

SPECWARE

Graphical User Interface

Preliminary Design

December 10, 1998

I

Copyright © 1998 by Kestrel Institute and Kestrel Development Corporation, Palo Alto. All rights reserved.

SPECWARE is a registered trademark of Kestrel Development Corporation.

The use herein of the above trademarks does not create any right, title, or interest in or to the trademarks.

Table of Contents

Table of Contents	3
1 Basic Concepts	13
Units and Classes	13
Projects and Libraries	13
Projects	14
Libraries	14
Hierarchy of Items	14
Accessibility of Units	15
Exported Items	15
Subscription	15
Summary of Accessibility	16
Defining Terms	16
Project Interchange Files	16
UI-Specific Classes	17
Views	17
Text View	17
Picture View	17
Information View	17
Present-Value View	18
Table Views	18
Servers and Registries	18
Universe of Available Items	18
Administrative Information	19

Working with SPECWARE	19
Getting Started	19
Browsing and Visiting	20
Modifying the Current Project	21
Modifying Items in the Current Project	21
Saving Changes	21
Undo History	22
Commit and Evaluation	22
Administration Functions	22
 2 General Interface Conventions	 25
Windows and Dialog Boxes	25
Active Window Area	25
Menus	26
Flat and Hierarchical Lists	26
Flat Lists	26
Hierarchical Lists	26
Selecting	28
Selecting in Text Areas	28
Selecting in Non-Text Areas	28
Scrolling	29
Drag and Drop	30
Editing Text Elements	30
Resizing Window Areas	31
Visiting	31
Visiting Undefined Items	31
Visiting Anonymous Units	32
 3 Project Window	 33
Opening and Closing a Project	34
Opening a Project	34
Closing a Project	35
Title Bar	35
Version Tree Area	35
Version Tree	35
Creating a New Version	36
Mouse-Controlled Actions	36
Keyboard-Controlled Actions	36
Selecting	37

Reverting	37
Moving Nodes and Branches	37
Version Description Area	38
Status Display	39
Menu Bar	39
File Menu	40
Edit Menu	42
Search Menu	43
Version Menu	44
Outline Menu	45
Windows Menu	46
Help Menu	47
On-Line Help	47
Menu entries	47
Tool Bar	47
Command Buttons	48
Editing Buttons	48
Outline Buttons	48
Zoom Controls	49
Magnification Pulldown	49
 4 Browsers	 51
Opening a Browser	52
Modes of Operation	52
Title Bar	52
Index	53
Mouse-Controlled Operations	53
Keyboard-Controlled Operations	54
Visit Area	54
Status Display	54
Window Settings	55
Visit History	56
Peeling Off a Browser	57
Menu Bar	57
File Menu	58
Edit Menu	60
Search Menu	61
Visit Menu	62
View Menu	64
Window Options	64

Index Options	64
View Options	65
Help Menu	65
Tool Bar	65
Command Buttons	66
Editing Buttons	66
Navigation Buttons	66
Zoom Controls	67
Magnification Pulldown	67
5 Class-Specific Controls and Behavior in Browsers	69
Project and Library Classes	69
Table Views	70
Target Project	70
Cut-and-Paste Operations	71
Drag-and-Drop Operations	71
Text View	72
Information View	72
File Menu	73
View Menu	75
Column-Display Options	75
Table-Display Options	76
Order Menu	77
Project Menu	78
Klang Spec	81
Text View	81
Picture View	81
Colimit Specs	81
Translation Specs	82
Import Specs	82
Other Specs	83
Defining a New Spec	83
Information View	83
Present-Value View	84
Klang Spec Morphism	84
Mapping Table View	84
Table Header	85
Table Body	86
Text View	86
Picture View	86
Composition Morphisms	87
Other Morphisms	87

Defining a New Morphism	87
Information View	87
Present-Value View	88
Edit Menu	88
View Menu	89
Order Menu	89
Possible Mappings List	90
Diagram Classes	90
Text View	90
Picture View	90
Information View	91
Present-Value View	91
File Class	91

6 View-Specific Controls and Behavior in Browsers 93

Text View	93
Text-Editing Capabilities	93
Keyboard Equivalents	94
Syntax-Based Editing	94
Text Menu	94
Picture View	95
Drawing Conventions in Graphs	96
Layout	96
Labels	96
Syntax for Labels	97
SPECWARE-Supplied Label IDs	97
Viewing Labels	98
Modifying Labels in the Graph	98
Selection Operations	98
Cut and Paste Operations	98
Cutting	98
Pasting Nodes	98
Pasting Arcs	99
Pasting Graphs	99
Replacing Subgraphs	99
Drag-and-Drop Operations	100
Moving Parts of the Graph	100
Copying Parts of the Graph	101
Adding to the Graph	101
Making Arcs Curve	101
Search Menu	102
View Menu	103
View-Specific Index Option	103

