

*Java Management API
Help Developer's Guide*

spec release



Sun Microsystems Computer Company
A Sun Microsystems, Inc. Business
2550 Garcia Avenue
Mountain View, CA 94043 U.S.A.
415 960-1300 FAX 415 969-9131
Part No.: 802-7711-01
Revision C, September 1996

Copyright 1996 Sun Microsystems, Inc., 2550 Garcia Avenue, Mountain View, California 94043-1100 U.S.A. All rights reserved.

This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any.

Portions of this product may be derived from the UNIX[®] system, licensed from Novell, Inc., and from the Berkeley 4.3 BSD system, licensed from the University of California. UNIX is a registered trademark in the United States and other countries and is exclusively licensed by X/Open Company Ltd. Third-party software, including font technology in this product, is protected by copyright and licensed from Sun's suppliers.

RESTRICTED RIGHTS: Use, duplication, or disclosure by the U.S. Government is subject to restrictions of FAR 52.227-14(g)(2)(6/87) and FAR 52.227-19(6/87), or DFAR 252.227-7015(b)(6/95) and DFAR 227.7202-1(a).

Sun, Sun Microsystems, the Sun logo, SunSoft, and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the United States and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK[®] and Sun[™] Graphical User Interfaces were developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

THIS PUBLICATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.



Content

1. Introduction	1
Overview of the JMAPI Help System	1
Components of the Help System	2
Benefits	2
Open Systems, Open Browsers	3
Single Help Information Space	3
Automatic Integration	3
Search Engine	4
Java Ready	4
Standard Help System Behavior	4
How a JMAPI Application and Help System Interact	7
2. Authoring	9
Producing HTML Files	9
Creating the Master Help File	9
Creating New Help Files	11

Extending Existing Help Files	11
Tools and Aids	11
Standard Help Files	11
Help File Templates	12
Generated Files	12
Doc Generator	13
Importing Other Formats	13
The Help Authoring Interface (HAI)	13
Specifying an Index Entry	13
Specifying a Glossary Entry	16
Specifying a TOC Entry for Navigation	18
Specifying an Object TOC Entry	20
Specifying a Related Topics TOC Entry	21
Specifying an Extensions TOC Entry	22
Testing Your Document	23
3. Integration Guidelines	25
Process Overview	25
Required Application Help	26
Object Help	26
Object Property Book Information	28
Content Manager Help	30
Standard Information Structure	30
Help on Help	31
Overview	32

Requirements	33
Tasks	35
Key Concepts	36
Help on Pages.....	37
Error Messages.....	39
Writing Help on Custom Pages and Content Managers.....	39
Writing Help on Managed Objects	39
Linking Guidelines	40
From Application Pages	40
Between Help Files	40
Outside the Help JMAPI Information Space	40
Help File Naming and Directory Structure.....	40

Overview of the JMAPI Help System

At the highest level, the Java(TM) Management API (JMAPI) provides a set of Java classes that can be used to create Java-based system and network applications that can be displayed in a Java-enabled browser. The JMAPI Help System provides a means for application developers and help authors to build page-sensitive application help.

The primary objective of the Java Management API (JMAPI) Help System is to provide a general purpose help environment. General purpose means it is intended that JMAPI application developers and ISVs can use the help system in a range of ways to provide application online help. As an example, applications that comply with JMAPI standards for conformity could have help files that integrate seamlessly into one common help system shared by multiple applications. This means that multiple applications would share one common index, glossary, table of contents/navigator, and so on. Applications that do not comply with JMAPI standards for conformity could have help files activated and displayed within the JMAPI help system, but those help files are *not* integrated into one common help information space. Help for a non-conforming application simply has use of the help system, but does not reap benefits of integration such as consistency and related ease of use.

In keeping with the goal of a general-purpose help system, JMAPI help does not require any proprietary tools, scripts, or programs to produce or generate help files. While tools may come along to facilitate authoring of JMAPI help, they are not required. The only requirement is that help files be in HTML

format and include a set of JMAPI help authoring tags that can be parsed during installation time to produce an index, glossary, and table of contents / navigator.

Components of the Help System

The JMAPI Help System includes the following:

- **TOC/Navigator** — Displays a hierarchy of topics available in application help.
- **Doc Generator** — The `jmapidoc` program is available for generating index, glossary, or TOC / Navigator files for an application. This doc generator can also be used by help authors to prototype and test their work.
- **Standard Help Files** — These are help files that application developers and ISVs can pass to their end users. These primarily include information such as Help on the Help System, and Help on basic user interface components.
- **Help Templates** — These are HTML template files, including JMAPI help keywords and examples of their use.
- **Search Engine** — TBD

Benefits

The benefits of using the JMAPI Help System are many. Briefly, it provides:

- **Open system and open browser** — The help can be displayed on any system supporting Java and in any Web browser that is Java-enabled.
- **Single help information space** — Help files for JMAPI compliant applications will be integrated into a single help information space.
- **Open architecture** — Although help files can be integrated for JMAPI compliant applications, applications that are not JMAPI compliant can still access the help browser and display HTML help files in it.
- **Single source of navigation** — Help files for JMAPI compliant applications will be accessible from a single TOC / Navigator and Index. Any time a new JMAPI compliant application is installed, help files are parsed and the global TOC / Navigator and Index are updated.
- **Search engine** — Help files for JMAPI compliant applications will be subject to a single, Java-based search engine.

- **Java-ready** — Help files are in HTML, so Java applets can be easily incorporated.

Open Systems, Open Browsers

One of the obvious advantages of Java-based applications is that a single compiled application can run on multiple platforms. JMAPI and its help system go one step further. JMAPI applications and help can run across multiple browsers. You do not need to develop your application and its accompanying help for any one browser. This not to imply that your JMAPI-based applications and their help systems will run in every existing browser; they require a Java-enabled browser that supports frames. Fortunately, most of the popular browsers provide this capability. This minimizes the strategic concern for selecting a browser for your application and help. By using JMAPI help, your application is virtually browser independent.

Single Help Information Space

Another benefit of using the Java Management API is that applications from disparate vendors can look like, feel like, and work like a single, integrated application. The same is true of the help information. This means that multiple vendors can plug their respective application help into one integrated help information space. The result is that help for multiple applications can be accessed from a single TOC/Navigator and a single index. The scope of a search would be the entire set of integrated information, and one glossary would be common across compliant applications. This makes the presentation of help information consistent across multiple applications so that users can develop an understanding of the way JMAPI help information is organized and accessed. Users will, in effect, become JMAPI help literate.

