

*Using Rational SoDA
for Frame
2001A.04.00*

**Copyright © 1994-2001 Rational Software Corporation.
All rights reserved.**

Part Number: 800-024999-000

SoDA Release 2001, revised May 2001

This document is subject to change without notice.

Note the Reader's Comments form at the end of this book, which requests your evaluation to assist Rational in preparing future documentation.

RESTRICTED RIGHTS LEGEND:

Use, duplication, or disclosure by the U.S. Government is subject to restrictions set forth in subparagraphs (a) through (d) of the Commercial Computer Restricted Rights clause at FAR 52.227-19 when applicable, or in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013, and in similar clauses in the NASA FAR Supplement.

Trademark acknowledgments:

The word "Rational" and Rational's products are trademarks of Rational Software Corporation. References to other companies and their products use trademarks owned by the respective companies and are for reference purposes only.

Rational Software Corporation
18880 Homestead Road, Cupertino, CA 95014



Contents

Preface ix

How to Use This Guide ix

Prerequisites ix

Conventions x

Chapter 1 Getting Started 11

Understanding What SoDA Does 11

Information Retrieval 11

Document Generation 12

Customizing Templates 12

Starting SoDA 13

Starting SoDA 13

Customizing Your Startup Files (Optional) 14

Using SoDA with Rational Apex 15

Teaching Yourself the Basics 16

Learning FrameMaker+SGML 16

Accessing SoDA's Online Help 16

Using SoDA's Online Tutorial 17

Selecting a Template 17

Book Templates 18

Copying the Template 18

Browsing Through a Template 19

Connectors 19

Fields 19

	Replicators	19
	Stipulators	20
	Setting Up Source Locations	20
Chapter 2	Generating and Editing Documents	23
	Generating a Document or Book	23
	What Happens When You Generate a Document	24
	Resolving Connectors	24
	Resolving Replicators	24
	Resolving Fields	24
	Resolving Stipulators	24
	Generating a Document	25
	Generating a Report	25
	Generating a Book	26
	Printing a SoDA Book	26
	Browsing Through a Generated Document	27
	Visit Source	28
	Adding to a Generated Document	28
	Adding Descriptive Text or Graphics	28
	Creating Cross References	29
	Making Changes to Generated Text and Graphics	30
	Maintaining a Generated Document	30
	Modifying the Link to a Source Object	30
	Changing the Path of Many Source Objects	31
	Checking Document Consistency	31
	Regenerating a Document or Book	32
Chapter 3	Standard Domains and Templates	33
	The Domain Model	33
	The Rose Domain	34
	Templates for the Rose Domain	34
	The Apex Domain	34
	In-Process Documentation for the Apex Domain	35

	Templates for the Apex Domain	36
	The TestMate Domain	36
	The Summit Domain	36
	The File System Domain	36
	File Records	37
	The Frame Domain	38
	The ClearCase Domain	38
	The ClearQuest Domain	38
	The Rose RealTime Domain	39
Chapter 4	Creating and Customizing Templates	41
	The Template Customization Process	41
	An Example in the File System Domain	42
	Creating Document Templates	42
	Writing a Skeletal Outline	43
	Adding SoDA Elements	44
	Creating Book Templates	44
	Characteristics of Book Templates	44
	Converting a Document Template into a Book Template	45
	Creating the cover document	45
	Creating the book file	46
	Adjusting the book file's structure	46
	Creating Connectors	48
	Creating Replicators	50
	Standard Replicators	50
	Creating Hypertext Link Anchors	53
	Replicators Within Tables	54
	Setting Up a SoDA Table	54
	Adding a Row Replicator	54
	Adding a Replicator within a Cell	55
	Document Replicators	55
	Creating the first document to be replicated	55

Creating the second document to be replicated	56
Creating Fields	57
Fields as Hypertext Link References	59
Creating Stipulators	59
Editing Generated Documents	62
Displaying Hidden Text	62
Editing SoDA Elements	63
Editing a Single, Specific SoDA Element	63
If you already use FrameMaker or FrameMaker+SGML	65
API Clients	65
Templates	66
Using the Online Tutorial	67
Preparation for Each Exercise	68
Element Catalog and Structure View	68
Text Symbols and Element Boundaries	68
Exercise 1-1: Inserting Connectors	68
Instructions	69
Exercise 1-2: Inserting Fields	72
Instructions	73
Exercise 1-3: Inserting a Replicator	78
Instructions	80
Exercise 2-1: Generating a Document	85
Instructions	87
Exercise 3-1: Regenerating a Document	90
Instructions	91
Regenerating with the Delete Mode Set to Remove	91
Regenerating with the Delete Mode Set to Hide	94
Editing a Link Before Regenerating	96
Exercise 4-1: Replicating Table Rows	97
Instructions	99
Exercise 4-2: More SoDA Elements in Table Cells	102
Instructions	103

Exercise 5-1: Generating a Book	107
Instructions	108
Index	111



Preface

SoDA, Rational's **Software Document Automation** product, helps you automate the process of documenting your software, and this guide, *Using SoDA 2001*, contains the information you need to use SoDA once your system administrator has completed the installation.

How to Use This Guide

All SoDA users need to familiarize themselves with the basics of document generation, and therefore should read Chapter 1, "Getting Started" and Chapter 2, "Generating and Editing Documents" In addition, at least one person per project should understand the available templates and how to customize them, as covered in Chapter 3, "Standard Domains and Templates" and Chapter 4, "Creating and Customizing Templates"

The setup instructions will work with the C, Bourne, and Korn shells (**csh**, **sh**, and **ksh**). Most steps are the same regardless of which shell you use. Steps that are shell-dependent provide specific instructions for each shell. Follow the specific instructions for whichever shell you use.

Prerequisites

To follow the instructions in this guide, you need to:

- Know how to log into your workstation
- Have a general understanding of your:
 - UNIX operating system
 - UNIX shell

- X Window System environment
- Be familiar with FrameMaker+SGML (see FrameMaker+SGML's *Getting Started* and *Using FrameMaker+SGML*)

Conventions

This guide uses the following text conventions:

Example	Meaning
<i>soda_release</i>	Indicates a variable element for which you must supply a value. For example, you might enter /someplace/rational/soda in place of <i>soda_release</i> .
soda	Specifies literal text that you enter exactly as shown.
csH	Identifies UNIX utilities, pathnames, or filenames.
File > New	Specifies the menu and the menu button for an operation. It means: From the File menu, choose the New menu button.
Return	Represents a key that must be pressed to initiate or complete an action.
Control-C	Represents keys that must be pressed simultaneously. For example, while holding down Control , press C .
%	Represents your shell prompt when you are a regular user (in the C shell). (Do not type the % as part of the command.)
\$	Represents your shell prompt when you are a regular user (in the Bourne or Korn shell). (Do not type the \$ as part of the command.)
#	Represents your shell prompt when you are the super user (root). (Do not type the # as part of the command.)



Chapter 1

Getting Started

The purpose of this chapter is to guide you through the steps necessary to begin generating and editing documents using SoDA. It includes instructions for starting SoDA and accessing its online help and tutorial.

This chapter includes the following sections:

- “Understanding What SoDA Does” on page 11
- “Starting SoDA” on page 13
- “Teaching Yourself the Basics” on page 16
- “Selecting a Template” on page 17
- “Browsing Through a Template” on page 19
- “Setting Up Source Locations” on page 20

Understanding What SoDA Does

It is critical that you gain a basic understanding of the features of SoDA, even before you start the program for the first time.

The primary function of SoDA is to *retrieve information* from various sources and use it to *generate a document* according to a *customizable template*. Let’s look in more detail at each part of that function.

Information Retrieval

SoDA has the ability to extract information from various sources, known as domains. SoDA includes domains for most of the Rational products, such as Rose and Apex. Embedded in each domain is the knowledge of how information is modeled by that source, in terms of classes, attributes and relationships.

For instance, the Rose domain understands that a class category has attributes of name and documentation, and contains relationships to classes, and has import relationships to other categories. This domain-specific knowledge enables you to retrieve exactly the information you want to document.

One of the benefits of SoDA is that it can support multiple domains, even within the same document. This means that project team members need to use (and learn) just one documentation tool throughout the software life cycle, rather than using a different tool unique to design, coding, or testing.

Document Generation

There are several products that generate documents in an automated fashion. What does make SoDA unique is how well it can *regenerate* a document. SoDA maintains consistency between the document and its source(s) through a process called Intelligent Document Merging. If you delete an object in the source domain, such as a class in Rose, SoDA will remove the section(s) of the document generated from that object. If you add to the source, SoDA will create a new section of the document in the proper location.

And consistency is only half the story of Intelligent Document Merging. The other half is how SoDA handles information that is *not* extracted from the source domain. SoDA lets you add descriptive text, including special formatting such as bulleted lists, equations, and even drawings. When a document is regenerated, SoDA updates the information taken from the domain(s) without affecting the user-added portions.

Another key feature of SoDA is its ability to regenerate a portion of a document. If you know you made a change to just the source for one paragraph in the document, you can select that paragraph and update it, which is much faster than regenerating your entire document every time.

Customizing Templates

Although SoDA comes with several templates, every project has different preferences for documentation. SoDA templates are completely customizable; you can add, change, or delete

portions at will. You can even start from scratch and create your own template.

The customization process itself is completely interactive. You don't need to be a programmer or learn a new macro language to make template changes. You use the publishing tool's WYSIWYG interface to specify structure and style, and you use dialog boxes to specify the SoDA-specific elements.

With this brief introduction, you are now ready to begin learning the SoDA product itself.

Starting SoDA

SoDA uses a publishing tool called FrameMaker+SGML. FrameMaker+SGML provides the WYSIWYG editing of paragraphs, tables, and characters. When you start SoDA, you are starting FrameMaker+SGML as well.

If you already use FrameMaker or FrameMaker+SGML, be sure to follow the instructions in "Special Setup for Prior Frame Users" on page 65, first.

SoDA runs in your X Window System, so your X server and window manager must be running. The examples herein are based on the Motif Window Manager (**mwm**).

Starting SoDA

To start SoDA, follow these steps at a shell prompt inside an **xterm** window:

1. If you want to run SoDA on a remote workstation but have it display on your local workstation:
 - a. In the C shell (**cs**h), enter this command on the remote workstation:

```
% setenv DISPLAY local_host:0
```

Or in the Bourne shell (**sh**) or Korn shell (**ks**h), enter this command:

```
$ DISPLAY=local_host:0; export DISPLAY
```

Replace *local_host* with the unique network name of your local workstation.

- b. To give the remote workstation access to the X server on your local workstation, enter the following command on your local workstation, where *remote_host* is the unique network name of the remote workstation:

```
% xhost +remote_host
```

2. Start SoDA:

```
% $SODA_HOME/bin/soda
```

If you have not defined the **SODA_HOME** environment variable, replace **\$SODA_HOME** with the full pathname to your SoDA release—for example: **/someplace/rational/soda**.

The **soda** script sets and exports several environment variables, adjusts your **PATH** variable, and starts FrameMaker+SGML, which in turn starts the SoDA API client. The main FrameMaker+SGML window appears, and SoDA is ready for your use:



Customizing Your Startup Files (Optional)

For added convenience, you can add some of the above commands to the startup files in your home directory. For example:

1. If you use the C shell (**cs**) as your login shell, edit your **.login** file.
 - a. Define the **SODA_HOME** variable:

```
setenv SODA_HOME full_path_to_SoDA_release
```
 - b. Append **\$SODA_HOME/bin** to your **PATH** variable

- c. If you regularly run SoDA on a remote workstation but have it display on your local workstation, define the **DISPLAY** variable:

```
setenv DISPLAY local_host:0
```
2. If you use the Bourne shell (**sh**) or Korn shell (**ksh**) as your login shell, edit your **.profile** file.
 - a. Define the **SODA_HOME** variable:

```
SODA_HOME=full_path_to_SoDA_release  
export SODA_HOME
```
 - b. Append **\$SODA_HOME/bin** to your **PATH** variable
 - c. If you regularly run SoDA on a remote workstation but have it display on your local workstation, define the **DISPLAY** variable:

```
DISPLAY=local_host:0  
export DISPLAY
```
3. If you regularly run SoDA on a remote workstation but have it display on your local workstation, edit your **.xinitrc** file to add this line:

```
xhost +remote_host
```
4. Log out and log back in again so the changes will take effect.

Now you can start SoDA simply by entering:

```
% soda
```

Using SoDA with Rational Apex

You can use SoDA with or without Rational Apex. To use SoDA with Apex, Apex must also be running (before you start SoDA).

To start SoDA with Apex¹, enter this command at a shell prompt inside an Apex shell window:

```
% soda
```

As described previously, the main FrameMaker+SGML window appears, and SoDA is ready for your use.

¹It is also possible that, during your SoDA installation, **apexinit** was adjusted to always start SoDA automatically.

Teaching Yourself the Basics

In addition to this guide, there are three sources that you can use to teach yourself about SoDA:

- The *Using FrameMaker+SGML* manual
- The Online Help
- The Online Tutorial

Learning FrameMaker+SGML

Before you can use SoDA effectively, you should familiarize yourself with basic FrameMaker+SGML concepts. Consider studying the following chapters in the *Adobe FrameMaker+SGML User Guide*:

- *Introduction*
- *Quick Tour*, Chapter 1
- *Documents*, Chapter 2
- *Work Area*, Chapter 3
- *Elements*, Chapter 4
- *Text*, Chapter 5

If you expect to be customizing SoDA templates, the following chapters may help as well:

- *Tables*, Chapter 6
- *Cross-References, Variables and Footnotes*, Chapter 7
- *Tables of Contents and Indexes*, Chapter 9
- *Books*, Chapter 10

Accessing SoDA's Online Help

To access SoDA's online help:

- Main Menu
 1. Choose **Help > SoDA Help** from a SoDA document window.
The SoDA main help menu appears.
 2. Click on one of the SoDA topics listed.
- API Applications menu

1. Choose **Help > Context-Sensitive Help** from a SoDA document window. (You can also press the F1 key on most keyboards.)

Your cursor becomes a question mark.

2. Using the question-mark cursor, choose an item from the **SoDA** or **Control** menu.

The FrameMaker+SGML help menu appears.

3. Click on **SoDA Help** and choose a topic to see its online help.

Navigate through SoDA's online help as you would through FrameMaker+SGML's help. Choose **FrameMaker Help** from the main FrameMaker+SGML help window for more information on using help.

Using SoDA's Online Tutorial

If you are new to SoDA, its online tutorial can introduce you to the key concepts and procedures you need to know before you start working on your own documents.

To use the tutorial, you can either:

- In the online help main menu, click on **SoDA Help**, and then click on **Tutorial** under the **Online Manuals** heading.
- Choose **SoDA Tutorial** from the **Help** menu on any SoDA document window.

A tutorial document will open; follow its instructions.

Also note that—to avoid taking up too much space on your workstation screen—the instructions for some of the tutorial's exercises are delivered in Appendix B of this document.

Selecting a Template

Once you understand how to use FrameMaker+SGML and have it running, you are ready to start using a SoDA template.

The standard SoDA templates are stored in a series of directories in `$SODA_HOME/share/templates`. See the **SoDA Templates** section in the SoDA help menu or see Chapter 3, "Standard

Domains and Templates,” to help you choose the template that meets your requirements.

Document templates may be opened by selecting **NEW** in the FrameMaker+SGML main window, then navigating to the SoDA templates and choosing one. Book templates must be copied to their new location, as described below.

Book Templates

Most templates used in SoDA are not single documents, but books of documents, or *book templates*.

Sometimes a SoDA document becomes unwieldy after generation because so much information is extracted from the sources and inserted into the document. Some documents are better arranged as a book with chapters than as one large document. Also, it is common for more than one person to have to work on a large document.

For all of these reasons, you may be using a book template that is a collection of document templates.