Label Options	103
Graph Menu	104
Adding Nodes and Arcs	104
Expanding and Contracting Subgraphs	105
Showing and Hiding Details	105
Controlling Layout	105
Label Field	105
Graph Tools Button	106
Information View	106
Text-Editing Capabilities	107
Outline Editing	107
Selected Paragraphs	108
Outline Menu	108
Outline Buttons	109
Present-Value View	109
Table Views	109
Table-Specific Controls	110
Table Rows and Columns	110
Hierarchical Tables	111
Selecting	111
Mouse-Controlled Actions	111
Keyboard-Controlled Actions	112
 7 Auxiliary Windows	 113
Command Window	113
Opening the Command Window	114
Controls	114
Command Field	114
Command Log	114
Command Buttons	114
Dependency Windows	115
Opening a Dependency Window	116
Controls	116
Item Path Name	116
Checkboxes and Lists	116
Radio Buttons	116
Show Dependencies Button	117
Find/Replace	117
Opening the Find/Replace Window	117
Search Area	117
Controls	118
Describing the Search and the Replacement Text	118
Command Buttons	118

Find/Replace in Project	120
Opening the Find in Project Window	120
Target Browser	120
Controls	121
Search String	121
Items to be Searched	121
Replacement Text	122
Command Buttons	122
Graph Tools Palette	123
Opening the Graph Tools Palette	123
Changing the Nodes and Arcs	123
Common Functionality	124
NN: New Node	124
NA: New Arc	124
NC: New Colimit	125
SC: Sequential Composition of Arcs	125
MN: Merge Nodes	126
Expanding and Contracting Subgraphs	126
X: Expand Subgraph	126
C: Contract Subgraph	126
Showing and Hiding Details	126
S: Show Details	127
H: Hide Details	128
Controlling Layout	128
Help Window	128
Opening the Help Window	129
Search for Items	129
Opening the Search for Items Window	129
Controls	130
Filtering by Hierarchy	130
Filtering by Class	131
Filtering by Name	131
Filtering by Status	131
Filtering by Dependency Network	131
Search Button	132
Search for Morphisms	132
Opening a Search for Morphisms Window	133
Controls	133
Unit Path Name	133
Source and Target	133
Morphism List	134
Show Morphisms	134
Selection-List Window	135
Opening the Selection-List Window	135
Controls	136

List of Entries	136
Hierarchical List Checkbox	136
Use As Browser Index	136
8 Dialog Boxes	137
Create Library	137
Opening the Create Library Dialog Box	138
Controls	138
Library Name Field	138
Naming Sublibraries	138
Creating Libraries from Subscribed-To Projects	139
Command Buttons	140
Go To Line	140
New Directory	141
New Version	141
Open File	142
Opening the Open File Dialog Box	142
Controls	142
Directory Pulldown	143
Contents List	143
Selected Directory	143
File Name Field	143
Command Buttons	143
Page Setup	144
Preferences	144
Opening the Preferences Dialog Box	144
Project Window Options	145
Browser Options	145
Project and Library Options	146
Picture-View Options	146
Text Format Options	147
Color Use	148
Keyboard Shortcuts	148
Interface Options	148
SPECWARE Options	149
Prover Options	150
Printing Options	150
Miscellaneous	150
Print	151
Opening the Print Dialog Box	151
Rename	151

Replace Selection with Graph	152
Opening the Replace Selection with Graph Dialog Box	153
Operation	153
Controls	154
Graph Display	154
Command Buttons	154
Select Directory	155
Opening the Select Directory Dialog Box	155
Controls	155
Directory Pulldown	155
Subdirectory List	156
Selected Directory	156
Command Buttons	156
Select Project	157
Opening the Select Project Dialog Box	157
Controls	157
Container Pulldown	158
Contents List	158
Selected Container	158
Open in Read-Only Mode	158
Command Buttons	159
Write to File	159
Opening the Write to File Dialog Box	159
Controls	160
Directory Pulldown	160
Contents List	160
Selected Directory	161
File Name Field	161
View Pulldown	161
Format Pulldown	161
Command Buttons	161
Zoom Options	162

Appendix A Keyboard Shortcuts 163

Appendix B Displaying and Editing Labels in Picture View . . . 167

Displaying Labels	167
Editing Labels	168

Appendix C Variance with *Design Decisions* 173

1 Basic Concepts

This chapter introduces the basic terminology and concepts that are fundamental to the proposed graphical user interface (GUI) for Kestrel Institute's SPECWARE product.

Units and Classes

A *unit* is any SPECWARE entity that has a name and a defining term. Defining terms can be expressed textually in a language called Klang (Kestrel LANGuage). The Klang representation of a defining term is called a *Klang definition*.

Klang includes:

- ❑ The current unnamed general language used in SPECWARE *.re files.
- ❑ The current Slang, which (according to some reports) includes only definitions of specs.
- ❑ Additional statements that define units for which no syntax currently exists.

Textual representation of one unit's defining term can refer to another unit by name.