Automatic Integration

When a JMAPI-based application is installed, the help files are parsed and several files used by the JMAPI Help System are generated. These include:

- **TOC/Navigator File** — Used by the TOC/Navigator applet that navigates the help information space from the top down.
- **Index File** — Invoked by the Index link on any help page.
- **Glossary File** — Invoked by the Glossary link on any help page.

- **Object TOC File** — A mini-TOC used within a help page to link to help files related to the application page. Refer to Figure 1-1.
- **Related Topics File** — Another mini-TOC used within a help page to link to help files related to the browser display. Refer to Figure 1-1.

For an application that complies with JMAPI standards for conformance, the related help files are integrated into one common information space; that is, there's one common TOC/Navigator, Index, Glossary, and so on. If an application is based on or extends an existing application, the help files are integrated in such a way as to highlight that relationship, with appropriate modifications and extensions to the base application.

Search Engine

TBD

Java Ready

JMAPI applications require a Java-enabled browser from which the Help System clones an application browser to display the help. As a result, any help file can include an HTML applet tag to invoke a Java applet from within a help file.

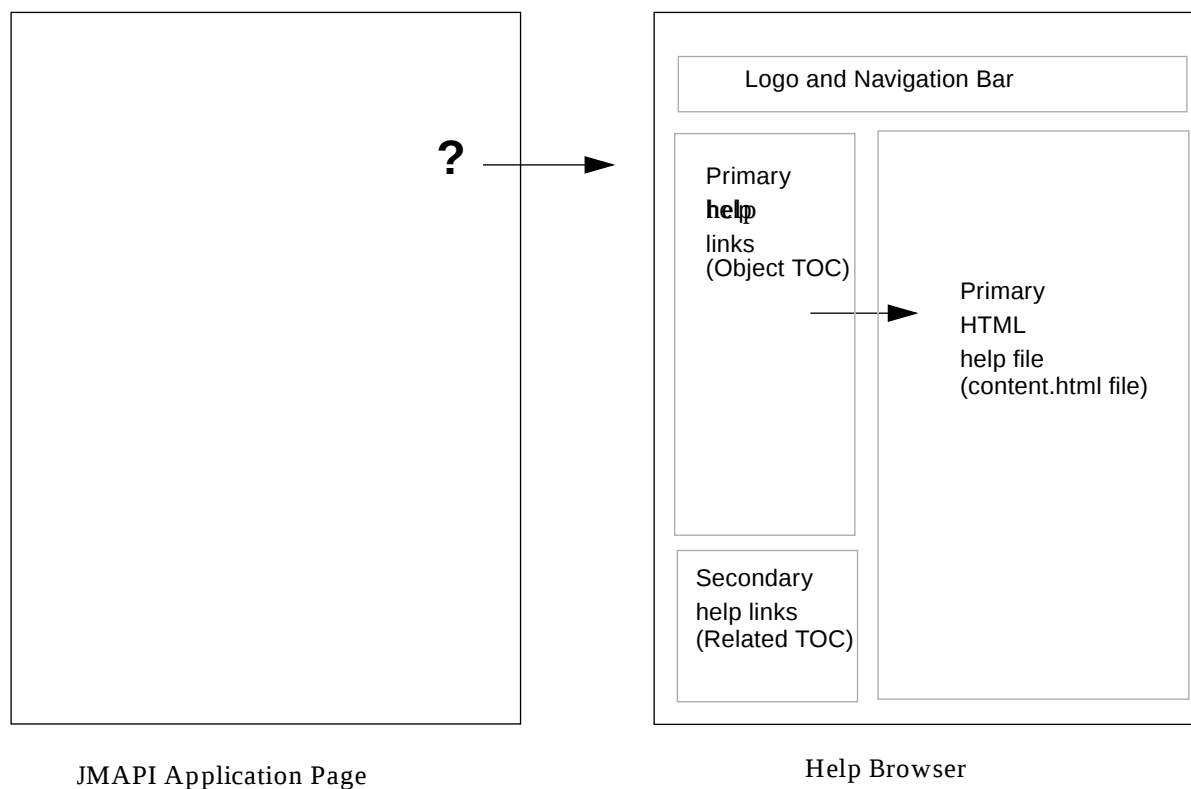
Standard Help System Behavior

Every page in a JMAPI application has a help button. When a user clicks on the help button, the JMAPI Help System will:

1. Open a new help browser cloned from the browser used to display the application or, if a help browser is already displayed, use the existing help browser.
2. Display a known help file. That help file varies depending on the UI element that activates the help system. To aid help authors, this document includes sections that specify which help files are required for which UI elements.

The next figure shows at a high level what happens when a user invokes help from a JMAPI application.

Figure 1-1 Help System Behavior



When help is invoked from the application page, a browser is cloned to be used specifically for displaying help. The help browser displays a set of links to the primary object help and related (secondary) help. All pertinent help files are available from the set of links. In this way, the user can select the help that is most relevant.

The next figure shows a prototype JMAPI help page.

Figure 1-2 Prototype Help Page

Help Page – The help page is specified by a single HTML template that defines the respective frames and headers.

Navigation Frame

ObjectTOC Frame – This page provides links to other pertinent help files.

Related Topics TOC Frame – This page provides links to secondary help topics such as help on the user interface, help on related objects, and so on.



Help Content Frame – This is the primary help file for the application.

