ITEMS>

    <FIELD_UINT16>
      <FIELD_NAME> "N0" </FIELD_NAME>
    </FIELD_UINT16>

    <SET_ENDIANNES> LITTLE_ENDIAN </SET_ENDIANNES>

    <FIELD_UINT16>
      <FIELD_NAME> "N1" </FIELD_NAME>
    </FIELD_UINT16>

    <FIELD_UINT16>
      <FIELD_NAME> "N2" </FIELD_NAME>
    </FIELD_UINT16>

  </STRUCT_ITEMS>
</DEFINE_RESOURCE_TYPE>
```

In this example, the resource type is specified as big-endian. Thus the first integer value N0 is written to the binary data as big-endian. This is next overridden by the SET_ENDIANNES specifier so that the subsequent N1 and N2 fields are written to the binary data as little-endian.

Index

Numerics

1.5X density 30, 40, 97
1X density 30, 40, 97
2X density 30, 40, 97
3X density 30, 40, 97

A

'aalt' 24
'abmp' 39
'abyt' 43
'aalt' 45
'adwd' 46
'afnt' 51
'afnx' 47
'afrm' 54
'aint' 113
alert
 category 22
 definition 24
alert dialog 24
ALERT_TYPE 24
'alpf' 34
ALPF_BUILD_OS_VERSION 35
ALPF_FLAG_NO_OVERLAY 37
ALPF_FLAG_NOTIFY_DB_CHANGE 37
ALPF_FLAG_NOTIFY_FIND 36
ALPF_FLAG_NOTIFY_INSTALL 36
ALPF_FLAG_NOTIFY_RESET 36
ALPF_FLAG_NOTIFY_TIME_CHANGE 36
ALPF_FLAGS 36
ALPF_MIN_OS_VERSION 35
ALPF_STACK_SIZE 36
ALPF_VERSION 35
'amnu' 100
app info
 definition 33
APP_DEFAULT_CATEGORY_RESOURCE 26
APP_EXTENDED_PREFS_RESOURCE 27
APP_ICON_BITMAP_RESOURCE 28
APP_ICON_NAME_RESOURCE 33
APP_INFO_STRINGS_RESOURCE 34
APP_LAUNCH_PREFS_RESOURCE 35
APP_PREFS_RESOURCE 38

APP_VERSION_RESOURCE 39
appl database type 18
application default category
 category 21
 definition 26
application extended preferences
 category 21
 definition 26
application icon bitmap
 category 21
 definition 27
application icon name
 category 21
 definition 33
application information strings
 category 21
 definition 33
application launch preference
 definition 34
application launch preferences
 category 21
application preferences
 category 21
 definition 38
application resource 21
application version
 category 21
 definition 39
'aprf' 38
attribute 15
 comment 8
 locale 7
 overlay status 8
 resource id 6
AUTO_SHIFT 70
'awrd' 118
'axpf' 26

B

BACKGROUND_BITMAP_ID 68, 94
backslash character 13
binary data 10
binary_data 15
BIT_DEPTH 29
BITMAP 28, 40

- bitmap 62
 - category 22
 - definition 39
 - high density 32
- bitmap color matching 42
- bitmap font 51
- bitmap image file 31
- BITMAP_DATA 29, 42
- BITMAP_DENSITY 30, 40
- BITMAP_FILE 29, 42
- BITMAP_ID 62, 75, 77, 79
- BITMAP_RESOURCE 39
- BITMAPS 28, 39
- black-and-white image 39
- 'BLST' 43
- BLUE 46
- boolean 15
- boolean value element 10
- BOUNDS 54, 63, 65, 67, 69, 72, 74, 76, 78, 81, 84, 85, 87, 89, 91, 93, 95
- bprc file 7
- button 63
 - graphic 74
 - graphic repeating 78
- button frame 64
- button text 64
- BUTTON_FRAME 64, 75, 79, 89
- BUTTONS 25
- byte integer list
 - category 22
 - definition 43
- BYTE_LIST_RESOURCE 43

C

- calculating object widths 27
- 'cert' 44
- character references 12
- check box 65
- 'code' 106
- code signing
 - definition 44
- CODE_CERT_RESOURCE 44
- CODE_SIGN_RESOURCE 45
- COLOR 45

- color image 39
- color matching 42
- color table 45
 - category 22
 - definition 45
- COLOR_TABLE 45
- COLOR_TABLE_RESOURCE 45
- comment attribute 8
- comment text element 16