When a unit's term is *evaluated*, SPECWARE creates its corresponding *present value*.

Every unit belongs to some *class*. Current classes include specs, spec morphisms, interpretations, interpretation morphisms, spec-morphism diagrams.

Some classes have a picture representation as well as a textual one. For example, spec-morphism diagrams (and all other classes of diagrams) have a picture representation.

Some classes may have additional representations of defining terms.

Every unit in a project is uniquely identified by its class plus name. A unit's *identifier* has the form *ClassName UnitName*.

Projects and Libraries

Units are grouped hierarchically into projects and libraries.

Projects

A *project* is a collection of units that are under development by one or more persons. The units within a project may be organized hierarchically. That is, any project can contain other projects as well as units. A project is a non-leaf node in the hierarchy; a unit is a leaf node. While a project is under development, its *development history* consists of a tree of the various versions of the project that the user has saved.

When a project is first created, it has a single version (with the default name Initial); this version has no parent in the development hierarchy. At a stable point in the development process, or when the user wants to explore alternative approaches, the user saves a new version, giving it a name and (optionally) a description. The user can revert to any existing version and continue development using that version as a starting point.

At any point in the development of a project, exactly one version is *active*; the user makes changes to the active version. When the user switches to a different version, the version that was active becomes *dormant*.

To create a new version, the user gives a name and (optionally) a description of the version. A copy of the active version is saved with the indicated name. The new version is added to the project's development history between the existing parent (if any) of the active version and the active version itself. Unless the active version has already been named by the user, its name is changed to the form *SavedVersion.n*.

Any saved version is *frozen*; that is, it cannot be modified. The user can revert to any frozen version. After doing so, if the user tries to make any change, a copy of the frozen version is made the new active version and the change is made to the active version. The new active version is given the default name of the form *FrozenVersion.n*.

If the user reverts to an existing dormant version, that version (and not a copy of it) becomes the active version.

Libraries

A *library* is a hierarchical collection of read-only units that can be shared by projects. Once developers have finished working on a project and have tested it, they can make it available for use in other projects by creating a library from the selected version of the project.

- ☐ A sublibrary is created from each of subproject in the project's hierarchy.
- ☐ A read-only unit is created from each unit in the project's hierarchy.

A library can be thought of as a read-only project that has no development history.

Hierarchy of Items

Units, projects, and libraries are the entities that a SPECWARE user creates and works with. The term *item* refers to any project, library, or unit.

The universe of all items forms a hierarchy. Any unit belongs to exactly one project or library; the unit is said to be *defined in* the project or library to which it belongs.

A user can *duplicate* units, libraries, subprojects, or entire projects, either to create a new top-level project or to add the duplicates to an existing project. Duplicating a project or library makes a deep copy; that is, the resulting project contains a copy of each unit in the original and a deep copy of each subproject/sublibrary in the original.

Accessibility of Units

The defining term of one unit may reference any other unit that is *accessible* to it. Any unit in a project or library is accessible to all other units defined within the same project or library. Accessibility can be extended beyond the containing project or library by exportation and subscription.

Exported Items

A project or library may *export* some or all of its items to extend their accessibility. The exported units constitute the “API” of the project or library. Any units that are not exported are *internal* to the project or library in which they are defined. (Typically, exported units are defined in terms of internal units.)

If the user indicates that a morphism is exported, its source and target are automatically exported. If the user exports a diagram, all its morphisms are automatically exported.

Units exported by a subproject are accessible to units in the containing project; a project can export a subproject to re-export the subproject’s exported (or re-exported) items. Similarly, units exported by a sublibrary are accessible to units in the containing library; a library can export a sublibrary to re-export the sublibrary’s exported (or re-exported) items.

Subscription

A project can *subscribe to* a library to gain access to the items that the library exports. The library must be one of the following:

- ☐ A top-level library in its registry
- ☐ A sublibrary that is exported or re-exported by a top-level containing library

A subproject can also subscribe to a sibling subproject to gain access to the items that the sibling project exports. (Two subprojects are siblings if they are defined in the same A containing project.) Sibling subscription is not transitive; each subproject must explicitly subscribe to every sibling whose units it wants to access.

A project or library can export a subscribed-to library or project to re-export the exported (or re-exported) units in the subscribed-to container.

When the user creates a project from a library, all export and subscription relationships that held within the project hierarchy are duplicated in the library hierarchy. If project A subscribes to a sibling project B, when the user creates a library from A, a library is also created from B and the former library subscribes to the latter.

✂ Need to get feedback from Lambert on whether a library is created automatically for a subscribed-to sibling project.

Summary of Accessibility

The accessibility of units is defined by the following rules.