How a JMAPI Application and Help System Interact

At a lower level, the interaction between a JMAPI application and the Help System is achieved by the application code including calls to the appropriate

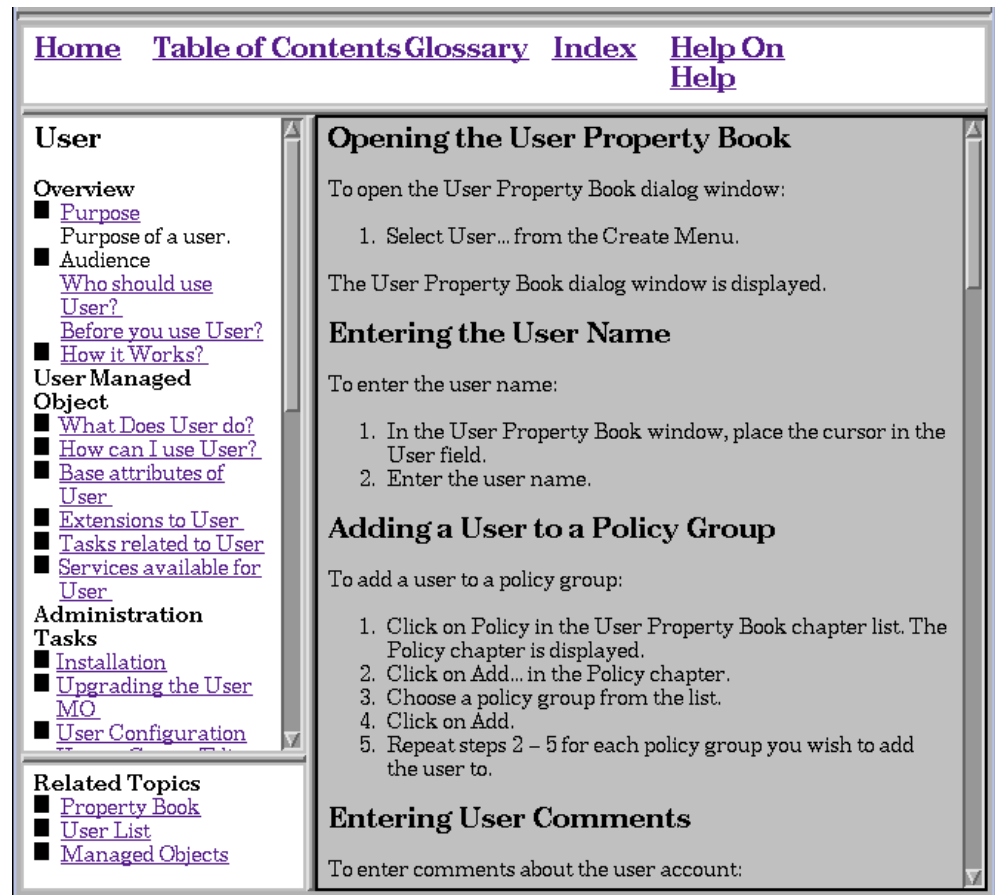
TBD

Producing HTML Files

The raw input to the JMAPI Help System is a set of HTML files. The JMAPI Help System does not require use of a particular tool to generate the help file, but it does require use of JMAPI help authoring tags within the HTML.

Creating the Master Help File

This file does not contain any help information, per se. Instead, it defines the navigation header and the frame structure of the help page. For example, look at the help page in the next figure.



A master help page would define the links in the two left frames and the name of the content file in the right frame. The master help page would also name the Object TOC and Related Topics files. The following HTML defines this help page. (Note that a JMAPI Help includes a template for duplicating this structure whenever needed.)

```
<HTML>
<HEAD>
</HEAD>
<title>JMAPI Help User MO Description</title>
<FRAMESET ROWS="60,88%">
```

```
<FRAME SCROLLING=NO NORESIZE SRC="./thetop.htm" TARGET="_top">
<FRAMESET COLS="200,65%">
<FRAMESET ROWS="70%,90">
<FRAME src="./test.htm" SCROLLING="YES" NORESIZE TARGET="content">
<FRAME SCROLLING=NO NORESIZE SRC="./related.htm" TARGET="_top">
</FRAMESET>
<FRAME src="./usercre2.htm" SCROLLING="YES" NORESIZE NAME="content">
</FRAMESET>
</FRAMESET>
</HTML>
```

Creating New Help Files

To create new help files, you can start with the help file templates provided with the JMAPI Help System. These files are available in *TBD_dir* directory.

Extending Existing Help Files

The application you're documenting may be an extension of capabilities already built into the system. In this case, you may need to modify or extend an existing help file to reflect changes to attributes or available services. In this case, you need to: To Be Supplied.

Tools and Aids

The JMAPI Help System comes with several tools and aids to assist help authors.

Standard Help Files

The JMAPI Help System comes with a set of standard help files that contain generic help text. These files can be made available to the end users of a JMAPI-based application. These files include:

- **Help on the help system** — Generic help on using the help system
- **Help on managed objects** — An overview of managed objects and services, which are the foundation of all JMAPI-based applications
- **Help on UI elements** — Generic help on JMAPI UI elements used by all applications.

All this information is generic to the JMAPI environment and does not need to be rewritten by help authors. It is likely that application developers or ISVs will create new managed objects and have their own implementations of the standard UI elements. These new object or elements will require their own help files.

Help File Templates

JMAPI help follows a pre-defined information structure. Help files need to share a common structure so that help files from different applications can be integrated together. In this way the structure is an aid in creating a common TOC/Navigator. This structure is embedded into the help templates provided with the JMAPI Help System. These help templates include the following:

- `object.html` — A template for the master help page.
- `object_content.html` — A template for a primary content file displayed when a user invokes help from the application or from the TOC/Navigator.
- `contentmgr.html` — A template for a help file on a content manager.
- `master_index.html` — An HTML template that specifies the index navigation bar and the actual index file (`index.html` and `index_navbar.html`, respectively).
- `master_glossary.html` — an HTML template that specifies the `glossary.html` and `glossary_navbar.html` files.
- `master_toc.html` — An HTML template that specifies the TOC/Navigator applet.

Generated Files

When a JMAPI-based application is installed on a system, the JMAPI Help System generates a number of files based on use of JMAPI authoring tags in the HTML help files. Specifically, the help system generates:

- `toc.html` — The global TOC used by the TOC/Navigator applet.
- `index.html` — The global index derived from all application help files.
- `index_navbar.html` — The A-Z navigation bar on top of the index help page.

- `glossary.html` — The global glossary derived from all application help files.
- `glossary_navbar.html` — The A-Z navigation bar on top of the glossary help page.
- `objecttoc.html` — The TOC displayed in a JMAPI help browser frame. Links in it can be used to navigate to new help content pages in the help browser.
- `objectrelated.html` — The TOC of secondary, related topics available for any given help. It contains links to generic JMAPI help files.

Doc Generator

`jmapidoc`

See Appendix A.