Copying the Template

To copy a template for your personal use, follow these steps from the shell prompt in an X terminal window:

1. Create a directory that will contain the document (or book). If you are using Rational Apex, consider using a working view.
2. From this working directory, copy all the files from the template directory. For example, to copy the Rose-based in process template, use this command:

```
% cp $SODA_HOME/templates/RoseDomain/InProgress/* .
```
3. Use **chmod** if necessary to give yourself write permission on the files.
4. Change the name of your document or book if desired.

Browsing Through a Template

Once you have your own copy of a template use the **OPEN** command in the main FrameMaker+SGML window to open the document (or one of the documents in the book). Display the structure view, and observe how the structure is used to help create consistent documents by forcing you to follow the structure rules.

Each SoDA template contains one or more of these SoDA elements: connectors, fields, replicators, and stipulators. The following sections include a brief description of each element.

Connectors

Connectors identify a particular object in a source domain. They normally provide the highest abstraction of a domain object from which other SoDA elements can be defined. For instance, connector might identify a file in a directory, a Rose model, or an Apex subsystem.

Fields

Fields are attributes of a source object that are inserted into a generated document. A field can be either text or a graphic. Any of the following might be fields:

- The name of an Rose class category
- The cardinality of a “has” relationship
- An encapsulated PostScript file
- The name of a directory
- The contents of a text file

Replicators

Replicators identify structures within a document that are replicated, or repeated, for each object found based on some relationship.

Replicators are useful when there is a set of objects in the source domain that need to be uniformly documented. The document template defines the structure, format, and generic

content of the section. SoDA builds a section for each object found in the source. SoDA maintains the consistency between the document and the objects in the source.

Stipulators

Stipulators are expressions that evaluate to True or False. If False, the structure to which the stipulator is attached is left out of the document.

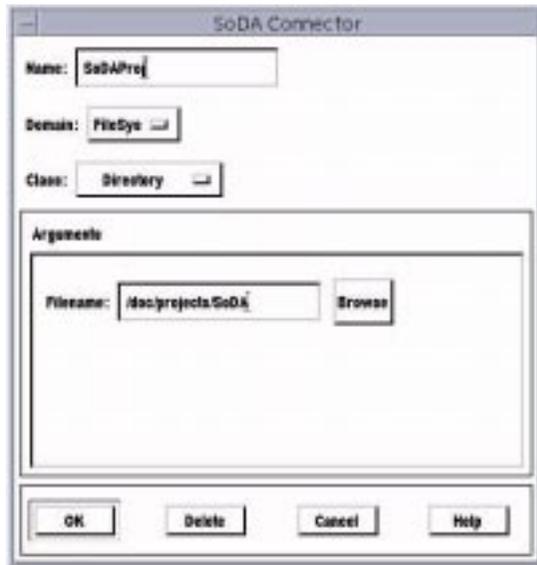
Setting Up Source Locations

Your document template uses connectors to specify the location of your source(s). Before you generate the document for the first time, check the connectors to make sure they are pointing to the proper locations.

To verify or change the source locations, open your document (or the first document in your book) and display the structure view. The connectors are usually among the first elements in the document:



Select the element, and choose **SoDA > Connector**. SoDA will display a dialog box similar to this one:



Examine the Arguments field. If the specified path name is not correct, change it. Do not change the other fields in the dialog box.

Be sure to save these changes using **File > Save**. You are now ready to generate your document.



Chapter 2

Generating and Editing Documents

This chapter will take you through using an existing document template to generate and edit a SoDA document.

The chapter assumes that a template exists for you to work with. If you do not have a project-specific template, you can use one of the templates supplied with SoDA. If you are responsible for creating or modifying the document templates for your project, you will need to refer to Chapter 4, “Creating and Customizing Templates.”

The following sections are included in this chapter:

- “Generating a Document or Book” on page 23
- “Browsing Through a Generated Document” on page 27
- “Adding to a Generated Document” on page 28
- “Maintaining a Generated Document” on page 30
- “Regenerating a Document or Book” on page 32

Generating a Document or Book

Once you have a template, you use SoDA by following this straightforward process:

1. Generate your document, which will retrieve information from your sources.
2. Edit the generated document, adding descriptive text and graphics that supplement the data available from your sources.
3. Regenerate the document, or just a portion of the document, whenever you make changes to your sources.

What Happens When You Generate a Document

When you generate your document, SoDA searches through the information sources referenced in your document by SoDA elements. SoDA takes some action within the document depending upon the type of element being resolved.

Resolving Connectors

When a connector is resolved, it is used as a pointer to a particular object within the referenced source domain. After generation, connectors remain in the document.

Resolving Replicators

For each object returned by a replicator, a structure is created, and a link replaces the replicator in that structure. (The structure created can be a section, a list item, or any other element that it makes sense to replicate.) The link refers only to the one object that corresponds to the new structure, not to the whole group of objects as the replicator does. Replicators remain in the document, but they are hidden using conditional text.

Resolving Fields

When a field is resolved, the attribute specified in the field is returned and inserted into the document.

Resolving Stipulators

Stipulators result in a new section or nothing.

If the expression specified in the stipulator is True, the section will be created, and the stipulator remains in the document playing a role similar to a link.

If the expression specified in the stipulator is False, the stipulator remains in the document, but it is hidden using conditional text. Whenever the document is regenerated, the stipulators are checked again to see if the expression has changed to True.

Generating a Document

After setting up your SoDA document template with the correct source locations, you are ready to generate your document.

Follow these steps:

1. Position your cursor anywhere within your document and choose **SoDA > Generate Document**.
The Generate dialog box will appear.
2. Choose generation options (detailed in the online help) and click **OK** to generate your document.

As document generation begins, SoDA's progress indicator will appear.

When generation is complete, you are informed with a SoDA information dialog. If generation completed successfully, this message appears: "Document generation completed with no errors." If there were errors during generation, an error report will appear.

The error report is a hypertext document describing the sections where errors occurred. Clicking on a particular error description will take you to the section in your document being described. After traversing to the incorrect portion of your document, you can change the SoDA elements to make your document correct and regenerate your document.

An example error report is shown below:

```
ERRORS ENCOUNTERED GENERATING: /kirk/andrea/example.mstr
*** Unable to evaluate connector "openFailed:
    Can't find directory object, name = /doc/projects/Rose"
*** Error evaluating text field "contextNotFound: SoDAproj"
*** Unable to evaluate replicator "contextNotFound: SoDAproj"
```

Generating a Report

Intelligent Document Merging has a cost, and that cost is performance. If you don't need to perform incremental regeneration, you will be better off generating reports instead of templates. Reports can only be generated from a SoDA template, that is, the document must not have been previously generated. Since SoDA does not have to keep track of what information was entered manually, it can generate the report

much faster than a document. To generate a report, choose the **SoDA > Generate Report** command.

Generating a Book

To generate a SoDA book:

1. Choose **SoDA > Generate Book** from the book file's menu bar. The Generator dialog box will appear.
2. Choose generation options and click **OK** to generate your book.
 - ❑ To generate a book suitable for printing, choose **Produce Printable Book**. (See “Printing a SoDA Book” on page 26 for more information.)
 - ❑ To update the contents of documents within the book, choose **Update Documents From Sources**. This option opens each document in the book and generates the document.
 - ❑ To update a book's documents—but not the documents' contents—choose **Update Book From Sources**. This option creates (from a document to be replicated) a document for every object found in the source. The contents of those documents are *not* generated.
 - ❑ To generate the entire book including the contents of the book's documents, choose both **Update Documents From Sources** and **Update Book from Sources**.

When a book is generated, any documents that are not currently open on the desktop are opened, updated, saved, and closed as necessary.

Printing a SoDA Book

Some of the information in SoDA books is not appropriate for printing. Each document replicator—documents with percent signs (%) in their filenames—contains internal SoDA information. If SoDA did not “hide” replicators after generation, those sections would print whenever you print your generated document.

It is not possible to hide document elements within a book. The only way to remove the document replicators is to “delete” them from the book file (remove the reference to them from the book

file). If you delete those documents, however, your book will no longer generate correctly.

SoDA gives you the option of generating a printable book. This printable book is separate from your SoDA book template and your generated book. It has no references to the document replicators. Only the resulting generated documents remain.

Caution: *Any changes to the structure or contents of the book must be made in the original—not in the printable book. No changes are retained in the printable book.*

To generate a printable book:

1. Choose **SoDA > Generate Book** from the book file's menu bar. The Generator dialog box will appear.
2. Select **Produce Printable Book** and click **OK** to generate your document.

When SoDA completes the printable book, both book files will be open.

The printable book is just a normal, FrameMaker+SGML book. It has all the same files as the original book, except the document replicators have been removed (the documents with % in their filenames).

The printable book will be named with a **.prn** extension. For example, if the generated book is named **testBook.book**, the printable book will be named **testBook.book.prn**.

Browsing Through a Generated Document

When you have completed document generation, your SoDA document will include replicated sections, tables, and field values according to the information in the source and the definition of the template.

As you browse through a generated document, you will notice several differences from the original template:

- Field placeholders have been replaced by actual field values
- In the structure view, SodaReplicator elements no longer appear

- Sections generated by replicators have attached *links*, or pointers back to their sources.

It is possible that after you generate a document the first time you notice changes you wish to make to the template itself. In this case, use the **File > Revert to Saved** command to restore the original template. Then follow the procedures described in Chapter 4 to modify the template.

Note: *It is much easier to change a document template before supplemental text and graphics are added to the document than after.*

Visit Source

As you browse through a generated document, you may wish to traverse to the source that corresponds to a particular section. Perhaps you discovered a misspelling or other error. Most domains support hypertext-like links to the sources. To use this feature, select a SoDA field, or select the section in the structure view that includes the link (the **SodaPrimaryLink** element), and choose the **SoDA > Visit Source** command.

To improve performance, SoDA reports do not contain a full set of links. For this reason, the visit source command may not be available for all sections in a generated report.

Adding to a Generated Document

Once you have generated a document for the first time, you will notice areas where you can add supplemental information. This section covers the process of adding to a SoDA document, including cross-references.

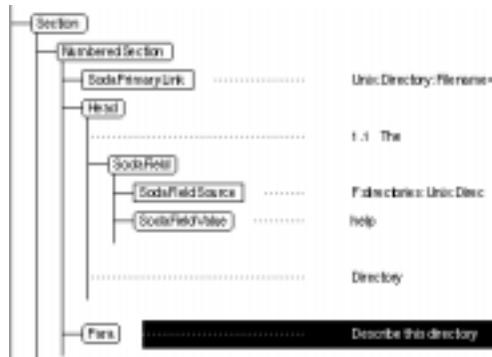
Adding Descriptive Text or Graphics

Most document templates include text placeholders for information that is not stored in the source domain. The document is now ready for these placeholders to be replaced with real descriptions.

To add descriptive text:

1. Select a text placeholder in the document window by dragging your cursor through the text. Ensure that only the text is selected—not the entire element.

When the text is selected, the structure view of a paragraph will look something like this:



2. Begin typing the descriptive text.
The text placeholder is replaced by the descriptive text you type.
3. Replace all the text placeholders with real descriptive text.

Many templates include prompts to help you identify placeholders. You can traverse from prompt to prompt using the **Edit > Find/Change** command, and searching for variables.

Creating Cross References

A *cross-reference* is text that refers to (and is linked to) an element in another document (or perhaps in the same document). The most common use of cross-references in SoDA templates is for requirements traceability. With this feature you can tie a part of your design to a specific requirement with a few mouse clicks.

If a cross-reference is included in your SoDA template you will see a title that typically includes placeholders for a paragraph number and a title, such as “0.0.1 Capability cross-reference.” To create an actual cross-reference, double-click the place-

holder. If the cross-reference is to another document, SoDA will ask you if you want it opened; click **OK**.

You should now see a list of the headings that make up the possible cross-references. Click on the one you need, then click on **Replace**.

Note: *The cross-references created this way are “live,” meaning as your requirements document changes and the paragraph numbers change, the cross-references will be updated accordingly.*

Making Changes to Generated Text and Graphics

The Rule: Never make changes to generated text!

If you edit the text generated by SoDA, changes will be overwritten when the document is regenerated. If you see an error of any sort in generated text, correct it in the source.

To help you identify generated text, SoDA uses a special character format called **SodaField**. For color monitors, the format will set SoDA fields to blue. If you have a monochrome display, you may want to change the format to underline the fields instead.

Maintaining a Generated Document

After your document is generated, you can change SoDA elements within the document, and you can also change the source objects in the information source domain. When you make source changes, however, you must regenerate the corresponding portions of your document to keep the document and source information consistent.

If you change the names of source objects, and you have added additional information to the generated section, you must change links that refer specifically to those objects. If you move objects, you must change all SoDA elements that refer to those objects.

Modifying the Link to a Source Object

Edit a link when the name of a source object has changed.

To edit a link:

1. Select the entire section containing the link (the element called **SodaPrimaryLink**) and choose **SoDA > Utilities > Edit Link**.

You will see the Edit Link dialog box listing the selected link and its possible sources. The list of sources is the result of re-evaluating the replicator from which the selected link was created.

2. Choose the link you want to change and click **OK**.
The selected link is updated to reflect its new source.

Changing the Path of Many Source Objects

If you have a document that uses UNIX files as sources, and you move those sources to another directory, SoDA would lose the connection between the sources and their generated sections. Thus, the next time you generate the document SoDA would delete all of those sections and you would lose any supplemental information that you added.

SoDA > Utilities > Modify Links allows you to make global pathname changes to your links so that you do not have to change each element individually.

To change the path in many SoDA elements:

1. Position your cursor anywhere in your document, and choose **SoDA > Utilities > Modify Links**.
The Modify Links dialog box will appear.
2. Enter the old path in the **From** field, enter the new path in the **To** field, and click **OK**.

When SoDA completes the task, you will see a report indicating which links were updated successfully, and which failed.

Checking Document Consistency

The **SoDA > Check Document** command lets you determine whether your document is consistent with your source(s). When you choose this command, SoDA goes through all of the steps necessary to regenerate your document, but rather than

making updates, it creates a report listing what updates would be made.

Regenerating a Document or Book

Whenever you make changes to the information in your source domains, you will want to regenerate your document. You can regenerate the entire document, or you can regenerate a selected section.

To regenerate your entire document (or book), follow these steps:

1. If you have made recent changes to your document(s), save your work using **File > Save As**.
2. Choose **SoDA > Generate Document** from the document window or **SoDA > Generate Book** from the book window.

The Generator dialog box will appear.

3. Choose generation options and click **OK** to generate your document.

If you know you only made changes to the source corresponding to a particular section of a document, you can regenerate just that section, and save some time in the process. To do so, first select the section by clicking the proper node in the structure view, then using the **SoDA > Regenerate Section** command.

Note: *If you are not sure which sections of a document have changed, regenerate the entire document. It will be faster than using SoDA > Check Document followed by regenerating the changed sections.*



Chapter 3

Standard Domains and Templates

If you have used Chapters 1 and 2 to generate a document using a standard template, you may have discovered that you wish to modify the template to obtain different information from the domain. In this chapter you will learn how to determine what information can be retrieved from your sources, as well as what templates are available. In Chapter 4, you will use the information you learned here to modify a template or create a new one.

After a brief discussion on the domain model, the standard domains and templates are described in the following sections:

- “The Rose Domain” on page 34
- “The Apex Domain” on page 34
- “The TestMate Domain” on page 36
- “The Summit Domain” on page 36
- “The File System Domain” on page 36
- “The Frame Domain” on page 38
- “The ClearCase Domain” on page 38
- “The ClearQuest Domain” on page 38
- “The Rose RealTime Domain” on page 39

This chapter does not describe the details of each domain, only the basic concepts; see the online help for more information.

The Domain Model

SoDA domains are defined using classes, relationships, and attributes. Each of these terms is described here:

Class: one of several types of objects that are represented in the source domain. One class (known as a *child* class) can *inherit* from another class (known as a *parent* class), which makes all relationships and attributes defined for the parent class available to the child class.