- ❑ Unit A is accessible to unit B if A is accessible to the project or library that contains B.
- ❑ All units defined in a project or library are accessible to that project or library.
- ❑ A project P that is a subproject of project Q can export a unit to make it accessible to:
 - ❑ The project Q.
 - ❑ Any project in Q that subscribes to P. (The subscribing project is a sibling of project P.)
- ❑ A library L at the top-level of its registry can export a unit to make it accessible to any project or library that subscribes to L.
- ❑ A library L that is an internal sublibrary of library M can export a unit to make it accessible to the library M.
- ❑ A library L that is an exported sublibrary of library M can export a unit to make it accessible to:
 - ❑ The library M.
 - ❑ Any project or library that subscribes to L.
- ❑ A project that exports a subproject or subscribed-to container makes all exported and re-exported items of that container accessible to the next higher level project or to a subscribing sibling project. (While this has no immediate meaning for a top-level project, it obtains one if at some later stage a library is created from that project.)
- ❑ A library that exports a sublibrary or subscribed-to library makes all exported and re-exported units of the latter library accessible to the next higher level library or to a subscribing project or library.

Defining Terms

To provide a uniform interface for dealing with entities at all levels of the hierarchy, projects and libraries are considered to have defining terms.

The Klang syntax for a project definition would look something like the following:

```
project MyProject is
  projects A, B, C           % list pathnames of subprojects
  subscribe to library1, ... % list pathnames of libraries
  export spec Dollar-Account, ... % list class and name of exported items
  specs Account, Amount, ... % Class and names of units of that class
  morphisms ...             % Class and names of units of that class
  ...
endProject
```

Project Interchange Files

Specware will support ASCII *project interchange files* that enable users to move projects easily between platforms, send projects as email attachments, and so on.

A project interchange file might contain the Klang definition of the project and the Klang definitions of each subproject and unit declared in the project definition.

-
- ✎ The project interchange file also needs to record the tree of versions. The form of this information is TBD.
-

UI-Specific Classes

To provide a uniform interface for dealing with entities at all levels of the hierarchy as well as viewing arbitrary text files, the notion of *class* is extended to include the following classes:

Project	The class of all projects (at any level of the hierarchy)
Library	The class of all libraries (at any level of the hierarchy)
File	The class of all file-system files

Views

The defining term of an item can be displayed in a number of different *views*, depending on its class.

Text View

All classes have a text view. The text view of a SPECWARE item is its the Klang definition. The text view of a file is the ASCII content of the file.

Picture View

All classes of units, but not Project, Library, or File, have a picture view that displays a two-dimensional graph representing a diagram in some category. We use the term *diagram* to refer to the category-theoretical concept and *graph* to refer to its pictorial representation.

Information View

All SPECWARE classes also have an information view, but the File class does not. The information view does not show the item's defining term; instead, it shows read-only status information about the item and editable user notes about the item.

-
- ✎ We could add an information view for files that gives full path name, modification date, and size, but this information is readily available outside of SPECWARE, so this feature is probably not necessary.
-

Present-Value View

All classes of units have a (read-only) present-value view; the Project, Library, and File classes do not. A unit's present-value view is the full Klang text of its defining term, expanded to replace references to other units with their defining terms. The present-value view is analogous to Show Text (long) in the current interface.

If a unit is invalid, its present-value view contains any progress messages and warnings from the evaluation process and a reason why evaluation failed.

Table Views

Any class for which a tabular presentation is useful has one or more table views. For example, a project or library can be viewed as a table of the units and subprojects or sublibraries it contains.

In a table view, each row represents an "element" of the item; each column contains a class-specific attribute (particular piece of information) for each element of the item.

When the elements can be organized into groups (by category or container), a *hierarchical table* view is possible. In a hierarchical table, the rows are organized to reflect the group to which the associated elements belong.

Servers and Registries

Every site where SPECWARE is installed, can have:

- ☐ A *public registry* containing:
 - ☐ The public projects installed at the site.
 - ☐ The public libraries installed at the site.
- ☐ A *private registry* for each local SPECWARE user, containing projects created by that user (any not yet made public).

Servers at other sites provide access to the public registries at those sites.

Universe of Available Items

The universe of all units is a hierarchy.

- ☐ The universe contains zero or more registries.
- ☐ Each registry contains zero or more projects and/or libraries.
- ☐ Each project contains zero or more subprojects and/or units.
- ☐ Each library contains one or more sublibraries and/or units.

The term *container* refers to any registry, project, or library.

This document uses the following icons for container items in the hierarchy:

 Registry

 Project

 Library

➤ In general the context should make it clear whether a project or library is at the top level of its registry or is a subproject or sublibrary. If we find contexts in which we want to avoid ambiguity, we could use different symbols, for example, fill the project or library icon if it's at the top level and leave the icon as an outline for a subproject or sublibrary.