Importing Other Formats

TBD

The Help Authoring Interface (HAI)

The syntax for JMAPI authoring tags is as follows:

```
<!--  
  Tag (start...stop) Attributes    Explanation  
-->
```

Notice that all JMAPI authoring tags fall between standard HTML comment line. Any number of JMAPI authoring tags can be placed between the opening and closing HTML comment tags. If you put JMAPI authoring tags outside HTML comments, those tags may be displayed in the help page.

Specifying an Index Entry

Following is the syntax for an index entry:

```
<!--  
  <index>                Identifies an index entry.
```

label="string" Label to be displayed in index.

name=string Name of an index entry. If present, it indicates entry will have subentries and no href definition will be generated.

parent=string Parent name of a index entry. Entry will be indented under parent.

anchor="string" Anchor to apply to HREF definition. Is ignored if name is present.

-->

Examples:

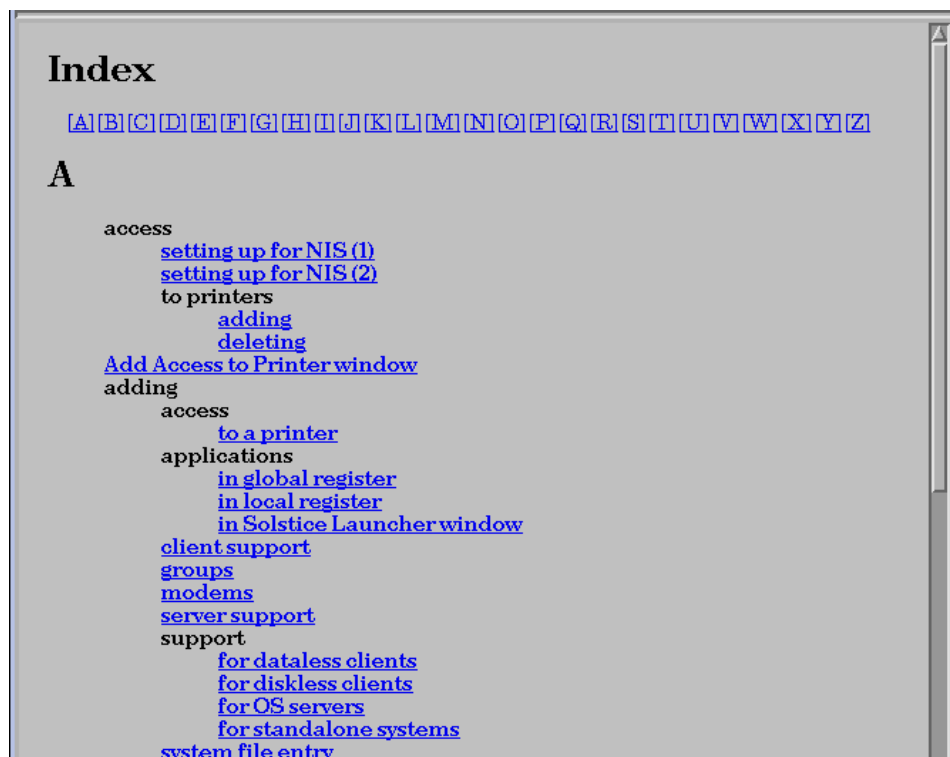
```
<!--
<index parent=access label="setting up for NIS">
<index name=printer parent=access label="to printers">
<index parent=printer label="adding">
<index parent=printer label="deleting">
<index label="Add Access to Printer window">
<index name=adding label="adding">
<!--
<index parent=printerAccess label="to a printer">
<index name=applications parent=adding label="applications">
<index parent=applications label="in global register">
<index parent=applications label="in local register">
-->

<!--
<index parent=applications label="in Solstice Launcher window">
-->

<!--
<index parent=adding label="client support">
-->

<!--
<index parent=adding label="groups">
<index parent=adding label="modems">
<index parent=adding label="server support">
-->
```

This would produce an index that looks like this:



Specifying a Glossary Entry

Following is the syntax for a glossary entry:

```
<glossary>...</glossary> Used to identify a glossary entry
    term="string"    Term displayed and defined in the glossary.
```

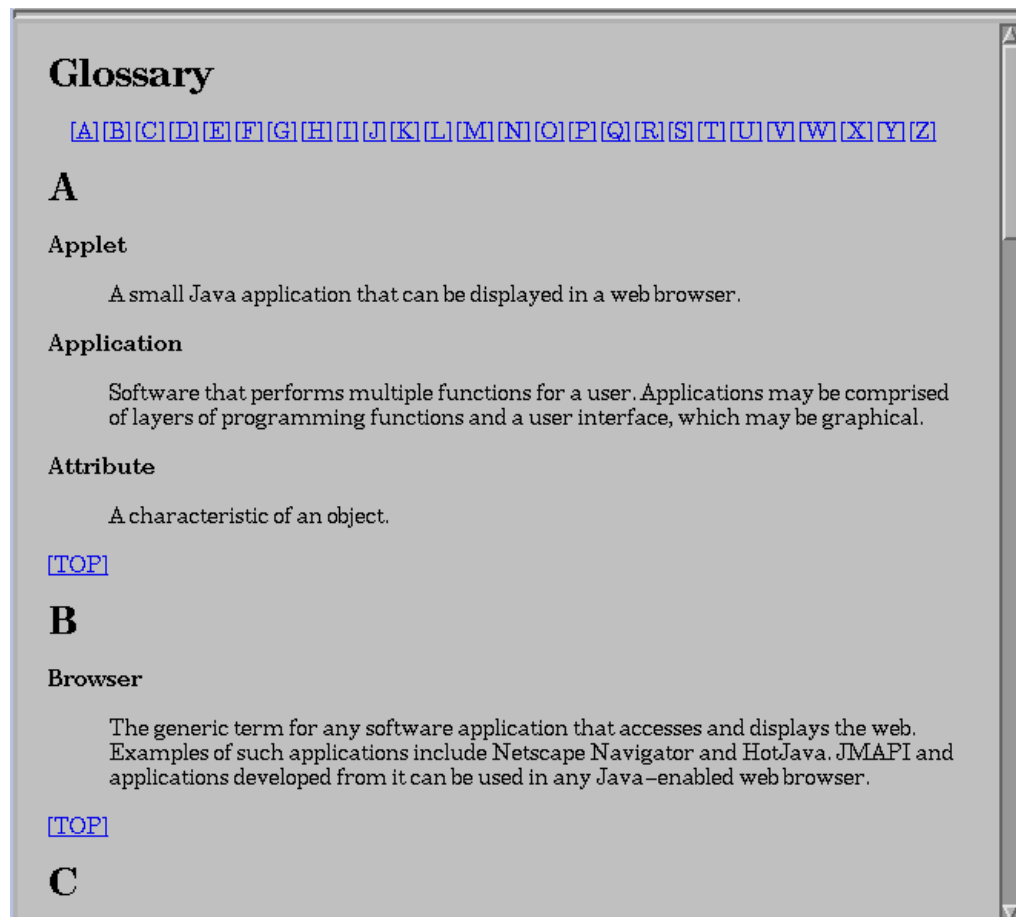
Examples:

```
<!--
<GLOSSARY TERM=Java>
A simple, robust, object-oriented, platform-independent, multi-
threaded, dynamic, general-purpose programming environment best
suited for creating applets and applications for the Internet,
intranets, and any other complex, distributed network.
</GLOSSARY>
-->

<!--
<GLOSSARY TERM=Java-enabled>Supporting Java.</GLOSSARY>
-->

<!--
<GLOSSARY TERM=Browser>
The generic term for any software application that accesses and
displays the web. Examples of such applications include Netscape
Navigator and HotJava. JMAPI and applications developed from it can
be used in any Java-enabled web browser.
</GLOSSARY>
-->
```