Relationship: information that can be retrieved from a class which results in another class or set of classes. Relationships can be *unary*, where the result is always a single object, or *nary*, where the result can be one or more objects. SoDA replicators are defined using nary relationships.

Attribute: information that can be retrieved from a class which results in either text or graphics. SoDA fields are created using attributes.

The Rose Domain

If you are using SoDA with Rational Rose, the Rose domain allows you to incorporate information from Rational Rose into your SoDA documents.

The Rose domain gives you access to information stored in a Rose model, including details from class diagrams, module diagrams, and object diagrams.

Templates for the Rose Domain

The Rose domain includes a full set of templates for documenting software development which follows the Rational Unified Process. Also included with the Rose domain are generic templates for an architecture document, several design documents, and some data dictionary documents. Two of the design documents include traceability to Use Case documents. Refer to the online help for more information on how to use the Rose domain.

The Apex Domain

If you are using SoDA with Rational Apex, the Apex domain allows you to incorporate information from Rational Apex into your SoDA documents.

The Apex domain supports CMVC-related and Ada-related classes of objects. The CMVC-related classes give you access to subsystem, view and file information. The Ada-related classes let you document the Ada units and the declarations within the units.

In-Process Documentation for the Apex Domain

With in-process documentation, a relationship is created between an Apex view and a template in SoDA. When a developer makes an addition or change to a unit in the view, he or she can choose a command directly from Apex that regenerates the document in real time. When the new document is displayed, the developer can add additional descriptive text, formatted text such as bulleted lists, or even extra figures. For example, you might want to explain alternative solutions to a change while you are thinking about it, but don't want to put that information in the comments.

When SoDA in-process documentation is installed, Apex users will see new commands in the **Tools** menu. When an engineer makes a change to an Ada unit, he or she can choose the **Tools > SoDA > Generate** command in Apex, and SoDA will regenerate the section of the document based on the selected unit.

The rest of the project team can take advantage of in-process documentation as well. If a team member needs to access a unit of another team member, he or she can choose **Tools > SoDA > Visit** to open the document associated with that unit, which will display a read-only copy of the additional information that is not part of the unit's documentation.

To use in-process documentation with Apex/Ada, you must follow these steps:

1. Visit the view that contains the code you wish to document.
2. Create a subdirectory called `doc`.
3. Copy the book and documents from `$SODA_HOME/share/templates/ApexDomain/InProgress` into the `doc` directory.

Once the book and its documents are in place, the commands from the **Tools** menu in Apex will work properly.

Templates for the Apex Domain

In addition to the in-process template, there are also templates that support the DoD-2167A standard and MIL-STD-498. The latter templates are also ISO9000-compliant and can be used to support any software development effort. Refer to the online help for more information on each template.

The TestMate Domain

If you are using SoDA with TestMate, the TestMate domain allows you to incorporate information from test cases, test lists, the test environment, and coverage information.

The Summit Domain

If you are using SoDA with Rational Summit, the Summit domain allows you to incorporate information from task domains, task kinds, tasks, and switches.

The File System Domain

The File System, or FileSys, domain lets you to incorporate information from the file system into your SoDA documents. This information can come from directories, files, or records within files.

All domain objects provide naming information, including full pathnames, simple pathnames, as well as parts of a name such as prefix and extension. The canonical name (its complete pathname from the root of the file system with all symbolic links resolved) is also available.

For directories, you can traverse the contents of the directory as well as parent directory.

For files, you can retrieve the contents of the file, assuming the file contains either ASCII text or a graphic.

File Records

ASCII text files can be further decomposed into file records. File records are especially useful for parsing flat database files.

By default, SoDA uses newlines to delimit records within a file, spaces to delimit fields within a record, and double quotes (") to surround a single field that includes spaces. Records must also contain key fields that uniquely identify each record. By default the first field is the key.

For example:

Single Record	["William (Bill)"	Clinton	Democrat	1993	2000
		George	Bush	Republican	1989	1992
		Ronald	Reagan	Republican	1981	1988
		"James (Jimmy)"	Carter	Democrat	1977	1980
		┌──────────┐	┌────────┐	┌──────────┐	┌──┐	┌──┐
Field01	Field02	Field03	Field04	Field05		

You can add directives at the beginning of the file to change the default delimiters and key fields. For example, you can specify that a semicolon delimits records in a file rather than a newline. Or, that the key field is the third field combined with the fourth field.

For example:

```
#FIELD_DELIMITER /
#KEY_FIELDS 2
William (Bill)/Clinton/Democrat/1993/2000
George/Bush/Republican/1989/1992
Ronald/Reagan/Republican/1981/1988
James (Jimmy)/Carter/Democrat/1977/1980
```

Now, with the directives added to the example:

- Fields are separated by a slash instead of a space.
- The second field (containing the last name) is used as the key.

The Frame Domain

The Frame domain allows you to incorporate information from any structured FrameMaker+SGML (or SoDA) documents into your SoDA documents.

The Frame domain lets you traverse a Frame document and locate specific elements (nodes in the hierarchy of the document's structure) from structured documents, plus markers and cross-references from structured or unstructured (FrameMaker) documents.

The ClearCase Domain

If you use ClearCase to manage your configuration management environment, the SoDA ClearCase domain enables you to incorporate version control and build management information into your SoDA documents.

The ClearCase domain supports retrieval of information pertaining to VOB objects, including object attributes, labels, hyperlinks, version histories, etc. Additionally, non-VOB objects such as view-private objects may be documented through inheritance from the File System domain.

For more information on the ClearCase domain, refer to the online help.

The ClearQuest Domain

If you use Rational ClearQuest, the SoDA ClearQuest domain enables you to incorporate change request/defect tracking information into your SoDA documents.

The ClearQuest domain automatically reads the schema of your ClearQuest database during template customization and document generation. The information is translated into a dynamic domain schema on which a template is based. The properties and relationships available for a ClearQuest template will depend on this dynamic schema. Since the ClearQuest database schema can vary from database to database,

ClearQuest templates written for one database may not generate using another database.

The ClearQuest domain supports retrieval of all information available from a ClearQuest database, including defects, enhancement requests, attachments, history, users, etc. In addition, templates can utilize replicators to iterate over the results of a ClearQuest query.

For more information on the ClearQuest domain, refer to the online help.

The Rose RealTime Domain

The Rose RealTime source domain allows you to incorporate textual and graphical information from Rational Rose RealTime models. To extract information from a Rose RealTime model, you would typically create a connector to the model specifying its filename. Once this command provides context for the model, you can traverse through the various components.

Before generating a document from a Rose RealTime model, you need to save the model. When you generate your SoDA template, the Rose RealTime domain will create a directory named <document name prefix>.dia and will fill it with .JPG files for each diagram requested from the model. On Solaris, by default, the environment variable `SODA_GRAPHIC_FORMAT` is set to `jpg`. If it is set to `tiff`, generation will produce TIFF files rather than JPEG files.



Chapter 4

Creating and Customizing Templates

This chapter guides you through creating your own templates, either by modifying an existing template, or by creating a new template from scratch.

After a brief discussion on how templates work, the following sections describe each of the SoDA template-creation process steps in detail:

- “Creating Document Templates” on page 42
- “Creating Book Templates” on page 44
- “Creating Connectors” on page 48
- “Creating Replicators” on page 50
- “Creating Fields” on page 57
- “Creating Stipulators” on page 59
- “Editing Generated Documents” on page 62

The Template Customization Process

When you need to customize a SoDA template, you follow these steps:

1. Copy either an existing SoDA template, or an empty FrameMaker+SGML template.
2. Insert SoDA elements where needed.
3. Save the template.
4. Generate a test document
5. Confirm that everything generated as expected.
6. Revert to the saved template.
7. Correct existing SoDA elements and insert new ones.

8. Repeat steps 3-7 until complete.

A *SoDA document template* is a FrameMaker+SGML template populated with SoDA elements. A *SoDA book template* is a structured book and its associated SoDA document templates.

A good understanding of your information as it exists within each domain and a detailed document plan will ensure that the placement of SoDA elements within a document or book template will yield the desired results after generation.

An Example in the File System Domain

The examples used in the following sections will use the `/doc/projects/SoDA` directories and files that are shown below:

```
/doc/projects/SoDA/  
  bin/  
  example_doc.doc  
  help/  
    connector.doc  
    field.doc  
    help.book  
    replicator.doc  
    stipulator.doc  
  install_guide/  
  release_note/  
  user_guide/  
    frame_domain.doc  
    key_concepts.doc  
    filesys_domain.doc  
    user_guide.book  
  writer_notes.txt
```

Creating Document Templates

You have two choices when you customize a template: you can modify an existing template, or start with an empty template.

In either case you should have some idea of what you want your document to look like before you start the process.

Writing a Skeletal Outline

You can add SoDA elements as you write the document, but you'll probably save time by designing the document first, and adding SoDA elements later.

For example, a simple document which describes directories and files extracted from the File System domain might have an outline like this:

```

1 DIRECTORIES AND FILES ASSOCIATED
WITH THE PROJECT

1.1 The Directory
Describe this directory

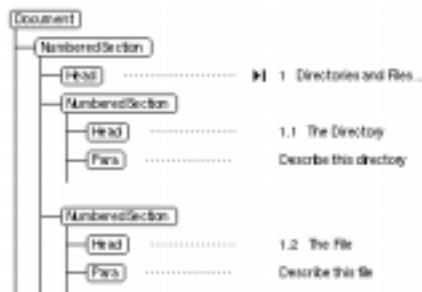
1.2 The File
Describe this file
    
```

This outline looks strange because additional information is needed:

- The project name should be inserted between “THE” and “PROJECT” in paragraph 1.
- The name of a directory should be inserted between “The” and “Directory” in paragraph 1.1.
- The name of a file should be inserted between “The” and “File” in paragraph 1.2.

This is information SoDA can easily generate for you.

The structure view of the example document looks like this:



Adding SoDA Elements

Once you have a document in the format you like, you can add the SoDA elements as described in the rest of this chapter.

Creating Book Templates

Sometimes, a SoDA document becomes unwieldy after generation because so much information is extracted from the sources and inserted into the document. Some documents are better arranged as a book with chapters than as one large document. Also, it is common for more than one person to have to work on a large document.

For all of these reasons, SoDA supports FrameMaker+SGML's book capability. (For details about books, see *Using FrameMaker+SGML*.)

Characteristics of Book Templates

Creating a book template is similar to creating a document template. You must:

- Connect your book and its documents to your sources using connectors.
- Define replicators for document sections and entire documents that you want replicated.
- Define fields to extract attributes from the source domain.

There are some differences, however:

- Any connectors which are to have a global scope within the book must be attached to the root element of the first document in the book.
- All documents (except the first document) in the book which contain references to global connectors must contain a **SodaBookName** element, containing the book's name, attached to the root element.
- In replicated documents, the **SodaReplicator** element must be the first child of the root element. In all other documents, the **SodaBookName** element is the first child element.

- Documents to be replicated must be named with field information delimited by percent signs (%). For example, a document which is replicated for each directory in `/doc/projects/SoDA` would have the name `%SimpleName%.doc`. The text between the percent signs indicates the attribute to be extracted (the simple name of each directory found by the document's replicator), and that attribute will create the unique name of each of the replicated documents. The resulting documents would be `help.doc`, `install_guide.doc`, `release_note.doc`, `user_guide.doc`, and `bin.doc`. You can also specify text "outside" the %. For example, `new_%SimpleName%.doc` would produce the following replicated documents: `new_help.doc`, `new_install_guide.doc`, `new_release_note.doc`, `new_user_guide.doc`, and `new_bin.doc`.

Converting a Document Template into a Book Template

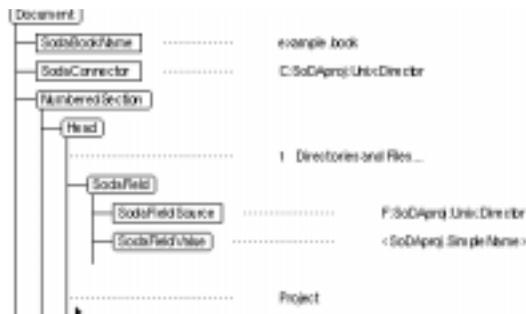
In a new example book derived from the example document, the first document—a "cover" document—contains the global connectors, the second document will be replicated for each directory in the source domain, and the third document will be replicated for each file in the source domain.

The following steps are used to convert the example document to a book with associated documents.

Creating the cover document

1. Save the document as `cover.doc`.
2. Delete the **Section** elements and all their children from the document. Leave the first **NumberedSection** element.
3. Add a **SodaBookName** element to `cover.doc`, and identify the book as `example.book`. To do so, position your cursor directly under the **Document** element and choose **Book Name** from the **SoDA** menu. Enter `example.book` and click OK.

The resulting document, **cover.doc**, looks like this:



Creating the book file

1. Generate **example.book** from **cover.doc**. (See *Using FrameMaker+SGML* for details.)

Adjusting the book file's structure

1. Generate/Update the file. (See *Using FrameMaker+SGML* for details.)

The book's structure view now looks like this:



2. Select the book's root element (now called **NoName**), and change it to a **Book** element. (See *Using FrameMaker+SGML* for details about changing elements.)

The **Book** element in the example is simply a container element that is valid as the highest-level element. It is also defined to contain **BookSection** and **Document** elements.

The book's structure view now looks like this:



3. Select the second **Document** element (the element representing %SimpleName%Directories.doc), and wrap it in a **BookSection** element.

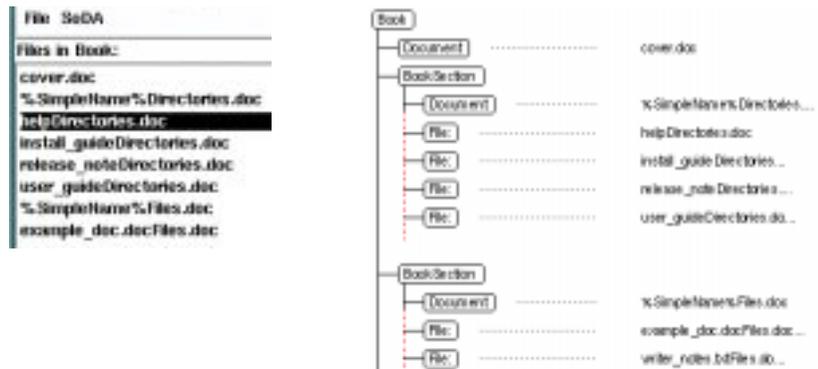
As discussed in “Creating Replicators” on page 50, sections to be replicated may not be siblings. Documents to be replicated may not be siblings, either.

4. Select the third **Document** element (the element representing %SimpleName%Files.doc), and wrap it in a **BookSection** element.

The book is now a valid SoDA book template. Its structure view looks like this:



The generated book looks like this (book window on the left, book’s structure view on the right):



Creating Connectors

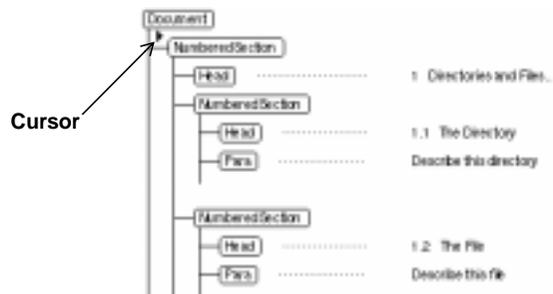
Connectors are generally defined to point to the top-most pieces of *your* information within a source domain. For example, in the document above, the top-most piece of information is the name of the project. Using the example information shown in “An Example in the File System Domain” on page 42, a connector to **/doc/projects/SoDA** should be created to provide a pointer from the SoDA document into the source domain.

Although the top-most directory in the example information is **/doc**, the scope of the example document is completely within **/doc/projects/SoDA**. When inserting SoDA connector elements into your structured template, you create connectors to the starting points in your information source.