An item's *pathname* gives its location in the universe. The path name has the form:

RegistryName : { *ProjectOrLibraryName* / }+ *UnitName*

Administrative Information

SPECWARE maintains and uses the following information at each site:

- ☐ The mapping between a unit pathname and the physical location of the unit's Klang definition (which may be in a file or a database or ...).
- ☐ The mapping between a server name and the URL, or IP address, or whatever by which the UI can contact the server.
- ☐ (Possibly) Which groups of users have read/write access to each project.
- ☐ (Possibly) Which groups of users have read-only access to each library and project.
- ☐ (Possibly) The list of authorized users, the groups to which they belong, and their access privileges.

This document makes no assumptions about how or where this administrative information is stored.

Working with SPECWARE

Most interaction with SPECWARE occurs in the project window and in browsers.

Getting Started

When the user starts SPECWARE, the *project window* opens. The project window allows the user to:

- ☐ Create a new project, which can be:
 - ☐ An empty project.
 - ☐ A duplicate (deep copy) of an existing project or library. A duplicate of a library is a project whose contents can be modified.
 - ☐ A project created from a project interchange file.

A user who creates a project automatically has write access to the new project. The new project is added to the user's private registry and is automatically made the current project. Unless and until the project is installed at the side (see *Administration Functions* on page 22), no other user can access this project.

- ☐ Open an existing project for modification:
 - ☐ A project in the user's private registry
 - ☐ A project at the local site to which the user has write access
- ☐ Open in read-only mode any project at any site to which the user has read access.

Only one project may be opened at any given time. If the open project was opened for modification, it is called the *current project*.

-
- The **only** project that a user can modify is the current project.
-

The project window displays the open project's version tree and allows the user to view descriptions of each version. If the project was opened for modification, the user can modify its development history as follows:

- ☐ Create new versions of the current project.
- ☐ Add or modify descriptive information about a version of the current project.
- ☐ Switch between existing versions of the current project, suspending development on one version and continuing development on another.
- ☐ Create a library in the user's own private registry from the active version of the current project.
- ☐ Delete versions of the current project.
- ☐ Freeze versions of the current project, disallowing future modifications to those versions.
- ☐ Rearrange versions of the current project.

Browsing and Visiting

The user can open a *browser* from the project menu. As the name implies, a browser lets the user browse his or her own private registry, the public registry at the local site, and the public registries at any remote site for which a server is available.

The browser's *index*, allows the user to peruse all accessible items. The user can set the index to show:

- ☐ The entire universe hierarchy
- ☐ The entire hierarchy of the current project
- ☐ The items that satisfy some filtering criteria
- ☐ Files in the file system

The index can be displayed its elements either organized hierarchically or as a flat alphabetical list.

The user can select items in the index to *visit* in the browser. Visiting an item displays its defining term in the browser. A browser allows the user to see the visited item in any of several available *views*. Each browser maintains a *visit history*; the user can easily go back to any item in the visit history.

The user can open any number of browsers at the same time to:

- ☐ Visit different items.
- ☐ Visit the same item in different views.
- ☐ Display different parts of some view of an item.

Modifying the Current Project

When the user visits the current project, the browser allows the user to modify the project in the following ways:

- ☐ Copy a unit from any available project or library (at any site) into the current project.
- ☐ Subscribe to any available library (at any site), making its exported units available to the current project.
- ☐ Add new items to the current project. The new items may be:
 - ☐ Empty
 - ☐ Duplicates of existing units, projects, or libraries
 - ☐ Items whose Klang definitions were read from a file (either a complete project interchange file or a file with Klang definitions of any number of units and projects).
- ☐ Move items around within the hierarchy of the current project.
- ☐ Indicate which units and subprojects the current project exports.
- ☐ Indicate which units and subprojects any subproject of the current project exports.

Modifying Items in the Current Project

When the user visits a subproject or unit of the current project, the browser allows the user to modify that item by editing its defining term in any available read/write view. Most views support standard editing operations like cut, copy, and paste as well as editing operations that are specialized to their presentation of the item's defining term. Some non-text views support editing operations that are specific to the class of the object being visited.

Changes made in one browser are reflected immediately in all other visible browsers visiting the same item.

Saving Changes

All changes to SPECWARE items are saved automatically to persistent storage; no separate user action is required. Changes are written out frequently so that few if any changes are lost in the event of a hardware or software failure.

Undo History

Any action that changes an item's defining term can be undone. The Undo command can be repeated to work back through a series of changes. The Redo command undoes the Undo command. It can be repeated to work forward after a series of undo's.

The undo history is tied to the item (unit, project, or file) being modified, not to the particular browser in which modifications occurred. Particular changes may have been made in different browsers visiting the same item, however, any change can be undone in any of the browsers.

Commit and Evaluation

Committing an item signals that it is in a stable state. Every SPECWARE item has a *commit history* of its committed states. A *modified* item has been changed since it was last committed.

Committing a unit enables evaluation from the current state. A unit's present value is defined as the result of evaluating the unit's **last committed state**.

Committing a unit *with dependencies* not only commits that unit, but also commits all units it references and, recursively, all units that those units reference.