These tags would produce a glossary that looks like this:



Specifying a TOC Entry for Navigation

Following is the syntax for a TOC entry:

```
<!--
<toc>           Used to identify a table of contents entry.
    label="string" Label to be displayed in table of contents.
    name="string"  Name of a table of contents entry. Existence indicates
                    entry will have subentries and no HREF definition will be
                    generated.
    parent="string" Parent name of a table of contents entry. Entry will be
                    indented under the parent name.
    anchor="string" Anchor to apply to HREF definition. This is ignored if
                    name is present.
    sortkey="string" Alternate string used in comparing table of contents
                    entries. If present, items will be sorted in order of the
                    sortkey instead of the label.
    href="URL"     The URL of a hypertext reference.
-->
```

The JMAPI Help System derives the appropriate HREF for a TOC entry based on the file's name and directory location. Notice that an HREF tag is available, but the intent of this is for use with files that point outside of the JMAPI information space.

Examples:

```
<!--
<toc name=overview label="Overview">
-->

<!--
<toc parent=overview name=purpose sortkey="A" label="Purpose">
-->

<!--
<toc parent=purpose label="What Is Solstice WorkShop?">
-->
```

```
<!--
<toc parent=overview name=audience sortkey="B" label="Audience">
-->

<!--
<toc parent=audience label="Who Should Use Solstice WorkShop">
-->

<!--
<toc parent=audience label="Before You Use Solstice WorkShop">
-->

<!--
<toc parent=overview name=howItWorks sortkey="C" label="How It
Works">
-->

<!--
<toc parent=howItWorks label="What Can You Do With Solstice
WorkShop?">
-->

<!--
<toc name=requirements label="Requirements">
-->

<!--
<toc parent=requirements sortkey="A" label="Hardware Requirements">
-->

<!--
<toc parent=requirements sortkey="B" label="Software Requirements">
-->

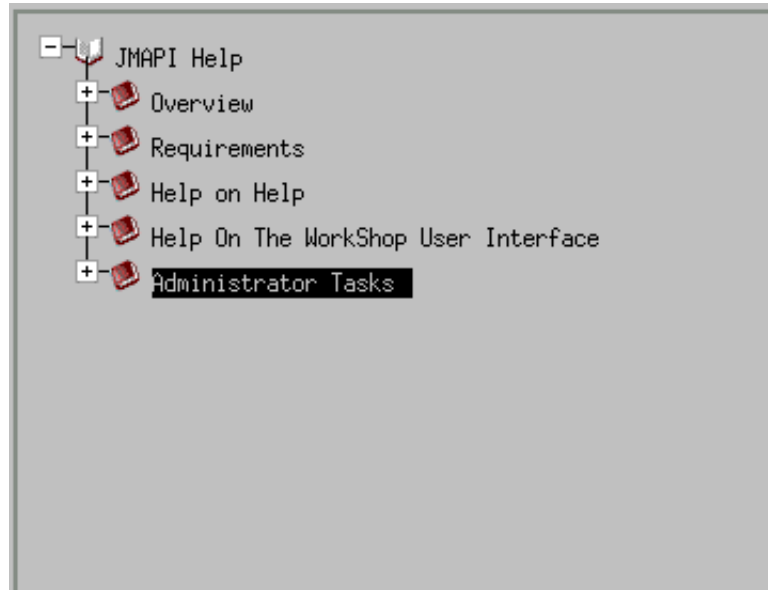
<!--
<toc parent=requirements sortkey="C" label="Authorization
Requirements">
-->

<!--
<toc parent=requirements sortkey="D" label="Network Requirements">
-->

<!--
<toc parent=requirements sortkey="E" label="Supported
Configurations">
-->

.
.
.
```

These tags would produce an expandable / collapsible TOC / Navigator that looks similar to this:



Specifying an Object TOC Entry

Following is the syntax for an object TOC entry:

`<!--`

<code><objecttoc></code>	Used to identify an object's table of contents entry for a given object. Should only exist in an object's HTML file.
<code>label="string"</code>	Label to be displayed in TOC.
<code>name="string"</code>	Name of a TOC entry. Existence indicates entry will have subentries and no HREF definition will be generated.
<code>parent="string"</code>	Parent name of a TOC entry. Entry will be indented under the parent name.
<code>href="URL"</code>	The URL of the hypertext reference for this anchor, where the URL given will be the resource that the browser retrieves when the user clicks the anchor's hotspot. The

URL should be relatively positioned. HREF is ignored if name is present apply to HREF definition. This is ignored if name is present.

sortkey="*string*" Alternate string used in comparing TOC entries. If present, items will be sorted in order of the sortkey instead of the label.

-->

Examples:

To Be Supplied.