To add a connector:

1. Position the cursor in the structure view of the document immediately under the highest-level element.

In the example document, position the cursor as shown below:

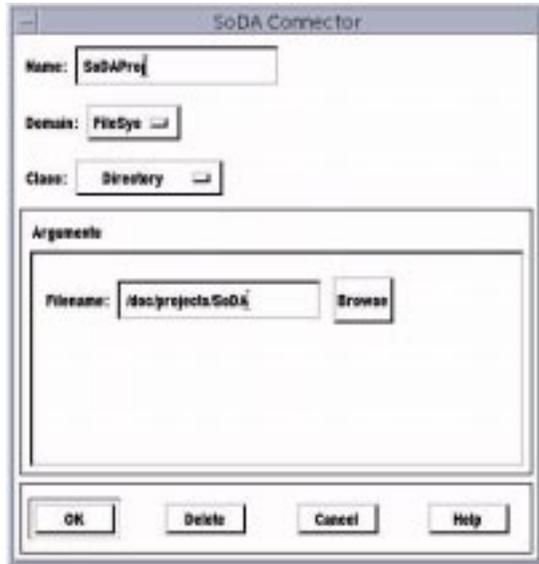


Note: When the cursor is positioned where a connector can be inserted, you will see **SodaConnector** listed in the *Elements catalog*.

2. Choose **SoDA > Connector**.
The Connector dialog box will appear.
3. Choose a name, a source domain, and a class of object for the domain, fill in the resulting arguments for the connector, and click **OK** (or click **Cancel** to dismiss the dialog

without inserting a connector). (For details about the Connector dialog box options, see SoDA's online help.)

The completed Connector dialog for the example looks like this:



The structure view of the example document now looks like this:



You can add multiple connectors to your document. The additional connectors point to information in the same source domain or to a different domain.

Creating Replicators

Replicators are used to create (for each object found in the source) sibling sections in your document. In the example document, one replicator is used to generate the “directory” sections, and one replicator is used to generate the “file” sections. The directory replicator creates a section in the document for each directory it finds in the source. The file replicator creates a section in the document for each file it finds in the source.

Standard Replicators

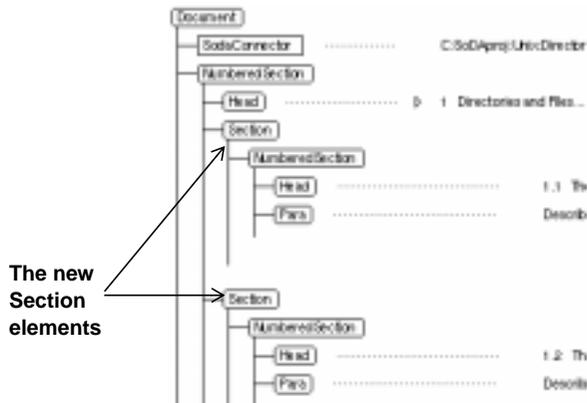
To add a replicator:

1. Make sure that any sections to be replicated are not siblings in the document hierarchy.

For example, numbered sections 1.1 and 1.2 are siblings, and they will both be replicated, yet they will be replicated with *different* replicators. The directory replicator will be for directories in the source, and the file replicator will be for files in the source. Because SoDA uses the hierarchical structure of your document to determine the scope for particular SoDA elements, these two sections cannot be replicated within the current document structure.

To create a valid document structure for the example document, wrap (see *Using FrameMaker+SGML* for details about wrapping elements) the directory and file **NumberedSec-**

tion elements each within its own **Section** element. The resulting structure view looks like this:



The new **Section** elements do not affect how your document looks. They just separate the replicated **NumberedSection** elements.

2. Select the element to be replicated. Ensure that the element and all its children are selected.

In the example, the first element to be replicated is the **NumberedSection** element with the 1.1-numbered heading. When the element is selected, the structure view looks like this:



3. Choose SoDA > Replicator.

The Replicator dialog box will appear.

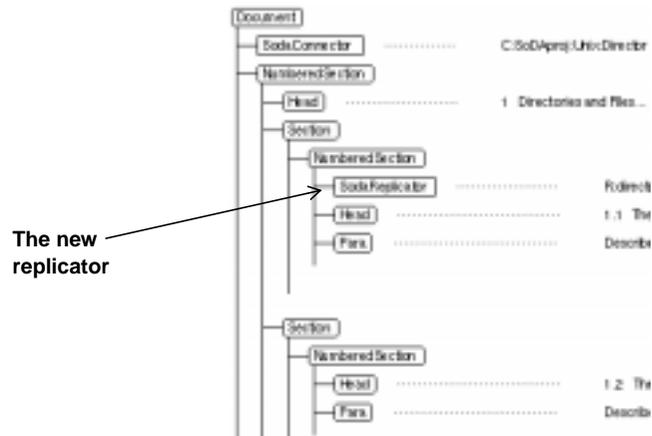
4. Choose a name, a parent SoDA element, what you would like to select from the domain, and the class of object to select and click **OK**, or click **Cancel** to dismiss the dialog without adding a replicator. (For details about the Replicator dialog box options, see SoDA's online help.)

The completed Replicator dialog for the example looks like this:

The screenshot shows the 'Replicator' dialog box with the following configuration:

- Name:** directories
- From:** SoDAAgent
- Recursive:** <None>
- Select:** Contents
- Where is a:** Directory
- And Where:** Logical Operator: AND
- And Where:** <Predicate>
- Ordering:** <Ordering>
- Ordering options:** Unique Only Recursive
- Buttons:** OK, Delete, Cancel, Help

The structure view of the example document now looks like this:



5. Add as many replicators as you need to replicate all the sections in your document.

In the example, an additional replicator is necessary for section 1.2.

Creating Hypertext Link Anchors

In larger documents you may need to document an object from the data source in one location, but reference in one or more other locations. For example, you may want to create a data dictionary for all classes in a Rose model. The relationships of each class refer to other classes within the same document.

SoDA can automatically create hypertext links to support online traversal of a referenced object to the location where it is documented.

Follow these steps:

1. Create the replicator where each link section will serve as the anchor, or destination of the hypertext link. Select the Create Link Anchor option in the replicator dialog box. Be sure to use this feature only once in a document for a given class of objects.

2. Create the text field that refers to the objects documented in the above replicator. Select the Create Link Reference option.
3. Generate the document.
4. Lock the document, using ESC-F-l-k. With the locked document, anytime you click on a displayed field value that was created using Create Link Reference will automatically jump you to the link section where the displayed value is documented, if it exists.

Note: if you do not wish to lock the document, you can traverse to the link anchor by holding down the CTRL key and clicking the right mouse button.

Replicators Within Tables

Using SoDA, table rows and cells can be replicated. Replicating rows and the contents of cells is very much like replicating sections and other document elements.

Adding SoDA elements to tables is different, however, because only one table row within a single table can be replicated.

Setting Up a SoDA Table

Any FrameMaker+SGML table can become a SoDA table (that is, a table with SoDA elements). The following are guidelines for creating a FrameMaker+SGML table for use as a SoDA table:

- The table must be wrapped in a Table element.
- The table must contain at least one row.
- The row to be replicated can be empty.

Adding a Row Replicator

Adding a replicator to a row is similar to adding a replicator to a document section, as described in the procedure below.

1. In the document window, select the first CELL element of the row to be replicated.
2. Choose **SoDA > Replicator**.

The Replicator dialog box is displayed.

3. Define the replicator as you would a section or other element replicator.
See “Creating Replicators” on page 50 for an example of defining a replicator.
4. Click OK.
The replicator is added to the cell in the table row.

Adding a Replicator within a Cell

Adding a replicator within a cell is the same as adding a replicator to any structured element:

1. Add to the CELL element an element to be replicated.
2. Select the element added in step 1.
3. Choose **SoDA > Replicator**.
4. Define the replicator.

(See “Creating Replicators” on page 50 for information about adding replicators to structured elements.)

Replicators within cells can reference their parent row replicator and any other connectors, replicators, and stipulators within the scope of the table.

Document Replicators

The following sections describe two examples of converting a document to a book with a document replicator.

Creating the first document to be replicated

1. Open the example document again and save it as **%SimpleName%Directories.doc**.
2. Add **%SimpleName%Directories.doc** to **example.book**.
(See *Using FrameMaker+SGML* for details.)
3. Select the first **NumberedSection** element and choose **Element > Unwrap**.
4. Delete:
 - The **SodaConnector** element
 - The **Head** element associated with the **NumberedSection** element eliminated in step 3

- ❑ The second **Section** element (the one meant to replicate sections for each file found in the source domain)
 - ❑ The **Para** element
 - ❑ The **Section** element containing the **SodaStipulator** element
 - ❑ The remaining **SodaReplicator** element
5. Select the **Section** element and choose **Element > Unwrap**.
 6. Position your cursor directly under the **Document** element and choose **SoDA > Book Name** to insert a **SodaBookName** element, identifying the book as **example.book**.
 7. Select the **Document** element (the root element of this document), choose **SoDA > Replicator**, and define a replicator to create a document for each directory found in the source domain (see “Creating Replicators” on page 50 for the correct Replicator dialog information).

The resulting document, **%SimpleName%Directories.doc**, looks like this:

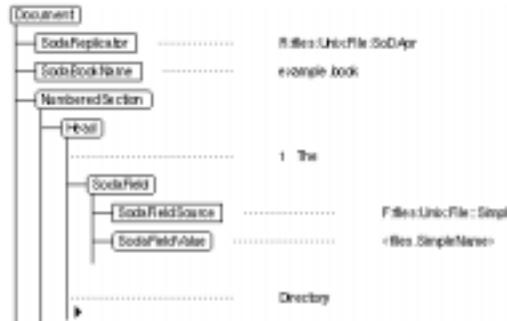


Creating the second document to be replicated

1. Open **%SimpleName%Directories.doc** and save it as **%SimpleName%Files.doc**.
2. Add **%SimpleName%Files.doc** to **example.book**. (See *Using FrameMaker+SGML* for details.)
3. Edit the replicator to create a document for each file found in the source domain (see “Creating Replicators” on page 50 for the correct Replicator dialog information).

- If you changed the name of the replicator in step 3, edit the field so that the correct replicator (the one named in step 3) is referenced in the From option menu.

The resulting document, **%SimpleName%Files.doc**, looks like this:



Creating Fields

Fields are generally used within a region of text to add attribute information generated from the source domain. For example, filenames can be inserted into headings using fields.

In the example document, the name of the project should appear in paragraph 1 between “THE” and “PROJECT.” The name of each directory should appear in paragraph 1.1 and its siblings between “The” and “Directory,” and the name of each file should appear in paragraph 1.2 and its siblings between “The” and “File.”

To add a field:

- Position your cursor precisely within the text of your document.

In the example document, the cursor is positioned between “THE” and “PROJECT” in paragraph 1, as follows:



It is easiest to work in the document window to insert fields.

2. Choose **SoDA > Field**.

The Field dialog box will appear.

3. Choose a parent SoDA element, an attribute, and modifiers and click **OK**, or click **Cancel** to dismiss the dialog without adding a field. (For details about the Field dialog box options, see SoDA's online help.)

The completed Field dialog for the example looks like this:



The structure view of the example document now looks like this:



4. Ensure that your text contains the proper spacing around the field.

In the example, a space exists before the field, but there is no space after the field (before the word “PROJECT”).

If you need to add space after the field:

- a. Position your cursor *after* the field. Ensure that your cursor is not *within* the field.

In the example document, the cursor would be positioned in the structure view like this:



- b. Type a space.

A space is inserted between the field and “PROJECT.”

5. Add as many fields as you need to your replicated and stipulated sections.

In the example, additional fields are necessary in the headings of sections 1.1 and 1.2.

Fields as Hypertext Link References

See “Creating Hypertext Link Anchors” on page 53.

Creating Stipulators

Stipulators are used to add particular information to a document under certain conditions. For example, one of the directories in the example shown on page 42 is the **bin** directory, used to store binaries. In our document, an extra explanatory note should be added to the section that is created for the **bin** directory. The note should only appear, however, in the

section which is created specifically for the **bin** directory. If a **bin** directory does not exist, a section will not be created, and the note will not appear. This specificity can be achieved through the use of a stipulator.

A section is added to the example document which describes the **bin** directory. The document outline now looks like this:

1 DIRECTORIES AND FILES ASSOCIATED WITH THE PROJECT

1.1 The Directory

Describe this directory

Note: The bin directory contains binaries used in document creation.

1.2 The File

Describe this file

The structure view of section 1.1 now looks like this:



To add a stipulator:

1. Select the element to be stipulated. Ensure that the element and all its children are selected.

When the section containing the note is selected, it looks like this:



2. Choose **SoDA > Stipulator**.

The Stipulator dialog box will appear.

3. Choose a name, a parent SoDA element, what you would like to select from the domain, and the class of object to select.
4. Specify a predicate that limits the use of this note to the paragraph created for the **bin** directory only and click OK, or click **Cancel** to dismiss the dialog without adding a stipulator. (For details about the Stipulator dialog box options, see SoDA's online help.)

The completed Stipulator dialog for the example looks like this:



The structure view of section 1.1 now looks like this:



5. Add as many stipulators as you need to stipulate all of the conditional sections in your document.

Editing Generated Documents

It should be a project goal to determine the structure of SoDA documents before team members begin documentation. Making changes to document structure is much more difficult after users have added descriptive text and other manual elements interspersed with the generated elements.

If you have determined that your document is not generating correctly, you will need to change the SoDA elements in your document.

Displaying Hidden Text

Before you can change SoDA elements, you must first display them. SoDA applies condition tags to replicator and stipulator elements during generation. When generation is complete, SoDA “hides” the elements (including their children, text, etc.) to which condition tags have been applied. (See *Using FrameMaker+SGML* for details about conditional text and condition tags.)

SoDA uses two tags to hide items in a document:

- **SodaMaster**

This tag hides master elements, for example, entire sections containing replicators. (Connectors and fields are always visible in the document. Links are always visible after the first generation when they are inserted.)

- **SodaStipulator**

This tag hides stipulated sections where the stipulator is false. (Stipulators that resolve to something remain visible in the document section that is created.)

To display hidden text:

1. Choose **Special > Conditional Text** and click the **Show/Hide** button. You will see the **FrameMaker+SGML - Show/Hide Conditional Text** dialog box. The two SoDA-specific conditional

tags (SodaMaster and SodaStipulator) will be listed in the Hide scroll list.

2. Double-click the tags hiding the text you wish to display.
 - ❑ To display master elements, double-click SodaMaster.
 - ❑ To display hidden stipulators, double-click SodaStipulator.

Any or all of the tags can be displayed at once.

3. Click the **Set** button.

You will see the hidden text displayed in your document.

- ❑ Master elements appear underlined.
- ❑ Stipulated text appears with a line striking through it.

Note that a document generated in report mode contains no hidden text. Since reports are not intended to be regenerated, they should not be edited.

Editing SoDA Elements

Once master elements are displayed, you can edit any SoDA element in your document. You may want to make changes to SoDA elements for the following reasons:

- If you defined an element incorrectly when you created it, you can delete the results obtained by generating that element, edit the SoDA element, and regenerate your document.

Editing a Single, Specific SoDA Element

1. Select the appropriate element in the structure view.
 - ❑ To edit a connector, select the connector.
 - ❑ To edit a replicator, select the replicator or the section which contains the replicator.
 - ❑ To edit a stipulator, select the stipulator or the section which contains the stipulator.
 - ❑ To edit a field, select the entire field (including the **SodaFieldSource** and the **SodaFieldValue** elements).
2. If supported on your version of SoDA, you may double-click on the element in the structure view to display the edit

dialog, or choose the appropriate command from the SoDA menu:

- To edit a connector, choose **SoDA > Connector**.
 - To edit a replicator, choose **SoDA > Replicator**.
 - To edit a stipulator, choose **SoDA > Stipulator**.
 - To edit a field, choose **SoDA > Field**.
3. Make the necessary corrections in the dialog box and click **OK**, or click **Cancel** to dismiss the dialog without making changes to the selected element.