Committing a project triggers the Commit command for every modified item it contains. Similarly, committing a project with dependencies triggers the Commit with Dependencies command for every modified item in the project.

Every SPECWARE item has a *commit history* of its committed states.

- ☐ If a recent series of changes don't work out, the user can *roll back* to the previous committed state of the item being visited.
- ☐ A user who rolls back too far in the commit history can *roll forward* to the item's next committed state.

Administration Functions

Each site where SPECWARE is installed has a SPECWARE *administrator* (or a collection of people, all of whom have administrator privileges). In addition to all user functions, the administrator can:

- ☐ Browse the public registry and all private user registries at the local site, and the public registries at any remote site for which a server is available.
- ☐ Install a project into the site's public registry, possibly limiting read and/or write access to a certain group of users. The installed project can be:
 - ☐ An empty project (to be developed collaboratively).
 - ☐ A copy of an existing project (typically from some user's private registry).
 - ☐ A project created from a project interchange file.
- ☐ Install a library into the site's public registry, possibly limiting read access to a certain group of users. The installed library can be:
 - ☐ An existing library in some user's private registry.

- ☐ A selected version of an existing project (anywhere in the hierarchy) from which the administrator creates a library.

Although a library is read only, it can be updated as follows:

- ☐ A user creates a new project by duplicating the library.
- ☐ The user (or some group of users) makes modifications and continues development on the project.
- ☐ A user creates a library from a selected version of the project.
- ☐ If the new library is to be public, the administrator installs it in the public registry.
- ☐ After all users have been able to adapt (if necessary) their projects to the updated library, the administrator removes the original library.
- ☐ Add a new server to the list of servers accessible from the local site.

-
- Depending on how protection and locking are handled, an administrator might also add new users and specify which projects they can access (or which groups they belong to), analogous to adding users and their access privileges to a database.
-

This document does not describe the interface for administrator functions. A separate administration program could be developed, or the standard UI could provide some way to enable administration mode for authorized users.

2 General Interface Conventions

This chapter describes conventions that apply to the various windows and dialog boxes in the interface.

Windows and Dialog Boxes

When the user issues a command that opens a window or dialog box, the window or dialog box appears in a default location on the screen. The default location is determined based on the location of the active window from which the user issued the command and available empty screen real estate. An auxiliary window or dialog box that was opened earlier in the same session, is placed in the same location where it was last closed.

All windows have a title bar and a close box. All dialog boxes have a title bar. The user can move a window or dialog box by pointing to its title bar, depressing the mouse button, and moving the mouse. A shadow outline moves with the mouse pointer. The window or dialog box is moved to the shadow's position when the user releases the mouse button.

A window stays open until the user clicks its close box. A dialog box stays open until the user clicks one of the command buttons that terminate the dialog.

Most windows can be resized and minimized. Standard controls for the chosen windowing system will be used to resize and minimize windows.

Active Window Area

The project window and browsers both have different window areas in which the user can interact. When the user clicks in a window area, in the border of the area, or in one of the area's scroll bars, that area becomes the *active area* of the window. The border of the active area is highlighted to give the user a visual indication of which area is active. The simple guideline is that the active area is the one used most recently.

- ❑ Keyboard input goes to the active area unless the only window area that can accept keyboard input is a different area; in the latter case, any keyboard input activates the area that accepts keyboard input.
- ❑ Some commands (typically available from menu items and their keyboard shortcuts) affect the active window.

Menus

Following standard GUI conventions:

- ☐ A menu entry that ends with ellipses (...) brings up a dialog box or auxiliary window.
- ☐ A menu entry followed by a right pointer (➤) has a submenu.
- ☐ A menu entry's keyboard equivalent, if any, is shown at the right margin of the line for that entry.
- ☐ A toggle menu entry is preceded by a checkmark (✓) to indicate that the option is currently enabled.
- ☐ A menu entry that represents one of several alternative options is preceded by a checkmark (✓) to indicate it is the currently chosen alternative.
- ☐ In a major window, the first two menus are named File and Edit. In addition to any SPECWARE-specific menu entries, those menus contain the standard entries for File and Edit menus.

Flat and Hierarchical Lists

In various windows, lists of SPECWARE items or files can be presented in either flat or hierarchical form. The items or files to be displayed in the list are called the *content entries*.

Flat Lists

A *flat list* contains the content entries in an alphabetical order. Each entry in a flat list of SPECWARE items consists of the item class, name, and full path name. Each entry in a flat list of file-system files consists of the file name and its directory path name.

Hierarchical Lists

A *hierarchical list* includes entries for any container entries necessary to show the location of the content entries. Container entries in a list of SPECWARE items are registries, projects, and libraries. Container entries in a list of files are directories.

-
- Any container entry that is not also a content entry is greyed out.
-

Hierarchical relationships are shown by indentation; entries for contained items are indented under the entries for their containing item. Each entry for a container item consists of an icon indicating the type of container and the item's name; each entry for a unit consists of the class name followed by the unit name; each entry for a file consists of the file name.