Specifying a Related Topics TOC Entry

Following is the syntax for an related topics TOC entry:

<!--

<related> Used to identify an object's TOC entry for a given object. Should only exist in an objects html file.

label="*string*" Label to be displayed in TOC.

name="*string*" Name of a table of contents entry. Existence indicates entry will have subentries and no HREF definition will be generated.

parent="*string*" Parent name of a TOC entry. Entry will be indented under the parent name.

href="*URL*" The URL of the hypertext reference for this anchor, where the URL given will be the resource that the browser retrieves when the user clicks the anchor's hotspot. The URL should be relatively positioned. HREF is ignored if name is present apply to HREF definition. This is ignored if name is present.

sortkey="*string*" Alternate string used in comparing TOC entries. If present, items will be sorted in order of the sortkey instead of the label.

-->

Examples:

To Be Supplied

Specifying an Extensions TOC Entry

Following is the syntax for a topic that extends an existing topic:

```
<!--
```

```
<related>          Used to identify an object's TOC entry for a given object.
                    Should only exist in an objects html file.
```

```
    object="string" Name of the object this extension should be added to.
```

```
    label="string"  Label to be displayed in the TOC.
```

```
    name="string"   Name of a table of contents entry. If present, it indicates
                    entry will have subentries and no HREF definition will be
                    generated.
```

```
    parent="string" Parent name of a TOC entry. Entry will be indented
                    under the parent name.
```

```
    href="URL"      The URL of the hypertext reference for this anchor, where
                    the URL given will be the resource that the browser
                    retrieves when the user clicks the anchor's hotspot. The
                    URL should be relatively positioned. HREF is ignored if
                    name is present apply to HREF definition. This is ignored
                    if name is present.
```

```
    sortkey="string" Alternate string used in comparing TOC entries. If
                    present, items will be sorted in order of the sortkey
                    instead of the label.
```

```
-->
```

Examples:

To Be Supplied

Testing Your Document

TBD

Process Overview

The following illustration show an overview of the help authoring process for JMAPI Help.

- Author help files per Help Content Template
- Define related help files and their links
- Author the help definition page per the Help Template
- Specify the help definition page to the application developer

Required Application Help

JMAPI-based applications are built on top of the JMAPI object and UI framework. Every JMAPI-based application is composed of JMAPI managed objects, property books, and content managers. For the purposes of integrating help from multiple vendors into a single information space, help developers are required to provide certain amount of help for their applications.

Object Help

A managed object is a representation of a system or network entity such as a host, a router, a peripheral device, and so on. Each application that provides a new managed object or an extended managed object (that is, a managed object based on an existing managed object) is obligated to deliver help for that object. Table 3-1 shows the kind of help information that is required.

Table 3-1 Required Object Help Information

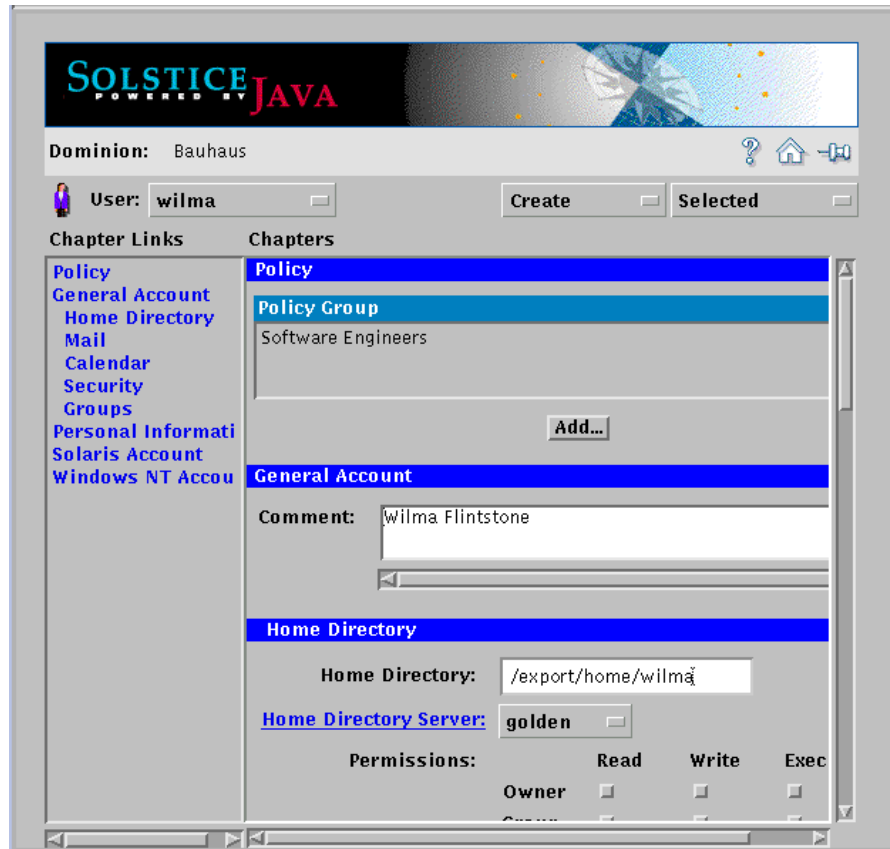
Information Type	Category	Help That Answers the Question
Overview	Purpose	What is the purpose of your object?
“	Audience	Who should use your object?
“	“	What does a user need to know before using your object?
“	How it works	What is your object’s relationship with other objects?
“	“	How does your object modify the system / network?
Key Concepts	Managed objects	What is your object?
“	“	What version is this?
“	“	What does your object do?
“	“	How can a user use your object?
“	“	What are the default attributes of your object?
Key Concepts	Managed object	How can your object be extended?
“	“	What tasks are enabled by your object?

Table 3-1 Required Object Help Information (Continued)

Information Type	Category	Help That Answers the Question
“	“	What services are available via your managed object?
“	“	Who owns your object?
Tasks	N / A	How do you upgrade to a new version of your object?
“		How do you install your object?
“		How do you add /remove your object?
“		How do you modify your object from an object list?
“		How do you add /remove / modify properties for your object?
“		How do you extend your object?
“		How do you add /remove services from your object?
“		How do you work around problems with your object? (For every object error message, there should be a Troubleshooting help file.)

Object Property Book Information

In a JMAPI-based application, every object is accessed and manipulated through a property book. For example, the following figure shows a sample property book for a user object. Generically, this is called a user property book. The object property book provides the means for an end user to manipulate the object, thus modifying the system or network in some way.



Any application that employs a JMAPI object property book is obligated to provide the help in Table 3-2.