- COMMENT_TEXT 16
- COMPRESSION 29, 41
- confirmation dialog 24
- CONFIRMATION_ALERT 24
- CONTROL_ID 83
- country codes 7
- creating an Edit menu 102
- custom resource
 - data model objects 130
 - definition 122
 - format control 145
 - meta-data field 143
 - name format 129
- custom resource type element 120
- custom resources 119
- custom types files
 - definition 123
- CUSTOM_DATA 122
- CUSTOM_RES_TYPE 122
- CUSTOM_RESOURCE 122
- CUSTOM_TYPES_FILE 123
- CUSTOM_TYPES_FILES 123

D

- DATA 10
- 'data' 106
- data element 10
- data file element 11
- data types 9, 15
- DATA_FILE 11, 106
- database change notification 37
- database header element
 - example 5
 - overview 4
- DATABASE_HEADER 17

DB_APP_INFO 19
DB_CREATOR 18
DB_FLAG_BACKUP 17
DB_FLAG_COPY_PROTECT 18
DB_FLAG_HIDDEN 18
DB_FLAG_RESET 17
DB_MOD_NUM 18
DB_NAME 17
DB_SORT_INFO 19
DB_TYPE 18
DB_UNIQUE_ID 18
DB_VERSION 18
DEFAULT_BUTTON 25, 55
DEFAULT_ITEM 43, 47
DEFINE_ENUM_TYPE 120, 124
DEFINE_RESOURCE_TYPE 120, 127
DEFINE_STRUCT_TYPE 120, 126
defining a confirmation dialog 24
defining a warning dialog 24
defining an alert dialog 24
defining an error dialog 24
defining an information dialog 24
defining custom resources 119
developer tools x
'DLST' 46
double density 30, 40, 97
double quotation mark character 13
dword integer list
 category 22
 definition 46
DWORD_LIST_RESOURCE 47
DYNAMIC_SIZE 70

E

Edit menu 102
EDITABLE 69, 95
element 15
 boolean value 10
 comment text 16
 data 10
 data file 11
 database header 4
 enum 10
 four-character code 11

 integer number 12
 palm os resource file 4, 17
 quoted text 12
 text 16
ENABLED 63, 66, 68, 75, 77, 78, 84, 86, 88, 91, 93
ENDIANNESS 128
enum 15
enum description 124
enum element 10
ENUM_DATA_TYPE 124
ENUM_ITEMS 124
error dialog 24
ERROR_ALERT 24
escape sequence 13
ext_file 15
EXTENDED 73
extended font
 definition 47
external file reference 11
external image file 31, 43

F

false enumerated value 10
feedback slider 67
field 69
find notification 36
'fnav' 58
font resource
 definition 51
FONT_ASCENT 47, 51
FONT_DENSITY 48
FONT_DESCENT 47, 52
FONT_EXTENDED_ITEM 48
FONT_EXTENDED_ITEMS 48
FONT_EXTENDED_RESOURCE 47
FONT_GLYPH 48, 52
FONT_GLYPH_CODE 48, 52
FONT_GLYPH_IMAGE 48, 52
FONT_GLYPHS 48, 52
FONT_ID 64, 66, 71, 80, 82, 84, 86, 88, 92
FONT_MISSING_GLYPH 48, 52
FONT_RESOURCE 51
FONT_TYPE 47, 51
form

- category 22
- definition 54
- form bitmap 62
- form button 63
- form check box 65
- form feedback slider 67
- form field 69
- form gadget 72
- form graffiti state 74
- form graphic button 74
- form graphic push button 76
- form graphic repeating button 78
- form label object 80
- form list object 81
- form navigation
 - category 22
- form object
 - bitmap 62
 - button 63
 - check box 65
 - feedback slider 67
 - field 69
 - gadget 72
 - graffiti state 74
 - graphic button 74
 - graphic push button 76
 - graphic repeating button 78
 - label 80
 - list 56, 81
 - pop-up 83
 - pop-up trigger 83
 - push button 85
 - repeating button 87
 - scroll bar 89
 - selector trigger 91
 - slider 93
 - table 95
 - title 96
- form pop-up 83
- form pop-up trigger 83
- form push button 85
- form repeating button 87