Appendix A

Special Setup for Prior Frame Users

This appendix contains instructions for many of the exercises in SoDA's online tutorial.

If you already use FrameMaker or FrameMaker+SGML

You do not need to complete any special, personal setup before starting SoDA. You only need to read this section if you already use FrameMaker or FrameMaker+SGML and have customized your setup by adding `fminit` or `fmtemplates` directories in your home directory.

API Clients

SoDA depends on an `apiclients` file that is installed in:

```
$FMHOME/fminit/lang/APIArea/apiclients
```

where, for example, *lang* is `usenglish`. SoDA will not start properly without the entries in the `apiclients` file named:

- Soda
- **Generator** (on certain platforms only)
- Element Catalog Manager
- Structure Generator
- CMVC (for when you are using SoDA with Rational Apex[®])
- **Command Line Server** (if SoDA is installed with Apex[®] or Rose[®])

If you have your own personal `apiclients` file, you need to merge these entries into it before starting SoDA.

Templates

SoDA includes templates that are installed to the directory:

```
$FMHOME/fmunit/lang/Templates/SoDA
```

If you have your own personal **fmtemplates** directory, you can merge the SoDA templates into it so that they are conveniently accessible when you click the **NEW** button in the main FrameMaker+SGML window. One easy way to accomplish this is to add a symbolic link named **SoDA** in your **fmtemplates** directory.



Appendix B

Instructions for Tutorial Exercises

This appendix contains instructions for many of the exercises in SoDA's online tutorial:

- “Exercise 1-1: Inserting Connectors” on page 68
- “Exercise 1-2: Inserting Fields” on page 72
- “Exercise 1-3: Inserting a Replicator” on page 78
- “Exercise 2-1: Generating a Document” on page 85
- “Exercise 3-1: Regenerating a Document” on page 90
- “Exercise 4-1: Replicating Table Rows” on page 97
- “Exercise 4-2: More SoDA Elements in Table Cells” on page 102
- “Exercise 5-1: Generating a Book” on page 107

Using the Online Tutorial

To use the tutorial:

1. Choose **SoDA Tutorial** from the **SoDA Help** menu on any SoDA document window, or click on **Tutorial** in the SoDA's main online help menu.
2. Follow the instructions in the tutorial document that appears online.

Keep this appendix handy for the tutorial exercises. Many of the exercises in the tutorial will refer you to the following, printed instructions to avoid either:

- Overcrowding your workstation screen
- The inconvenience of not being able to read instructions while a SoDA dialog box is displayed

Preparation for Each Exercise

The following instructions for preparing your display apply to all of the exercises.

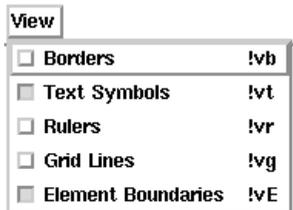
Element Catalog and Structure View

If you have not already done so, click on the two top buttons at the upper right side of your document window to display the element catalog and structure view:



Text Symbols and Element Boundaries

You can also use the **View** menu to turn on text symbols and/or element boundaries if you would like:



Exercise 1-1: Inserting Connectors

You do this exercise in **m1_ex1.doc**, which should already be open on your workstation screen. Click in the document window so that you have an insertion point and the document's structure is displayed in the structure view. You will see in the structure view that your document currently contains no SoDA elements.

Imagine that your document is a structured FrameMaker+SGML template that you want to transform into

a SoDA template. You typically begin that job by adding connectors.

Instructions

1. In either the document window or structure view, position the cursor within or below the existing **Title** or **Body** elements:



Notice that **SodaConnector** never appears in the element catalog, because the EDD for this document does not allow you to insert connectors in these places.

2. In the structure view, position the cursor immediately below the highest-level **Report** element:

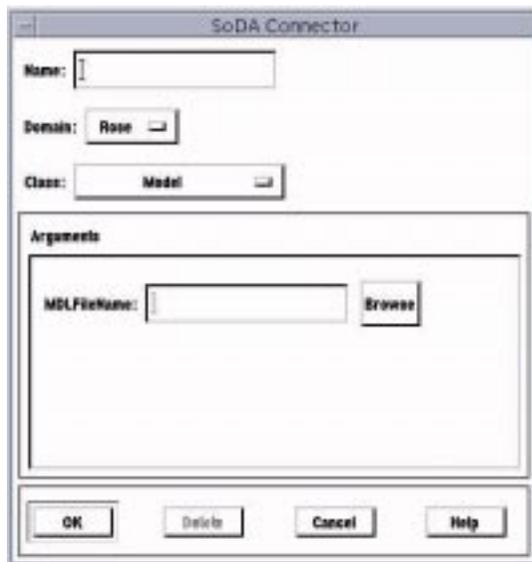


SodaConnector now appears with a heavy check mark in the element catalog, showing that it is valid to insert a connector in this position:



3. Choose **Connector** from the SoDA menu.

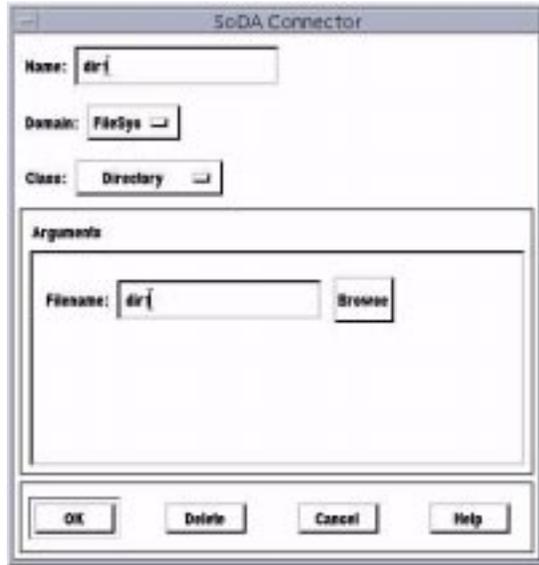
The SoDA Connector dialog box will appear. (The originally listed domain and class may be different on your system. It depends on which domains you have installed.)



- Note that you have already been provided with the following FileSys objects—located in the same directory as **m1_ex1.doc**—for this exercise:

```
dir1/  
  file1  
  file2  
  file3
```

Connect to the **dir1** directory by filling in the dialog box as shown below and clicking OK:



Notice how the dialog box changes as you make your choices. When you specify the source domain, the classes in that source domain appear. Then when you specify the class, the arguments related to that class appear.

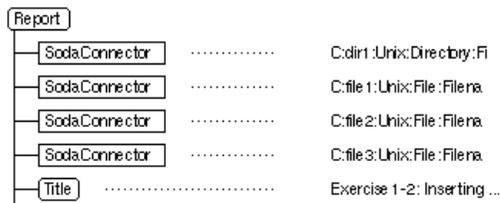
- Repeat, this time creating a connector named **file1** to the file **dir1/file1**. The appearance of the new connectors in your structure view should be similar to the following:



When you are finished practicing, choose **Close** from the **File** menu to exit **m1_ex1.doc**. (You do not need to save your changes, because subsequent exercises do not depend on the changed document.)

Exercise 1-2: Inserting Fields

You do this exercise in **m1_ex2.doc**, which should already be open on your workstation screen. Click in the document window so that you have an insertion point and the document's structure is displayed in the structure view. You will see in the structure view that your document already contains four connectors:

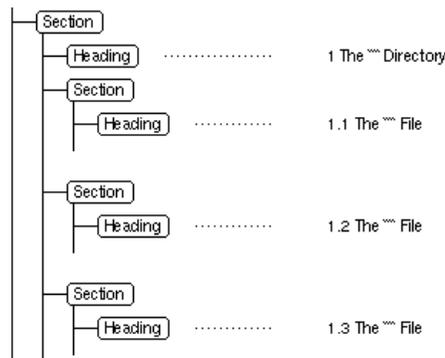


They connect to the following FileSys objects—located in the same directory as **m1_ex2.doc**—which have been provided for this exercise:

```
dir1/
  file1
  file2
  file3
```

Imagine that you are building a SoDA template. You have already inserted connectors to a directory and some files in it. Now you are adding a section for the directory and a subsection for each file.

The **Section** and **Heading** elements are in place:



You simply need to add the names of the FileSys objects to the headings for each section:

1 The "" Directory

1.1 The "" File

1.2 The "" File

1.3 The "" File

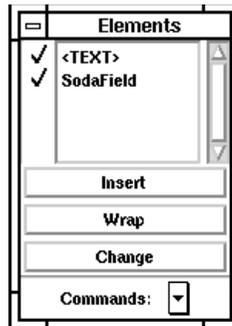
For that, you will insert SoDA fields.

Instructions

1. In the document window, position the cursor in between the double quotes in the first **Heading** element:

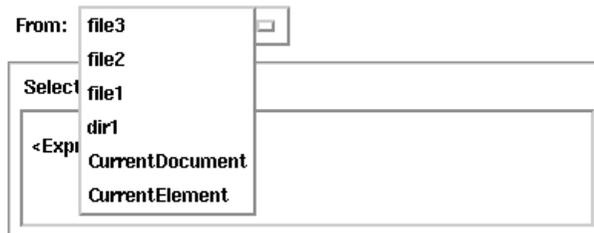
1 The " Directory

SodaField now appears with a heavy check mark in the element catalog, showing that it is valid to insert a field in this position:



2. Choose **Field** from the **SoDA** menu.
The SoDA Field dialog box will appear. It contains three areas—**From**, **Select**, and **Modifiers**—for you to fill in:
 - a. Use the **From** option menu to associate the field with a connector.¹

To choose a connector, move the cursor to the **From** option menu, press the left mouse button to pop up the menu, drag to your choice, and release the mouse button.

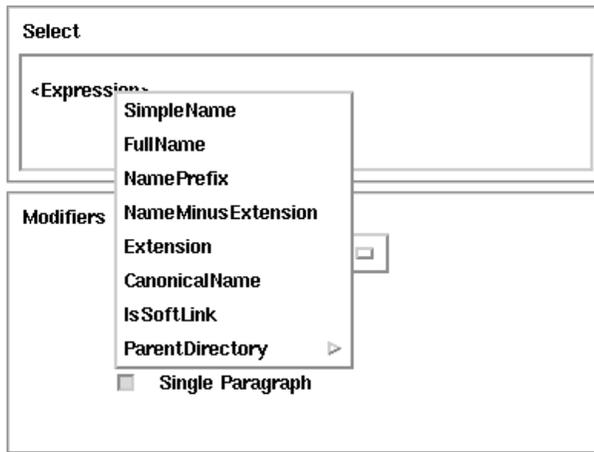


For your current heading, the appropriate choice is **dir1**.

- b. Use the **Select** area to specify the attribute you wish to extract from the object (to which the connector is connected).

To specify an attribute, move the cursor over the word **<Expression>** in the **Select** area, press the right mouse button to pop up a menu of attributes that are available for the object, drag to your choice, and release the mouse button.

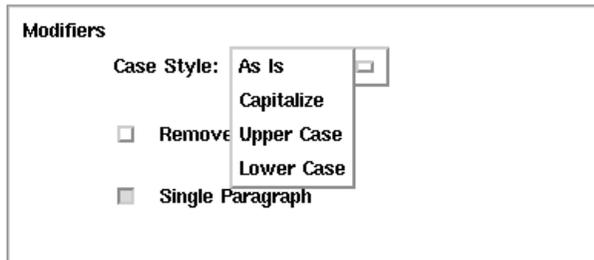
¹Or replicator or stipulator, to be described later and utilized in future exercises. The other choices—**CurrentDocument** and **CurrentElement**—are for use with the Frame domain, which is not utilized in this tutorial's exercises.



Because you specified the connector named `dir1`, which is connected to a FileSys directory, attributes² of FileSys directories are displayed.

For your current heading, the appropriate choice is **SimpleName**.

- c. Use the **Modifiers** area to control how the extracted text³ will be displayed in your document.



The **Case Style** option menu allows you to:

- Leave the case of the extracted text as is

²And, strictly speaking, relationships. At the bottom of the menu, **ParentDirectory** is a relationship. It specifies the parent directory of the object, and its submenu displays attributes of the parent directory.

³Or graphic. This tutorial's exercises do not utilize graphics.

- Capitalize the first letter of each word
- Convert all letters to upper case
- Convert all letters to lower case

Turning on the **Remove Punctuation** option will filter periods, commas, underscores, and so on out of the extracted text.

Turning on the **Single Paragraph** option will treat the extracted text as a single paragraph even if it contains newlines.

For your current heading, the default choices—**Case Style As Is**, do not **Remove Punctuation**, and treat as **Single Paragraph**—are appropriate.

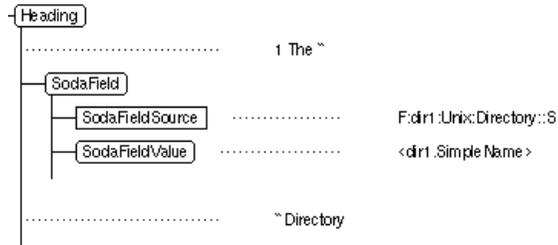
3. Finish completing the dialog box as shown:

The image shows a dialog box titled "SoDA Field". It has a "From:" field with the value "dir1". Below that is a "Select" field containing "SimpleName". Underneath is a "Modifiers" section with three options: "Case Style" is set to "As Is", "Remove Punctuation" is unchecked, and "Single Paragraph" is checked. At the bottom of the dialog are four buttons: "OK", "Delete", "Cancel", and "Help".

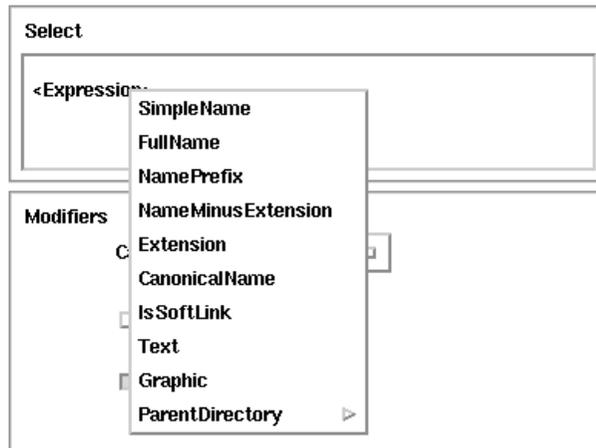
Then click **OK** to insert the field. It will appear in your document window as follows:

1 The "<dir1.SimpleName>" Directory

And it will appear in your structure view as follows:



- Repeat for each of the three subsections, specifying each of the connectors named **file1**, **file2**, and **file3** in turn. Select the **SimpleName** attribute and default modifiers for all. Notice that because you are now specifying connectors that are connected to FileSys files, a slightly different list of attributes is displayed than when you were doing a directory before:



(**Text** and **Graphic** attributes are added for FileSys files, although you will not choose them now for your headings.)

All of the new fields will appear in your document window as follows:

1 The "<dir1.SimpleName>" Directory

1.1 The "<file1.SimpleName>" File

1.2 The "<file2.SimpleName>" File

1.3 The "<file3.SimpleName>" File

5. **Optional sneak preview:** If you would like to see how SoDA fills in these fields even though document generation has yet to be discussed, you can:
- a. Choose **Generate Document** from the **SoDA** menu.
 - b. Click **OK** in the SoDA Generator dialog box that appears. The fields will be filled in as follows:

1 The "dir1" Directory

1.1 The "file1" File

1.2 The "file2" File

1.3 The "file3" File

When you are finished practicing, choose **Close** from the **File** menu to exit **m1_ex2.doc**. (You do not need to save your changes, because subsequent exercises do not depend on the changed document.)