Indented under the container entry for a project or a library are:

- ☐ A noncontainer entry for each unit defined in the container.
- ☐ A container entry for each subproject of the project or sublibrary of the library.

- ❑ A noncontainer entry for each library or project to which the container project or library subscribes. The name of the subscribed-to item is in italics to indicate that it is not defined in the container project or library.

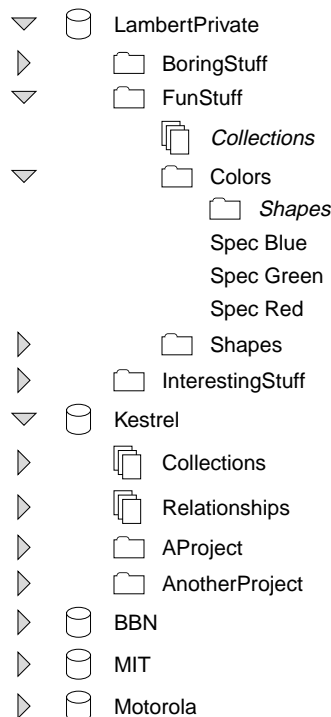
A triangle appears at the left margin of any entry for a container entry. The triangle behaves like the triangle on a line in an Acrobat Reader index:

- ❑ The triangle points right when the container's items are not shown; it points down when they are shown.
- ❑ The user clicks the triangle to toggle between showing and hiding the items in a container.
- ❑ Showing items in a container adds entries for those items to the list, directly below the container entry. Hiding items in a container removes their entries from the list, but remembers the state of the triangle indicators for any hidden containers.
- ❑ Triangles are active even for greyed out container items (that is, those that are not part of the list content, but are added to the list to show location in the hierarchy).

In most cases, entries at a given level in the hierarchy are ordered alphabetically. The only exception is in a list of SPECWARE items that spans the entire universe hierarchy. In such a list, items at all levels but the top level appear alphabetical from top to bottom. At the top level, registries appear in the following order from top to bottom:

1. The user's private registry
2. The public registry at the local site
3. Public registries at foreign sites with available serves, in alphabetical order

For example, a browser index is a list; one index option includes the list of all items in the universe. To user Lambert, the hierarchical form of the list might look as follows:



Selecting

Selecting from controls like menus and list follows standard GUI conventions. Slightly different selecting conventions are used in text areas (such as the visit area of a browser in text view or information view or the version description of the project window) and non-text areas (such as tables and graphs).

Selecting in Text Areas

Text areas support only one contiguous selection.

To select in a text area:

- ☐ Click to place the insertion bar, then depress the mouse button and drag across the text you want to select. Selection is extended a character at a time. When you release the mouse button, selection ends.
- ☐ Double-click to select a word. You can then extend the selection a word at a time by depressing the mouse button and dragging across the words you want to select.
- ☐ Triple-click to select a line of text. You can then extend the selection a line at a time by depressing the mouse button and dragging across the lines you want to select.
- ☐ Shift-click to extend the selection from the first selected character, word, or line to the character, word, or line where you click.

✂ Ideally clicking more times than three would be treated like a triple-click.

Selecting in Non-Text Areas

In a non-text area you can select non-text elements. The defining of what constitutes an “element” depends on the type of area.

- ☐ In a list, the element is a list entry (line in the list).
- ☐ In tables (like the table views of a project), an element can be either the entire row or a cell in the table.
- ☐ In a graph, the element is a node or an arc.

In many non-text areas, you can also select text within a text element. To do so, you enter text-edit mode on the text element as described in *Editing Text Elements* on page 30, then follow normal text-selection conventions.

In a non-text area:

- ☐ Click an unselected element to select it.
In a table the user clicks a selectable cell to select it and the table border next to a cell to select the row containing that cell. (It should be possible to select a column in an analogous manner; selecting a column is not meaningful in any of the current table views.)
- ☐ Shift-click an unselected element to add that element to the selection along with all elements between it and the originally selected element. The definition of “between” depends on the type of area. In a table, all cells or rows between the initially selected

cell or row and the clicked cell or row are added to the selection. In a graph, all nodes and arcs on any path from the initial selection to the clicked element are added to the selection.

- ☐ Shift-click a selected element to remove it from the selection.
- ☐ Ctrl-click an unselected element to add it to the selection without adding elements between it and the initial selection.
- ☐ Press the mouse button at one corner of an area containing elements you want to select; keep the button depressed as you move the mouse pointer to identify that area. As you move the mouse, you will see a dotted rectangle with one corner at the location where you pressed the mouse button and the diagonally opposite corner at the mouse pointer. As you move the borders of the rectangle, any elements within the borders are highlighted and any elements outside the borders are unhighlighted. When the rectangle encloses the elements you want to select, release the mouse pointer. This action selects all elements within the area you outlined.
- ☐ Depress the Shift key while you outline an area containing elements you want to select; this action:
 - ☐ Adds to the selection any unselected elements within the area you outlined and all elements between newly selected elements and the originally selected element.
 - ☐ Deselects any currently selected elements within the area you outlined

As you move the borders of the rectangle, any elements that would be selected by the operation are highlighted and any that would be deselected have their highlight removed.
- ☐ Depress the Ctrl key while you outline an area containing elements you want to select; this action adds any unselected elements within the area you outlined to the selection. As you move the borders of the rectangle, any elements within the borders are highlighted and any elements outside the borders are unhighlighted.