- form scroll bar 89
- form selector trigger 91
- form slider 93

- form table 95
- form title 96
- FORM_BITMAP 62
- FORM_BUTTON 63
- FORM_CHECKBOX 65
- FORM_FEEDBACK_SLIDER 67
- FORM_FIELD 69
- FORM_GADGET 72
- FORM_GRAFFITI_STATE 74
- FORM_GRAPHIC_BUTTON 74
- FORM_GRAPHIC_PUSH_BUTTON 76
- FORM_GRAPHIC_REPEATING_BUTTON 78
- FORM_ID 54
- FORM_LABEL 80
- FORM_LIST 81
- FORM_NAVIGATION_RESOURCE 58
- FORM_OBJECTS 56
- FORM_POPUP 83
- FORM_POPUP_TRIGGER 83
- FORM_PUSH_BUTTON 85
- FORM_REPEATING_BUTTON 87
- FORM_RESOURCE 54
- FORM_SCROLLBAR 89
- FORM_SELECTOR_TRIGGER 91
- FORM_SLIDER 93
- FORM_TABLE 95
- FORM_TITLE 96
- format control 145
- four_character_code 15
- four-character code element 11
- frame of a button 64
- 'frmb' 116
- 'fttf' 114

G

- gadget 72
- graffiti shift indicator 74
- graffiti state 74
- graphic button 74
- graphic family
 - category 22
- graphic repeating button 78
- GRAPHIC_FAMILY_RESOURCE 97

grayscale image 39
GREEN 45
GROUP_ID 66, 77, 87

H

HAS_SCROLLBAR 71
heap space setting 38
HEIGHT 67, 69, 73, 75, 77, 78, 84, 86, 91
HELP_ID 25, 55
high density bitmap 32

I

ID 63, 65, 67, 69, 72, 74, 76, 78, 80, 81, 83, 85, 87, 89, 91, 93, 95
ignore overlays 37
image file 31, 43
INDEX 45
information dialog 24
INFORMATION_ALERT 24
inline bitmap data 31
installation notification 36
integer 16
integer number element 12

J

JUSTIFICATION 70

L

label object 80
language codes 7
large application icon 28
LEFT 67, 69, 72, 74, 76, 78, 84, 85, 91
LEFT_ANCHOR 64, 75, 79, 84, 88, 92
line feed character 13
list object 81
LIST_ID 83
LIST_ITEMS 82
locale attribute 7
LOCATION 62, 74, 80
look for overlays 37

M

MAX_CHARS 71
MAX_VALUE 68, 90, 94
MAX_VISIBLE_LINES 71
'MBAR' 100
MENU 100
menu bar
 category 22
 definition 99
MENU_BAR_RESOURCE 100
MENU_ID 55
MENU_ITEM 102
MENU_ITEMS 101
MENUS 100
MESSAGE 25
meta-data field 143
MIDI
 definition 104
midi
 category 22
MIDI_FILE 104
MIDI_RAW_DATA 104
MIDI_RESOURCE 104
MIN_HEAP_SPACE 38
MIN_VALUE 68, 90, 93
MODAL 55

N

'NFNT' 51
'nfnt' 47
NO_OVERLAY 27
normal density 30, 40, 97
notify
 of database change 37
 of device reset 36
 of find search 36
 of installation 36
 of time change 36
NUM_VIS_ITEMS 82
number values 12
NUMERIC 71
numeric character references 12

O

one-and-one-half density 30, 40, 97
oprc file 7
overlay
 definition 105
overlay resource file 7
overlay status attribute 8
overlay use 37
OVERLAY_RESOURCE 105
'ovly' 105

P

PAGE_SIZE 68, 90, 94
Palm OS Developer Suite x
palm os resource file element 4, 17
PALMOS_CUSTOM_RESOURCE_TYPES 120
PALMOS_RESOURCE_FILE 17
pop-up object 83
pop-up trigger 83
prc database type 18
'pref' 38
PREFIX 113
PREFS_GENERATE_HEADER 19
PREFS_HEADER_FILE 19
PRIORITY 38
priority setting 38
push button 85
 graphic 76

Q

quoted text element 12
quoted_text 16
QVGA density 30, 40, 97

R

raw
 definition 106
raw resource
 category 22
RAW_RESOURCE 106
RED 45
repeating button 87
RES_DATA 106

RES_TYPE 106, 128
reset notification 36
resource element 19
resource id attribute 6
resource type 127
resource_id 16
ROW_BYTES 29, 40
RUNTIME_WIDTH_CALCS 27

S

SAVE_BEHIND 55
'scdb' 107
schema database
 definition 107
SCHEMA_DATABASE_RESOURCE 107
scroll bar 89