Exercise 1-3: Inserting a Replicator

You do this exercise in **m1_ex3.doc**, which should already be open on your workstation screen. Click in the document window so that you have an insertion point and the document's structure is displayed in the structure view. You will see in the

structure view that your document already contains a connector:



It connects to the FileSys directory **dir1**—located in the same directory as **m1_ex3.doc**—which is shown below along with the files that it contains:

```
dir1/
  file1
  file2
  file3
  readme
```

Imagine that you are building a SoDA template. You have already inserted a connector to a directory and created a section for the directory. Now you are adding a subsection for each file in the directory.

Prototypical **Section**, **Heading**, and **Body** elements for a subsection are in place:



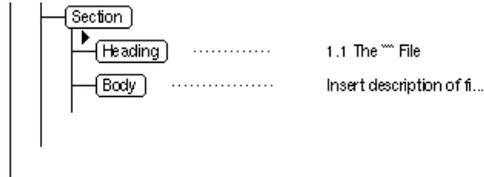
You simply need to:

1. Replicate the subsection structure for each file
2. Add a filename to each of the replicated headings

You could do this manually for each file using only connectors and fields (as illustrated in “Exercise 1-2: Inserting Fields” on page 72). However, that approach is tedious and difficult to maintain. A better solution is to insert a SoDA replicator and field in the prototypical subsection, and let SoDA do the rest.

Instructions

1. In the structure view, position the cursor immediately after the last **Section** element:



SodaReplicator now appears with a heavy check mark in the element catalog, showing that it is valid to insert a replicator in this position:



2. Click on the last section element to select it:



(**SodaReplicator** will no longer appear with a heavy check mark in the element catalog, because it is not a valid *replacement* for the selected structure. That is OK. When inserting a replicator, SoDA allows you to select the structure you want to replicate rather than just placing the cursor where the replicator is to be inserted.)

3. Choose **Replicator** from the **SoDA** menu.

The SoDA Replicator dialog box will appear. It contains several areas for you to fill in:

- a. In the **Name** text-entry box, enter `dir1_files`.
- b. Use the **From** option menu to associate the replicator with a connector.⁴

To choose a connector, move the cursor to the **From** option menu, press the left mouse button to pop up the menu, drag to your choice, and release the mouse button.

For the current task, the appropriate choice is `dir1`.

- c. Use the **Recursive** option menu when you want to associate this replicator with another replicator, making it recursive. The **Recursive** option menu will be grayed-out in your dialog box, showing that it is unavailable, because there are no other replicators in the document.
- d. Use the **Select** area to specify the relationship you wish to follow from the object (to which the connector is connected).

To specify a relationship, move the cursor over the word `<Expression>` in the **Select** area, press the right mouse button to pop up a menu of relationships that are available for the object, drag to your choice, and release the mouse button.

Because you specified the connector named `dir1`, which is connected to a `FileSys` directory, relationships of `FileSys` directories are displayed.

For the current task, the appropriate choice is `Contents`. The contents of a directory is all the objects that reside directly in the directory.

- e. Use the **Where Is A** option menu to specify the class of objects that the replicator can select.

⁴Or in some cases, another replicator or stipulator. The other choices—`CurrentDocument` and `CurrentElement`—are for use with the `Frame` domain, which is not utilized in this tutorial's exercises.

To do so, move the cursor to the **Where Is A** option menu, press the left mouse button to pop up the menu, drag to your choice, and release the mouse button.

In this case, pressing the left mouse button will pop up a list of classes of FileSys objects. For the current task, the appropriate choice is **File**.

- f. Use the **And Where** area when you want to specify a predicate that further restricts the set of objects that the replicator can select. For example, you could specify files with only certain extensions in their names.

For the current task, no predicate is required.

- g. Use the **Ordering** area when you want to sort the objects that the replicator selects. For example, you can:
- Sort alphanumerically or numerically
 - Make the sort case-sensitive or case-insensitive
 - Reverse the order of the sort (such as going from Z to A instead of A to Z).

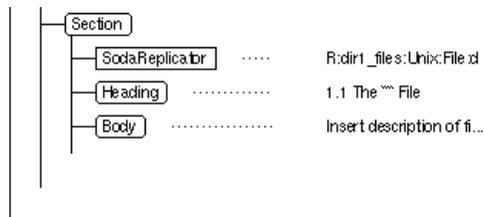
For the current task, no sorting is required.

4. Finish completing the dialog box as shown:



Then click **OK** to insert the replicator.

It will appear in your structure view as follows:



5. In the document window, position the cursor in between the double quotes in the last **Heading** element:

1.1 The “**File**”

Insert description of file here.

6. Choose **Field** from the **SoDA** menu.

The SoDA Field dialog box will appear.

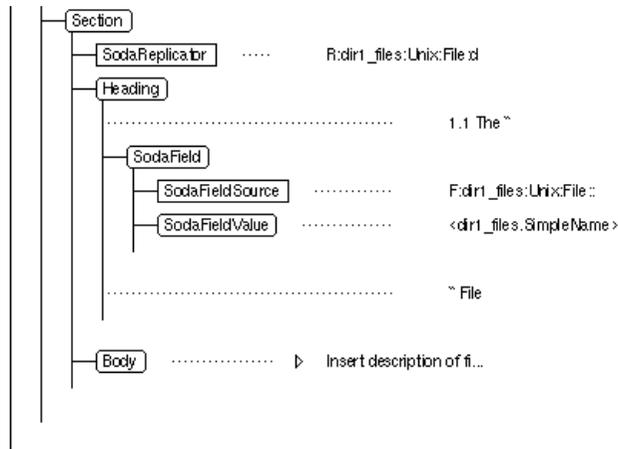
- a. In the **From** option menu, associate the field with the replicator named `dir1_files` that you just inserted.
- b. In the **Select** area, specify the **SimpleName** attribute.
- c. In the **Modifiers** area, use the default choices—**Case Style As Is**, do not **Remove Punctuation**, and treat as **Single Paragraph**.

Then click **OK** to insert the field. It will appear in your document window as follows:

1.1 The “<dir1_files.SimpleName>” File

Insert description of file here.

And it will appear in your structure view as follows:



7. **Optional sneak preview:** If you would like to see how SoDA replicates this section and fills in this field even though document generation has yet to be discussed, you can:
 - a. Choose **Generate Document** from the **SoDA** menu.

- b. Click **OK** in the SoDA Generator dialog box that appears. The replicated sections will appear as follows in your document window:

1 The “dir1” Directory

This section is to contain subsections that describe each file in the directory.

1.1 The “file1” File

Insert description of file here.

1.2 The “file2” File

Insert description of file here.

1.3 The “file3” File

Insert description of file here.

1.4 The “readme” File

Insert description of file here.

When you are finished practicing, choose **Close** from the **File** menu to exit **m1_ex3.doc**. (You do not need to save your changes, because subsequent exercises do not depend on the changed document.)

Exercise 2-1: Generating a Document

You do this exercise in **m2_ex1.doc**, which should already be open on your workstation screen. Click in the document window so that you have an insertion point and the document’s structure is displayed in the structure view. You will see in the structure view that your document already contains a connector:

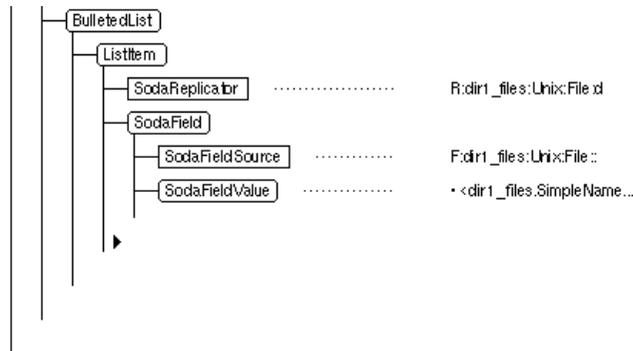


It connects to the FileSys directory **dir1**—located in the same directory as **m2_ex1.doc**—which is shown below along with the files that it contains:

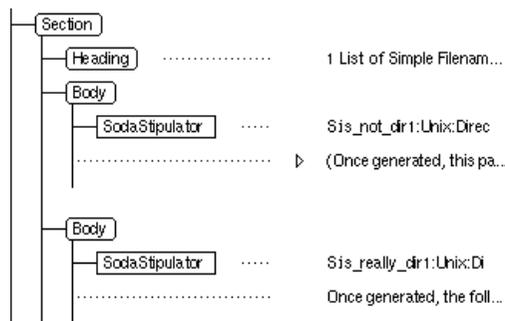
```
dir1/
  file1
  file2
  file3
  readme
```

Imagine that you have built a SoDA template. You have already inserted:

- A connector to a directory
- A replicator and field that will create a list of files in the directory:



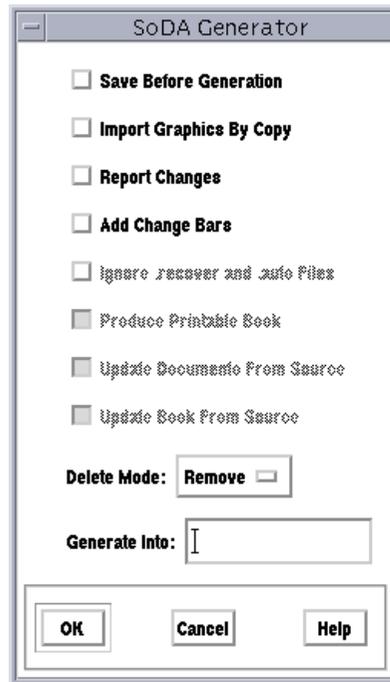
- Two stipulated body paragraphs. One or the other will appear in the generated document depending on the connector:



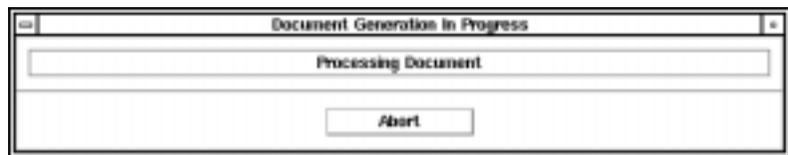
Now you are ready to quickly test the template by generating a document from it. You will do so in place, then close the template without saving the changes from generation.

Instructions

1. Position the cursor in the document.
2. Choose **Generate Document** from the **SoDA** menu.
The SoDA Generator dialog box will appear:

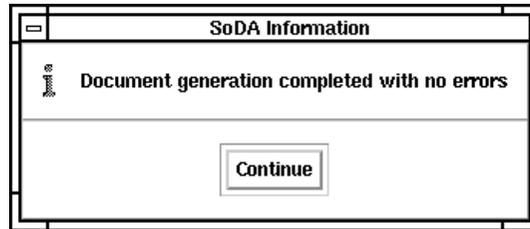


3. Click **OK**.
A Document Generation in Progress dialog box will appear:



It will continue to display progress messages until generation completes, or until you click the **Abort** button to abandon the generation. Do *not* abort this generation.

Finally, a SoDA Information dialog box will appear. It should say that there were no errors:



4. Click **Continue**.

The changes will appear in the document.

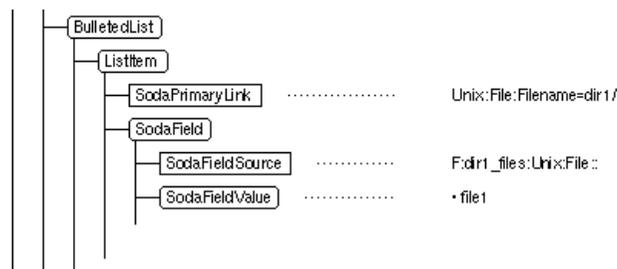
5. Inspect the generated document. Notice in the document window that one of the stipulated body paragraphs has disappeared, and that a list of files has appeared:

1 List of Simple Filenames

Once generated, the following bulleted list will contain an item for each **file*** in the **dir1** directory:

- file 1
- file 2
- file 3

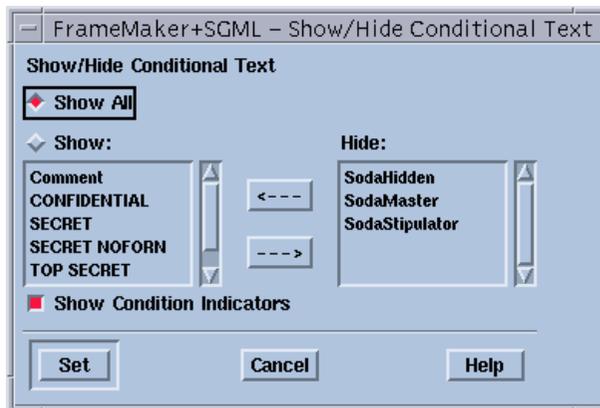
In the structure view, notice that the list items contain links instead of replicators, and that simple filenames have been stored in the field values:



Also notice that the stipulator remains in the generated document for use when it is regenerated:



6. Choose **Conditional Text** from the **Special** menu.
The **Conditional Text** dialog box will appear.
7. Click the **Show/Hide** button.
The **Show/Hide Conditional Text** dialog box will appear:



8. Click to turn on **Show All**, and then click **Set**.
The previously hidden elements will now be displayed in the document.
9. Once again inspect the generated document. Notice in the document window that the other stipulated body paragraph has reappeared with the **SodaStipulator** condition tag

applied to it, and that the master list item has reappeared with the **SodaMaster** condition tag applied to it:

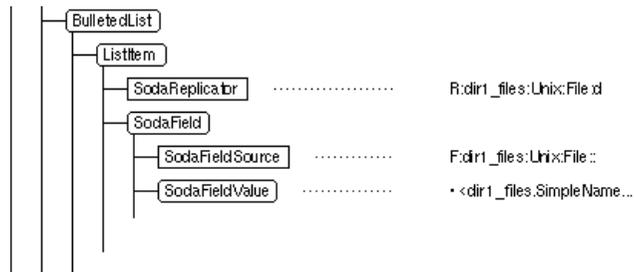
1 List of Simple Filenames

(Once generated, this paragraph will disappear so long as the document is properly connected to the **dir1** directory.)

Once generated, the following bulleted list will contain an item for each **file*** in the **dir1** directory:

- <dir1_files.SimpleName>
- file 1
- file2
- file3

In the structure view, notice that the master list item still contains its original replicator, not a link as in the generated list items, and its original field:



When you are finished, choose **Close** from the **File** menu to exit **m2_ex1.doc**. (You do not need to save your changes, because subsequent exercises do not depend on the changed document.)

Exercise 3-1: Regenerating a Document

You do this exercise in **m3_ex1.doc**, which should already be open on your workstation screen. Click in the document window so that you have an insertion point and the document's structure is displayed in the structure view. You will see in the

structure view that your document already contains a connector:



It connects to the FileSys directory **dir1**—located in the same directory as **m3_ex1.doc**—which is shown below along with the files that it *originally contained*:

```
dir1/
  file1
  file2
  file3
  readme
```

Imagine that you built a SoDA template and generated **m3_ex1.doc** from it. Your document now contains a replicated section for each file, and in each replicated section, a descriptive paragraph that you have added.

Now you are ready to regenerate the document, but since you first generated it, one of the files has been moved. The file previously named **file3** is now named **moved_file3**.

Instructions

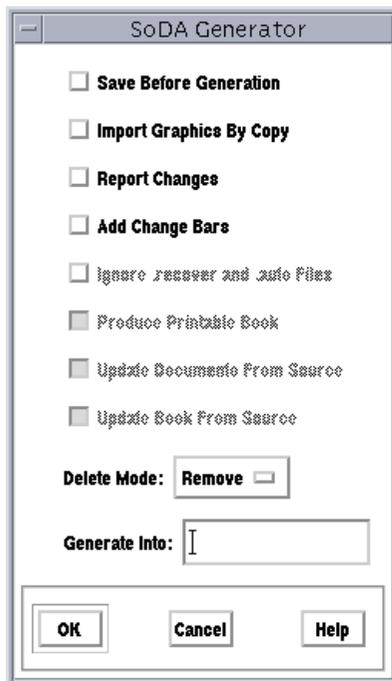


Caution: *The following steps instruct you to regenerate your document multiple times, with different options, so that you can see how SoDA performs in a variety of situations. In between, you are instructed to revert your document to the previously saved version so that you can start over again. Do not save your document when the instructions do not tell you to, or you won't be able to get back to your original version in subsequent steps.*

Regenerating with the Delete Mode Set to Remove

1. Peruse the replicated sections in your document, making a special note of the section for **file3**.
2. Position the cursor anywhere in the document.
3. Choose **Generate Document** from the **SoDA** menu.