Table 3-2 Required Object Property Book Help Information

Information Type	Category	Help That Answers the Question
Key Concepts	Property Book	What can you do from the alpha menu?
“	“	What can you do from the Create menu?
“	“	What can you do from the Selected menu?
Tasks	Configuration	How do you modify your object from within a property book?

Content Manager Help

Any application that employs a JMAPI content manager is obligated to provide the following help files.

Table 3-3 Required Content Manager Help Information

Information Type	Category	Help That Answers the Question
Key Concepts	Content Managers	What kind of information does the content manager display?
“	“	What does this content manager mean in relation to other content managers on the page?
“	“	How does a user manipulate items in the content manager (if applicable)?
“	“	What does the information in the content manager represent?

Standard Information Structure

One of the goals of the JMAPI Help system is extensibility of components and the ability to integrate new online help content and structure into existing content and structure. Templates are provided that integrate help topics into each section and subsection of JMAPI help. Currently, the JMAPI Help information structure consists of the following five bins of information.

- **Help on Help** – Help detailing how to use the JMAPI Help system.
- **Overview** – Overview material about ISV products integrated into the JMAPI environment and product requirements.
- **Tasks** – Task-oriented procedural information telling how to perform network and system administration tasks using applications developed in JMAPI.
- **Key Concepts** – Information about the Managed Objects and Services used for configuring and managing systems, and the graphical user interface (GUI) of a JMAPI application.

- **Error Messages** – Error messages and links to troubleshooting tasks in the Tasks section.

Application help files are integrated into these categories by use of the TOC JMAPI authoring tags.

The following sections of this document provide information about each section of the JMAPI online help system.

Help on Help

The Help on Help section provides information about how to use the JMAPI Help system. As the JMAPI Help system is in the prototype phase of development, this section is currently under development. As the JMAPI Help system is designed and developed, the content explaining the help system will be determined and documented.

The Help on Help section provides an overview subsection which describes the features and benefits of the JMAPI help system. The following outline demonstrates a possible organization of information within the Help on Help substructure.

I. Help on Help (represented as a book icon in the Table of Contents)

A. Overview of the JMAPI Help System (represented as a book icon)

1. Purpose of JMAPI Help (represented as a clickable page)
2. Audience of JMAPI Help (represented as clickable page)

B. Using JMAPI Help (represented as a book icon)

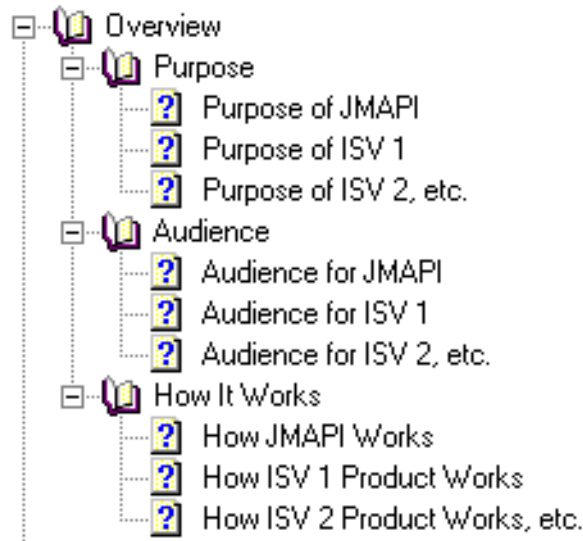
Provides information on how to use the JMAPI Help browser and other help features.

C. How it works.

Overview

The Overview section provides information about the purpose, audience, and intended functionality of JMAPI based products. The following illustration shows the structure of the Overview section as it may display in the hierarchical Table of Contents provided in JMAPI.

Figure 3-1 Help Overview Hierarchy



II. Overview

A. Purpose

Answers the questions: What is the primary use of this tool? How does this tool improve my current system? How, and for what, is this tool intended to be used?

1. Overview and Purpose of ISV Application 1 (for example, High Availability)
2. Overview and Purpose of ISV Application 2 (for example, Software Distribution)

B. Audience

Answers the questions: Who is the intended audience for this product? What are prerequisite skills and knowledge that Audience members need to know before using this product? References available as links (i.e., links to tutorials) for Audience members to update their skill sets before using this product. Also includes links to any getting-started procedures for the product.

1. Who should use ISV application 1 (e.g., High Availability)
2. Before you use ISV application 1
3. Who should use ISV application 2 (e.g., Software Distribution)
4. Before you use ISV application 2

C. How it works

Answers the questions: How is this product supposed to function? How is the functionality of this product enabled? (i.e., by which product features) Also includes links to key concepts, help on pages, and administrator tasks.

1. ISV application 1's relationship to other objects, the system / network, the DB, et al.
2. How does ISV application 1 modify the system / network ISV application 2's relationship to other objects, the system / network, the DB, et al.
3. How does ISV application 2 modify the system / network

Requirements

The Requirements section provides information about hardware, software, network, and protocol requirements of applications developed in JMAPI, as well as supported platforms and configurations on which JMAPI and the integrated ISV applications run. The following outline demonstrates the structure of the Requirements section as it may display in the hierarchical Table of Contents provided in JMAPI.

II. Requirements (a book icon under the Overview book icon in the TOC)

A. Hardware Requirements (represented as a book icon)

1. ISV1 Hardware Requirements (represented as a clickable page icon)
2. ISV2 Hardware Requirements (represented as a clickable page)
3. etc.

B. Software Requirements (represented as a book icon)

1. ISV1 Software Requirements (represented as a page icon)
2. ISV2 Software Requirements (represented as a page icon)
3. etc.

C. Supported Configurations (represented as a book icon)

1. ISV1 Configurations (represented as a page)
2. ISV2 Configurations (represented as a page)
3. etc.

Each subheading of the Requirements section--Hardware, Software, etc.--breaks down into topics for each integrated ISV product.

At the topic level, each topic under these sections will include detailed requirements pertaining to that section. For example, in the Hardware subsection, the topic, ISV 1 (for example, High Availability) Hardware Requirements will include detailed information about minimal and suggested optimal hardware requirements for that product.

Tasks

The Tasks section provides task-oriented procedures for using JMAPI-based system and network applications. For the first release of JMAPI, the tasks will not be organized under sub-headings. In future releases, subheadings may be provided to categorize tasks. Whether categorized or uncategorized, the same content will be provided in the Tasks section, including the following:

III. Tasks

A. Installation

Answers questions about how to install the product.