SDB_CATEGORIES 110
SDB_CATEGORY 110
SDB_CATEGORY_EDITABLE 110
SDB_CATEGORY_ID 110, 112
SDB_CATEGORY_NAME 110
SDB_COLUMN 108
SDB_COLUMN_ATTRIBUTE_NONSYNABLE 109
SDB_COLUMN_ATTRIBUTE_WRITEABLE 109
SDB_COLUMN_ATTRIBUTES 109
SDB_COLUMN_ID 108, 111
SDB_COLUMN_NAME 108
SDB_COLUMN_PROPERTIES 109
SDB_COLUMN_PROPERTY 109
SDB_COLUMN_PROPERTY_DATA 110
SDB_COLUMN_PROPERTY_ID 109
SDB_COLUMN_SIZE 109
SDB_COLUMN_TYPE 109
SDB_COLUMNS 108
SDB_DB_CREATOR 107
SDB_DB_NAME 107
SDB_DB_TYPE 107
SDB_DB_USER_NAME 107
SDB_DB_VERSION 107
SDB_HEADER 107
SDB_RECORD 111
SDB_RECORD_ATTRIBUTE_READONLY 111

SDB_RECORD_ATTRIBUTE_SECRET 111
SDB_RECORD_ATTRIBUTES 111
SDB_RECORD_CATEGORIES 111
SDB_RECORD_COLUMN_DATA 111
SDB_RECORD_DATA 111
SDB_RECORD_DATA_VALUE 111
SDB_RECORD_ID 111
SDB_RECORDS 111
SDB_SCHEMA_ID 108, 111
SDB_SCHEMA_NAME 108
SDB_SCHEMAS 108
search for overlays 37
security certificate
 definition 44
security signature
 definition 44
SELECTED 66
SELECTED_BITMAP_ID 75, 77, 79
selector trigger 91
shift indicator 74
'sign' 45
single density 30, 97
SINGLE_LINE 70
slider 93
small application icon 28
soft constant
 category 22
 definition 113
SOFT_CONSTANT_RESOURCE 113
sound 115
special characters 12
specify heap space 38
specify priority 38
specify stack size 38
specifying a bitmap font 51
specifying a color table 45
specifying button text 64
stack size setting 38
STACK_SIZE 38
standard Edit menu 102
string
 category 22
 definition 114
string list

 category 22
 definition 113
STRING_LIST_RESOURCE 113
STRING_RESOURCE 114
STRINGS 34, 113
struct description 126
STRUCT_ITEMS 126, 128

T

table 95
TABLE_COLUMNS 95
TABLE_ROWS 96
'tAIB' 28
'taic' 26
'tAIN' 33
'Talt' 24
'Tbmp' 39
'tclt' 45
TEXT 16, 26, 33, 39, 64, 66, 80, 84, 86, 88, 92
text element 16
text value 12
text_element 16
'tFRM' 54
THUMB_BITMAP_ID 68, 94
time change notification 36
'tint' 113
TITLE 25
title 96
tools documentation x
 Introduction to Palm OS Tools x
 Language and Library Reference x
 Palm OS Debugger x
 Palm OS Resource Editor x
 Resource Tools x
 Virtual Phone xi
TOP 67, 69, 72, 74, 76, 78, 84, 86, 91
triple density 30, 40, 97
true enumerated value 10
truetype font
 definition 114
TRUETYPE_FONT_RESOURCE 114
'tSTL' 113
'tSTR' 114
TTFONT_FILE 115

TTFONT_RAW_DATA 114

'tver' 39

U

UNDERLINE 70

USABLE 54, 62, 63, 65, 67, 69, 73, 75, 77, 78, 80, 81,
84, 86, 88, 90, 91, 93

V

VALUE 44, 47, 68, 90, 93

VALUES 43, 47

variables 15

VERSION 27

VISIBLE 73, 100

W

warning dialog 24

WARNING_ALERT 24

'wave' 115

wave sound

category 22

definition 115

WAVE_FILE 115

WAVE_RAW_DATA 115

WAVE_SOUND_RESOURCE 115

white space 9

WIDTH 67, 69, 72, 75, 76, 78, 84, 86, 91

window constraints 116

category 22

WINDOW_CONSTRAINTS_RESOURCE 116

WINDOW_CREATE_FLAGS 116

WINDOW_HEIGHT_MAX 117

WINDOW_HEIGHT_MIN 117

WINDOW_HEIGHT_PREF 117

WINDOW_WIDTH_MAX 117

WINDOW_WIDTH_MIN 117

WINDOW_WIDTH_PREF 117

WINDOW_X_POS 117

WINDOW_Y_POS 117

word integer list

category 22

definition 118

WORD_LIST_RESOURCE 118

'wrld' 118

X

XML

definition 2

file header 4

resource descriptions 6

unicode requirements 3

XML resource element 19

'xprf' 26

XRD

data types 9

definition 3

XRT file 123