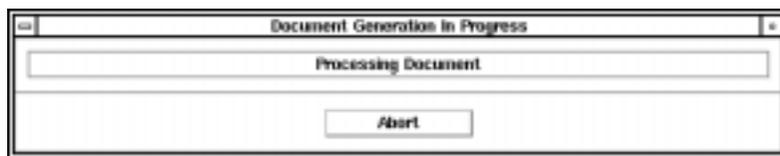
The SoDA Generator dialog box will appear:



Make sure that the **Delete Mode** is **Remove** and that the other options are toggled off. (Those are the conventional default settings. Some of the options will be grayed out because they apply only to books.)

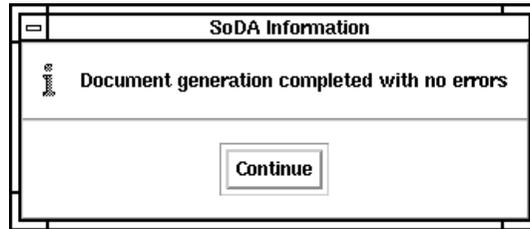
4. Click **OK**.

A Document Generation in Progress dialog box will appear:



It will continue to display progress messages until generation completes, or until you click the **Abort** button to abandon the generation. Do *not* abort this generation.

Finally, a SoDA Information dialog box will appear. It should say that there were no errors:



5. Click **Continue**.

The changes will appear in the document.

6. Inspect the generated document. Notice in the document window that the section for **file3** was deleted, and that a new section for **moved_file3** was added:

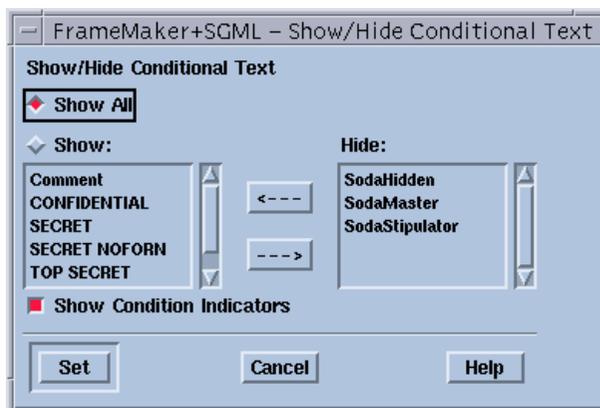
1.4 The "moved_file3" File

Insert description of file here.

You lost the text that you added to the section for **file3**. Can you retrieve it?

7. Choose **Conditional Text** from the **Special** menu.
The Conditional Text dialog box will appear.
8. Click the **Show/Hide** button.

The Show/Hide Conditional Text dialog box will appear:



9. Click to turn on **Show All**, and then click **Set**.
The previously hidden elements will now be displayed in the document.
10. Once again inspect the generated document. Notice in the document window that the master section has reappeared with the **SodaMaster** condition tag applied to it:

1.1 The “<dir1_files.SimpleName>” File

Insert description of file here.

However, the section for **file3** has *not* reappeared with the **SodaHidden** condition tag applied to it because you specified **Remove** as the **Delete Mode** in the SoDA Generator dialog box.

11. Choose **Revert to Saved** from the **File** menu to revert to your original document, thereby recovering your lost text. Now you can begin again.

Regenerating with the Delete Mode Set to Hide

1. Once again position the cursor anywhere in the document, without selecting anything, and choose **Generate Document** from the **SoDA** menu.

The SoDA Generator dialog box will appear. Set the **Delete Mode** to **Hide** and make sure that the other options are toggled off.

2. Click **OK**.
A Document Generation in Progress dialog box will appear, followed by a SoDA Information dialog box that should say there were no errors.
3. Click **Continue**.
The changes will appear in the document.
4. Inspect the generated document. Once again notice in the document window that the section for **file3** was deleted, and that a new section for **moved_file3** was added:

1.4 The “moved_file3” File

Insert description of file here.

You lost the text that you added to the section for **file3**. Can you retrieve it?

5. Choose **Conditional Text** from the **Special** menu.
The Conditional Text dialog box will appear.
6. Click the **Show/Hide** button.
The Show/Hide Conditional Text dialog box will appear.
7. Click to turn on **Show All**, and then click **Set**.
The previously hidden elements will now be displayed in the document.
8. Once again inspect the generated document. Notice in the document window that the master section has reappeared with the **SodaMaster** condition tag applied to it. Moreover, the section for **file3** has reappeared with the **SodaHidden** condition tag applied to it because you specified **Hide** as the **Delete Mode** in the SoDA Generator dialog box:

1.4 The “file3” File

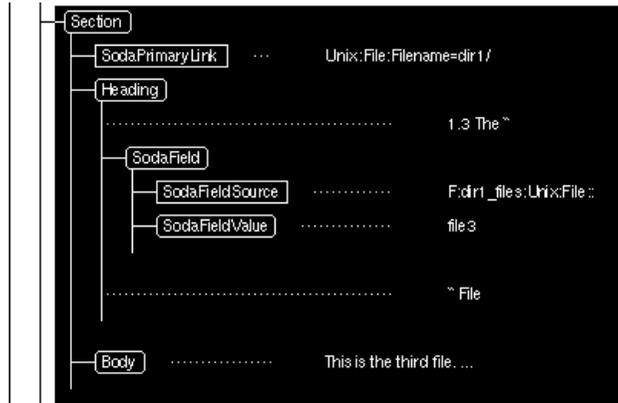
~~This is the third file. Imagine that you have edited this generated document, working long and hard to describe this file. The text in this paragraph would be text that you have entered manually and you would not want to lose it when regenerating the document. Beware, for the name of this file has changed!~~

Thus, you could retrieve your lost text, making it unconditional and pasting it into the new section for **moved_file3**. Instead though, let us pursue not losing the text in the first place.

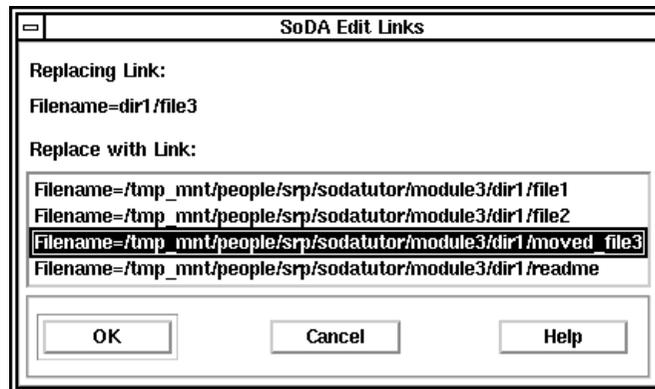
9. Once again choose **Revert to Saved** from the **File** menu to revert to your original document.

Editing a Link Before Regenerating

1. In the structure view, click on the next-to-last **Section** element to select the entire replicated section for **file3**:



2. Choose **Edit Link** from the **SoDA > Utilities** menu. The **Edit Link** dialog box will appear.



3. Click on the link for **moved_file3** to select it. Then click **OK**.
4. Once again position the cursor anywhere in the document, and choose **Generate Document** from the **SoDA** menu. The **SoDA Generator** dialog box will appear. Set the **Delete Mode** to **Hide** and make sure that the other options are toggled off.
5. Click **OK**.

A Document Generation in Progress dialog box will appear, followed by a SoDA Information dialog box that should say there were no errors.

6. Click **Continue**.

The changes will appear in the document.

7. Inspect the generated document. This time the original section for **file3** remains, but the new filename—**moved_file3**—has been inserted into the section heading:

1.3 The “moved_file3” File

This is the third file. Imagine that you have edited this generated document, working long and hard to describe this file. The text in this paragraph would be text that you have entered manually and you would not want to lose it when regenerating the document. *Beware, for the name of this file has changed!*

The text that you added to the section for **file3** has not been lost.

When you are finished, choose **Close** from the **File** menu to exit **m3_ex1.doc**. (You do not need to save your changes, because subsequent exercises do not depend on the changed document.)

Exercise 4-1: Replicating Table Rows

You do this exercise in **m4_ex1.doc**, which should already be open on your workstation screen. Click in the document window so that you have an insertion point and the document’s structure is displayed in the structure view. You will see in the structure view that your document already contains a connector:



It connects to a FileSys file named **passwd**—located in the same directory as **m4_ex1.doc**—which contains records with the following colon-separated fields:

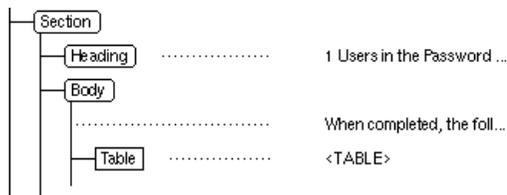
1. Login name

2. Encrypted password
3. User ID
4. Group ID
5. User's name
6. Full pathname of user's home directory
7. Login shell

Imagine that you are building a SoDA template for a system-administration report. You have already gotten a copy of your system's `/etc/passwd` file, added a “**#FIELD_DELIMITER :**” directive to it, and inserted a connector to it in your template. The exact contents of your local copy of the `passwd` file are now:

```
#FIELD_DELIMITER :
root:xI7Aav3rUFvEg:0:1:Operator:/:/bin/csh
sue:aEhHkHK8re4:101:100:Susan Jones:/usr/sue:/bin/csh
bill:UipT64EgDxm4:102:100:William Smith:/usr/bill:/bin/sh
steve:abjYi0PE31bV:103:100:Stephen Johnson:/usr/steve:/bin/ksh
judy:PQwy78zC5g:201:200:Judy Watson:/usr/judy:/bin/csh
chris:slkurnv4TYJ:202:200:Chris Paulsen:/usr/chris:/bin/csh
jane:kKEi57Kyoe:301:300:Jane Mathews:/usr/jane:/bin/csh
bob:qwUB6T01kV:401:400:Robert Duran:/usr/bob:/bin/ksh
mike:R37FUsnYW6:402:400:Michael Pollard:/usr/mike:/bin/sh
```

You have also created a table with columns for login names, user names, and groups:



Now you would like to fill in that table with a row for each record in the `passwd` file. You need to add structure and SoDA elements to some of the table cells, and then generate the document.

Instructions

1. In the document window, click in each cell of the table:

Login Name	User Name	Group

Notice that the structure view remains empty because none of the cells in the table are structured. More precisely, each table cell contains its own text flow, and those flows are unstructured.

There is no need to add structure to the cells in the heading row because you do not need SoDA to process the heading row in any way. However, you do need to add structure and SoDA elements to the cells in the body row because you want to replicate the body row.

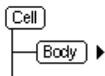
2. Add structure to each cell in the body row:
 - a. In the document window, click in a cell in the body row.
You will be in a **CellBody** paragraph in an unstructured text flow.
 - b. In the element catalog, select **Cell** and click the **Wrap** button.

In the previously empty structure view, a **Cell** bubble will appear. Click to its right:



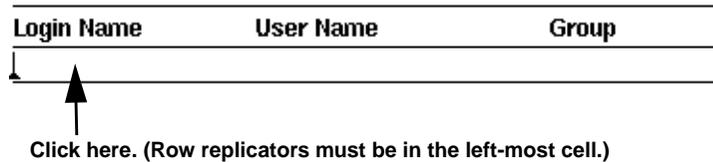
- c. Then, back in the element catalog, select **Body** and click the **Insert** button.

The cell now contains a valid structure that will suffice for the current task. It will appear in your structure view as follows:

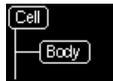


- d. Repeat for the other cells in the body row (or to expedite things, copy the structure you just created and paste it into the other body cells).

3. Now you are ready to add a row replicator:
 - a. In the document window, click in the *left-most* cell of the body row:



- b. In the structure view, click on the **Cell** bubble to select it:



- c. Choose **Replicator** from the **SoDA** menu.

The SoDA Replicator dialog box will appear. Fill it in as follows:

- i. In the **Name** text-entry box, enter **users**.
- ii. On the **From** option menu, choose **password_file**.
- iii. In the **Select** area, choose **Records**.
- iv. On the **Where Is A** option menu, choose **FileRecord**.
- v. In the **Ordering** area, specify **Alphanumeric on UniqueKey**.

Then click **OK** to insert the replicator. It will appear in your structure view as follows:



Note that SoDA will treat this replicator as a row replicator because it:

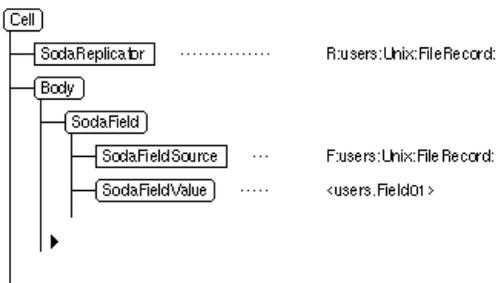
- Appears in the left-most cell of a table row
- Is attached to the cell's highest-level element

When you generate a document that contains a row replicator, entire table rows are replicated, not just structures within a single table cell.

4. Now you are ready to add a SoDA field to each cell in the body row.
 - a. In the document window, click in the left-most cell of the body row (under the “Login Name” heading). (You might still be there from the previous step.)
 - b. In the structure view, click to the right of the **Body** bubble, thereby obtaining an insertion in the **Body** element where you can insert text or a SoDA field:



- c. Choose **Field** from the **SoDA** menu. The SoDA Field dialog box will appear. Fill it in as follows:
 - i. On the **From** option menu, choose **users**.
 - ii. In the **Select** area, choose **Field01**.
 Then click **OK** to insert the field. It will appear in your structure view as follows:



- d. Repeat for the middle cell of the body row (under the “User Name” heading), but choose **Field05** in the **Select** area of the SoDA Field dialog box.
- e. Repeat for the right-most cell of the body row (under the “Group” heading), but choose **Field04** in the **Select** area of the SoDA Field dialog box.

5. That's it. You are ready to generate the document:
 - a. As a matter of habit, choose **Save** from the **File** menu to save your template so that you will have a copy of it in case you want to reuse it or want to correct problems in it and start over. Then choose **Save As** from the **File** menu to create a copy of the template to generate. For this exercise, name it **m4_ex1_gen.doc**.
 - b. Choose **Generate Document** from the **SoDA** menu.
 - c. Click **OK** in the SoDA Generator dialog box that appears.
 - d. Click **Continue** in the SoDA Information dialog box that appears.

The replicated table rows will appear as follows in your document window:

Login Name	User Name	Group
bill	William Smith	100
bob	Robert Duran	400
chris	Chris Paulsen	200
jane	Jane Mathews	300
judy	Judy Watson	200
mike	Michael Pollard	400
root	Operator	1
steve	Stephen Johnson	100
sue	Susan Jones	100

When you are finished practicing, choose **Close** from the **File** menu to exit **m4_ex1_gen.doc**. (You do not need to save your changes, because subsequent exercises do not depend on the changed document.)

Exercise 4-2: More SoDA Elements in Table Cells

You do this exercise in **m4_ex2.doc**, which should already be open on your workstation screen. Click in the document window so that you have an insertion point and the document's structure is displayed in the structure view.

You will see that **m4_ex2.doc** is similar to your finished template for exercise 4-1. The document that you generated

from that template included the preceding table of login names, user names, and groups.

Your document was well received, except that most readers had difficulty interpreting the group IDs in the third column. Thus, you would like to enhance your template to list group names instead of group IDs.