How to install object A

B. Upgrading

Answers questions about how to upgrade to a new version of the product

How to upgrade to new version of object A

C. Configuration

Answers general questions about manipulating the managed object.

How to add/remove an object A object

How to modify object A from an object list

How to add/remove/modify object A properties

How to extend object A

How to add/remove services from object A

D. Troubleshooting

Answers questions involving how to work around problems indicated by application error messages

This organization is subject to change or expansion based on feedback from future usability tests and customer inputs.

At the topic level, each procedure will be displayed with the following information structure:

- Heading with gerund (for example, “Installing PC-CacheFS”)

- Brief conceptual information related to procedure or introductory text starting with an infinitive (for example, “To install PC-CacheFS”)
- Steps, in sequential order (for example, “1. Insert the PC-CacheFS Start-Up diskette into the external disk drive”)
- Result of the procedure.

Key Concepts

The Key Concepts section provides information about fundamental JMAPI or application-specific concepts, including the concepts of objects and services. The Key Concepts section is another fully extensible piece of the JMAPI Help system into which ISV online help developers can integrate new help content. The following outline demonstrates the structure of the Key Concepts section as it may display in the hierarchical Table of Contents provided in JMAPI.

IV. Key Concepts (represented as a book icon in the TOC)

A. Overview of JMAPI Infrastructure (represented as a page icon)

This topic describes the basic JMAPI concepts of managed objects and services and explains how they work together. The subsection, Managed Objects, provides topics which describe individual managed objects, including objects that are part of ISV applications, and their related services.

Answers generic questions about managed objects and services in the framework of a JMAPI application.

B. Overview of Managed Objects and Services

C. What is a Managed Object, What are Services

D. Types of Managed Objects

1. Software Distribution Object
2. Disk Suite Object, and so on.

Topics in this section answer questions about definition, characteristics, and behaviors of a given managed object, including the following:

What does object A do?
How can I use object A?
What are default attributes of object A?
What extensions are available to object A?
What tasks are enabled by object A (from object menus)?
What services are available to object A?

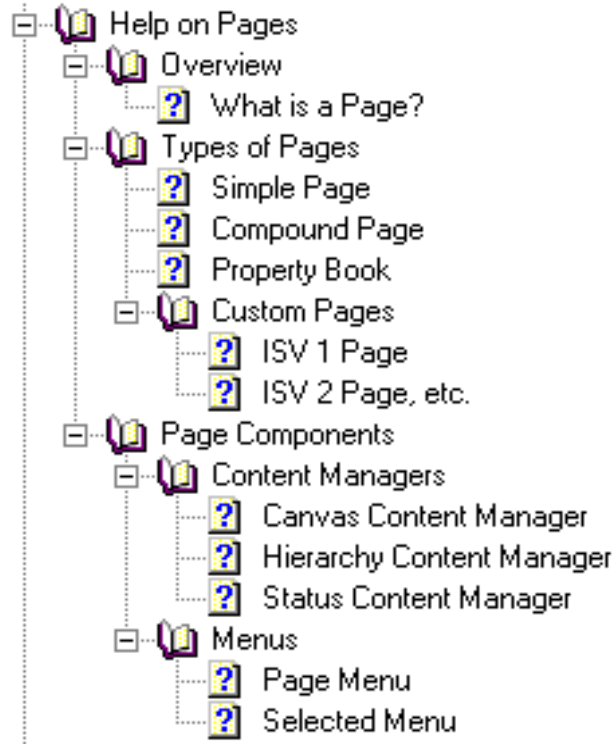
At the topic level, topics about managed objects will contain links to topics about services with which the managed object works. Also, topics about services will contain links to managed objects with which the service works. Both types of topics will have links to corresponding topics in the Tasks section.

Help on Pages

The Help on Pages section provides information about the components of the JMAPI UI. It defines elements of the UI including pages, content managers, and menus, and describes the different types of pages provided with, and used by, JMAPI. Help on Pages is intended to be a fully extensible component of the online help system into which ISV online help developers can integrate new online help content. The following illustration shows the structure of the Help on Pages section as it may display in the hierarchical Table of Contents provided in JMAPI.

Figure 3-2 shows the initial breakdown of the Help on Pages section. ISV developers may include online help for their customized pages, new menus, new content managers, buttons, and any other components they develop. (For example, under the book Custom Pages, the ISV 1 Page could be a new kind of page developed by the Disk Suite team).

Figure 3-2 Help on Pages Hierarchy



Additional subsections beyond Menus include: Buttons, Icons, and Status Line, and can be organized in the following manner:

Buttons (represented as a book icon in the TOC)

1. OK (represented as a clickable page icon)
2. Cancel “
3. Help “
4. Reset “

Icons (represented as a book icon in the TOC)

1. Help
2. Home
3. Sticky Page Push Pin

Status Line

1. A Description of Status Line Messages

This topic will describe what status line messages are and when they are used. Items in the three sections--Buttons, Icons, Status Line--and possibly other sections may also be mentioned in, linked to, or linked from, topics in the Help on Help section.

Error Messages

The Error Messages section provides a list of the error messages of JMAPI and ISV applications developed in JMAPI. The specific organization of the error messages is to be determined. For example, whether error messages will be organized by type of error message, according to the related page of the UI, or according to the owning application (JMAPI or ISV), has not yet been devised. The goal of the error message section is the seamless integration and display of ISV error messages with JMAPI error messages. Templates will be provided for developing error message topics. XX shows the structure of the Error Messages section as it may display in the hierarchical Table of Contents provided in JMAPI. All error message topics will contain links to the corresponding troubleshooting procedures in the Troubleshooting subsection of the Tasks section of JMAPI Help.

Figure To Be Supplied.

Writing Help on Custom Pages and Content Managers

To Be Supplied.

Writing Help on Managed Objects

To Be Supplied.

Linking Guidelines

To Be Supplied.

From Application Pages

To Be Supplied.

Between Help Files

To Be Supplied.

Outside the Help JMAPI Information Space

To Be Supplied.

Help File Naming and Directory Structure

Detail To Be Supplied.

`./java/admin/avm/core/help`

`./sun/admin/pkgname/help`

`./vendor/pkgname/help`