To do this, you have gotten a copy of your system's `/etc/group` file and added a `"#FIELD_DELIMITER :"` directive to it:

```
#FIELD_DELIMITER :
sysadm:1:root
devel::100:sue,bill,steve
pubs::200:judy,chris
mktg::300:jane
acct::400:bob,mike
```

The `group` file contains a record for each group, with the group name in the first field and the group ID in the third field. You can use these records to map the group IDs you get from the `passwd` file to group names that will be more understandable to your readers.

Instructions

1. Your template contains a connector to only the `passwd` file. Add a connector to the `group` file:
 - a. In the structure view, click in between the `SodaConnector` and `Title` bubbles to position the cursor as follows:



- b. Choose `Connector` from the `SoDA` menu. The `SoDA Connector` dialog box will appear. Fill it in as follows:
 - i. In the `Name` text-entry box, enter `group_file`.
 - ii. On the `Domain` option menu, choose `FileSys`.
 - iii. On the `Class` option menu, choose `File`.

- iv. In the **Arguments** area, in the **Filename** text-entry box, enter **group**.

Then click **OK** to insert the connector. It will appear in your structure view as follows:



2. Now you are ready to add a replicator to process the records in the **group** file. You will insert the replicator in the third (right-most) cell of the body row.

Remember that the body row already contains a row replicator in its left-most cell. The new replicator that you add to the right-most cell will not be a row replicator.

When you generate the document, the existing row replicator will create a row in the table for each user in the **passwd** file. Each of those rows will contain a copy of the new replicator in its third cell. That replicator will iterate through the records in the **group** file. The trick will be to define the new replicator so that it only selects the one record in the **group** file with a group ID that matches the group ID in the current password-file record.

- a. In the document window, click in the right-most cell of the body row (under the “Group” heading):

Login Name	User Name	Group
<users.Field01>	<users.Field05>	<users.Field04>


 Click here.

- b. In the structure view, click on the **Body** bubble to select it and its descendants:



(Do not select the **Cell** bubble. You want this new replicator to work just in the table cell. You are not adding another row replicator.)

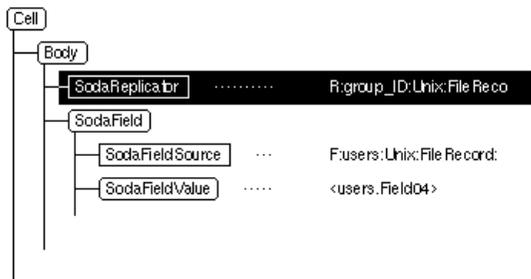
- c. Choose **Replicator** from the **SoDA** menu.

The SoDA Replicator dialog box will appear. Fill it in as follows:

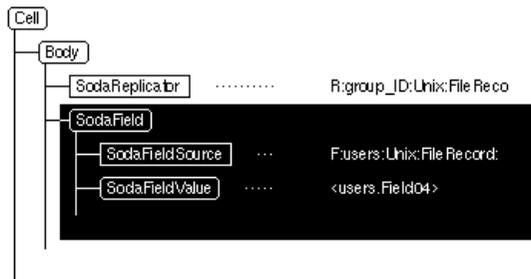
- i. In the **Name** text-entry box, enter `group_ID`.
- ii. On the **From** option menu, choose `group_file`.
- iii. In the **Select** area, choose **Records**.
- iv. On the **Where Is A** option menu, choose **FileRecord**.
- v. In the **And Where** area, to match the group IDs in the **passwd** and **group** files, specify:

`users Field04 = <Self> Field03`

Then click **OK** to insert the replicator. It will appear in your structure view as follows:

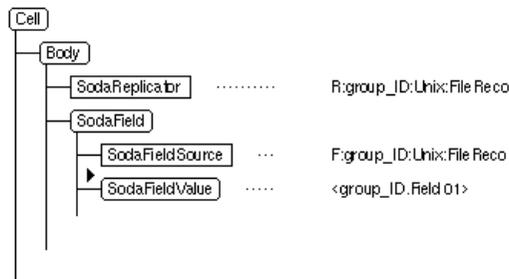


3. Finally, you need to edit the SoDA field in the right-most cell to display the group name from the **group** file instead of the group ID from the **passwd** file:
 - a. In the document window, click in the right-most cell of the body row (under the “Group” heading). (You might still be there from the previous step.)
 - b. In the structure view, click on the **SodaField** bubble to select the field:



- c. Choose **Field** (or **Edit Query**) from the **SoDA** menu. The SoDA Field dialog box will appear. Fill it in as follows:
 - i. On the **From** option menu, choose **group_ID**.
 - ii. In the **Select** area, choose **Field01**.

Then click **OK** to insert the field. It will appear in your structure view as follows:



4. That's it. You are ready to generate the document:
 - a. As a matter of habit, choose **Save** from the **File** menu to save your template so that you will have a copy of it in case you want to reuse it or want to correct problems in

it and start over. Then choose **Save As** from the **File** menu to create a copy of the template to generate. For this exercise, name it **m4_ex2_gen.doc**.

- b. Choose **Generate Document** from the **SoDA** menu.
- c. Click **OK** in the SoDA Generator dialog box that appears.
- d. Click **Continue** in the SoDA Information dialog box that appears.

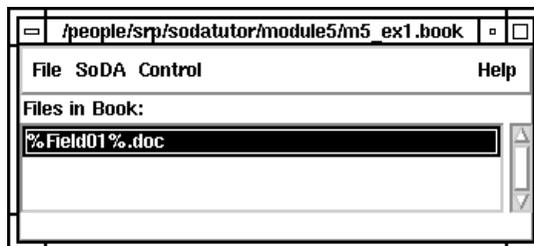
The replicated table rows will appear as follows in your document window:

Login Name	User Name	Group
bill	William Smith	devel
bob	Robert Duran	acct
chris	Chris Paulsen	pubs
jane	Jane Mathews	mktg
judy	Judy Watson	pubs
mike	Michael Pollard	acct
root	Operator	sysadm
steve	Stephen Johnson	devel
sue	Susan Jones	devel

When you are finished practicing, choose **Close** from the **File** menu to exit **m4_ex2_gen.doc**. (You do not need to save your changes, because subsequent exercises do not depend on the changed document.)

Exercise 5-1: Generating a Book

You do this exercise in **m5_ex1.book**, which should already be open on your workstation screen. Double-click on the one document it contains—**%Field01%.doc**—to open the document, too.



Then click in the document window so that you have an insertion point and the document's structure is displayed in the structure view.



You will see in the structure view that your document already contains:

- A connector to a FileSys **passwd** file (described on page 97)
- A connector to a FileSys **group** file (described on page 103)
- The book name (**m5_ex1.book**)
- A replicator to replicate this entire document for each group in the **group** file

The document replicator is the first child of the document's root element, as it must be. Attached as they are to the root element of the first document in the book, the connectors are global connectors. The document replicator depends on the connector to the **group** file, which is visible to the replicator thanks to the book name. Without the book name, the document replicator would not fall within the scope of the following connectors.

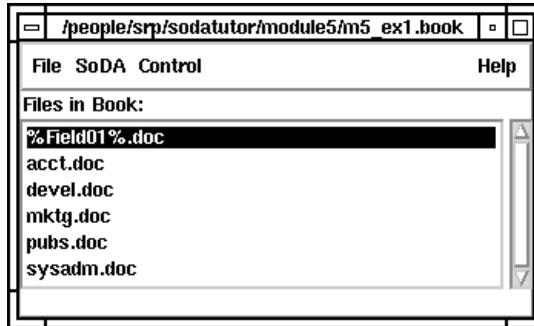
Instructions

1. Peruse the document until you are satisfied that you understand its structure. You can investigate the SoDA elements by clicking on their bubbles in the structure view to select them, choosing **Edit Query** from the **SoDA** menu, and then clicking **Cancel** to dismiss the dialog box when you are finished.
2. Generate the book:
 - a. Choose **Generate Book** from the book's **SoDA** menu (*not* **Generate Document** from the document's **SoDA** menu).
The SoDA Generator dialog box will appear.
 - b. Make sure that the **Update Documents from Source** and **Update Book from Source** options are toggled on, and that the other

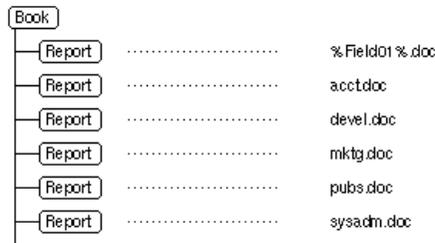
options (especially **Produce Printable Book**) are toggled off. Then click **OK**.

- c. Click **Continue** in the SoDA Information dialog box that appears.

The replicated documents will appear as follows in your book:



and the book will appear as follows in the structure view:



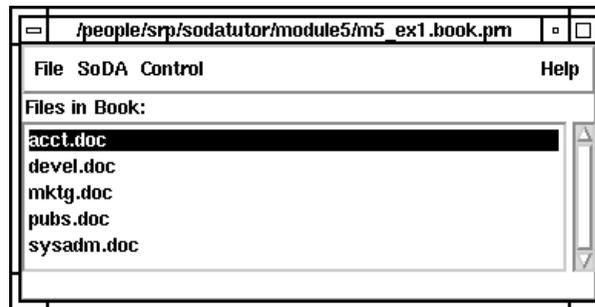
3. Save the newly generated book by pressing the **Shift** key and choosing **Save All Files in Book** from its **File** menu.
4. Browse through the files that were generated. Notice that their filenames were derived from the first field of the records in the **group** file, as specified by the template document's filename (**%Field01%.doc**).

Also notice that the first “real” document in the book (**acct.doc**) begins with section 2 on page 2, that the second “real” document in the book (**devel.doc**) begins with section 3 on page 3, and so on. That is because the template document (**%Field01%.doc**) occupies section 1 on page 1.

Now imagine that you have finished editing all the “real” documents in the book and are ready to print it. You do not want to print the template document (**%Field01%.doc**). You could just print the other documents, but then your printed copy still would not be quite right. You also want the section numbers and page numbers to be correct.

5. Generate a printable book:
 - a. Choose **Generate Book** from the book’s **SoDA** menu.
The SoDA Generator dialog box will appear.
 - b. Make sure that the **Produce Printable Book** option is toggled on. The other options can be toggled off (because in this case, nothing else needs to be updated in the book). Then click **OK**.
 - c. Click **Continue** in the SoDA Information dialog box that appears.

SoDA will generate a new, printable book. It will have the same name as the original book, but with a **.prn** extension:



6. Save the printable book by pressing the **Shift** key and choosing **Save All Files in Book** from its **File** menu.
7. You could print the printable book by choosing **Print** from its **File** menu, but you need not do that now.

When you are finished practicing, press the **Shift** key and choose **Close All Files in Book** from their **File** menus to exit **m5_ex1.book** and **m5_ex1.book.prn**. (You do not need to save your changes, because subsequent exercises do not depend on them.)



Index

A

Apex

- domain 34
- shell windows 15
- using with SoDA 15, 34

apexinit command 15

API clients

- CMVC** 65
- Element Catalog Manager** 65
- Soda** 14, 65
- Structure Generator** 65

apclients file 65

attributes

- of Apex objects 36
- of UNIX objects 38

B

bin directory 15

book masters

- see masters 42

Book Name command 45, 56

books 44

- generating 25, 26
- printing 26

Bourne shell ix

- see also **sh**

C

C shell ix

- see also **cs****h**

Check Document command 31

classes

- of Frame objects 38

CMVC API client 65

condition tags 62

- SodaMaster** 62

- SodaStipulator** 62

Conditional Text command 62

Connector command 48, 64

connectors 24, 48

Control menu 17

conventions, used in this guide x

cs**h** ix, 13, 14

D

delimiters

- in file records 37

directives, in file records 37

DISPLAY variable 13, 15

document masters

- see masters 42

documents

- editing after generation 23, 62
- regenerating 23, 32

domains

- Apex 34

E

Frame 38
Rose 34
TestMate 36

E

Edit Link command 31
editing generated documents 23, 62
Element Catalog Manager API client 65
Element menu
 Unwrap command 55, 56
Elements catalog 48
environment variables
 DISPLAY 13, 15
 PATH 14, 15
 SODA_HOME 14, 15
Exercise 1-1: Inserting Connectors 68
Exercise 1-2: Inserting Fields 72
Exercise 1-3: Inserting a Replicator 66, 78
Exercise 2-1: Generating a Document 85
Exercise 3-1: Regenerating a Document 90
Exercise 4-1: Replicating Table Rows 97
Exercise 4-2: More SoDA Elements in Table Cells 102
Exercise 5-1: Generating a Book 107
exercises, for tutorial 65, 67

F

Field command 58, 64
fields 24, 57
File menu
 Save As command 21, 32
fminit directory 65
fmtemplates directory 65, 66
Frame domain 38
FrameBuilder 65
FrameMaker 65

G

Generate Book command 26, 27
Generate command 25, 32
getting started 11

H

Help menu
 Main Menu command 16
 On Context command 17
 SoDA Tutorial command 17, 67
help, online 16
hidden text 62

I

information source domains
 see domains
instructions, for tutorial exercises 65, 67

K

keys, in file records 37
Korn shell ix
 see also **ksh**
ksh ix, 13, 15

L

links
 modifying 31
local host, displaying on 13
.login file 14

M

Main Menu command 16
Modify Links command 31
Motif Window Manager 13

see also **mwm**

mwm 13

N

NEW button 66

O

On Context command 17

online help 16

online tutorial 17, 65, 67

P

PATH variable 14, 15

prerequisites, for using this guide ix

primary links 24, 30, 31

.prn extension 27

.profile file 15

prompts, from shells x

R

regenerating documents 23, 32

remote host, running on 13

Replicator command 51, 54, 55, 56, 64

replicators 24, 50, 54, 55

Rose

domain 34

using with SoDA 34

S

Save As command 21, 32

setenv command 13, 14

setup 65

sh ix, 13, 15

shells ix

prompts x

see also C shell, Bourne shell, Korn shell

Show/Hide button 62

SoDA

API client 14, 65

bin directory 15

getting started with 11

setup 65

starting 13, 15

using with Apex 15, 34

using with Rose 34

using with TestMate 36

SoDA menu 17

Book Name command 45, 56

Check Document command 31

Connector command 48, 64

Field command 58, 64

Generate Book command 26, 27

Generate command 25, 32

Replicator command 51, 54, 55, 56, 64

Stipulator command 60, 64

Utilities submenu

Edit Link command 31

Modify Links command 31

Visit Source command 28

SoDA Tutorial command 17, 67

soda_builder command 14, 15

SODA_HOME variable 14, 15

SodaBookName element 44, 56

SodaConnector element 48, 55

SodaFieldSource element 63

SodaFieldValue element 63

SodaMaster condition tag 62

SodaReplicator element 44, 56

SodaStipulator condition tag 62

SodaStipulator element 56

source domains

see domains

Special menu

T

Conditional Text command 62
starting SoDA 13, 15
Stipulator command 60, 64
stipulators 24, 59
Structure Generator API client 65

T

tables 54
Templates/SoDA directory 66
TestMate
 domain 36
 using with SoDA 36
tutorial, online 17, 65, 67

U

Unwrap command 55, 56

V

Visit Source command 28

W

window manager 13

X

X server 13
X Window System 13
xhost command 14, 15
.xinitrc file 15
xterm windows 13



Reader's Comments

Please return this form to:
SoDA Development
Rational Software Corporation
18880 Homestead Road
Cupertino, CA 95014

Note: You can also submit documentation comments by sending electronic mail to **documentation@rational.com**. If you send mail, please indicate the book name, revision number, and page number.

Did you find this book understandable, usable, and well organized?
Please comment and list any suggestions for improvement.

Did you find all the information you needed? If not, please describe what was missing.

If you found errors in this book, please specify the errors and page numbers. If you prefer, attach photocopies with the errors marked.

Please list any additions or changes you would like to see in the index.

OPTIONAL: Name _____ Date _____
Company _____
Address _____
City _____ State _____ ZIP _____
E-mail _____ Telephone _____