The user may Shift-select or Ctrl-select any number of times; each such action adds to the current selection. In the case of multiple Shift-select actions, each one adds elements between the selected element(s) and the first element(s) to be selected. For example, if a user selects a node in a graph then performs multiple Shift-select actions, each Shift select selects elements in a path between the newly selected element(s) and the originally selected element(s), but not between the newly selected element(s) and all currently selected elements.

Scrolling

Window areas and lists that may not be large enough to show their entire content have adjacent scroll bars that allow the user to scroll through the content.

In the active window area of the project window or a browser, the standard keyboard shortcuts for scrolling are supported:

Page up	Scroll backward
Page down	Scroll forward
Home	Scroll to beginning of content
End	Scroll to end of content

Drag and Drop

The exact details of a drag-and-drop operation vary somewhat depending on the type of elements being dragged and the type of area in which they are to be dropped. In general, the user performs the operation with the following steps:

- ❑ Select the element(s) to be dragged.
- ❑ Point to the selection and depress the mouse button while moving the mouse. An outline of the selection follows the mouse pointer, indicating that the selection is being dragged.

A user preference allows the user to signal the start of a drag operation by pressing a “drag prefix” and then drag without the mouse button depressed.
- ❑ Drag the selection to the desired location. When the mouse pointer is over an element on which the selection can be dropped, that element is highlighted to indicate that it is the current target of the drop; when the mouse pointer leaves the element, the highlight disappears.
- ❑ Release the mouse button to drop the item (or click the mouse button if using a drag prefix).

Dropping outside the target area of the drop cancels the operation. When dragging within the same area of a window, dropping on or close to the original location also cancels the operation. The dropped item is “snapped back” to its original location.

Every scrollable window area and control support *autoscrolling* if the user drags past or just inside its border. Thus, moving the mouse pointer just inside or past the right edge of the window area causes the window area to autoscroll until the pointer moves back inside or reaches the edge of the canvas.

In the case of a window area whose canvas can be extended in any direction (for example, the visit area of a browser in picture view), the user can autoscroll in any direction; autoscrolling stops just before the visible section of the canvas would become all blank.

Editing Text Elements

Most non-text views include some text. For example, the nodes or arcs of a graph may have text labels; cells in a table may contain text. Some non-text views allow editing on some or all text that they contain. In this section, the term *text element* means any text within a non-text view.

You can enter text-edit mode on an editable text element in either of the following ways:

- ❑ Double-click the text element.
- ❑ If the element containing the text element is already selected, click the text element to enter text-edit mode.

When you enter text-edit mode, the entire editable text element is selected. You can type to replace it or click to place the insertion point and edit as in any text area.

Resizing Window Areas

Some regions of various windows can be resized. That is, the user can move the boundary between two adjacent regions of the window.

- ❑ The project window has a visible region resize bar marking the movable boundary between the version tree area and the version description area.
- ❑ When a browser's index is shown, a visible region resize bar marks the movable boundary between the index and display area.
- ❑ In information view, the visit area of a browser is divided into two area; a visible region resize bar marks the movable boundary between them.
- ❑ In table view, a browser's visit area contains a table. The columns of the table can be resized; the movable boundary between columns in the header are active although these borders are not visible.

When the user moves the mouse pointer over a movable boundary between window regions, the cursor changes from the normal arrow to a *resize icon*, which has one of the following forms:



While the mouse pointer is a resize icon, the user can depress the mouse button and drag the movable boundary between the two adjacent regions of the window, thus resizing them.

Visiting

Whenever the name or other representation of an item appears, the user can shift-double-click that representation to visit the indicated item. Typically, the representation of an item will appear in the visit area of a browser; one might also appear in the version display area of the project window. In certain controls (such as the index of a browser), double-clicking an item visits that item; shift-double-click also visits the item because the control ignores the state of the Shift key.

Visiting Undefined Items

Various views may include text that, from context, must be the name of an item of a particular class. For example, in the text view of a spec, Klang syntax might require a word to be the name of a diagram; in picture view of a spec-morphism diagram, Klang syntax might require a word used in the label on a node must be the name of a spec. If the visited item belongs to the current project, the user can shift-double-click such a name even if no item with that name exists. This action creates the item (with an empty defining term) and visits that new item in the default view for its class. The new item is located in the same (sub)project as the item from which it was visited.

Visiting Anonymous Units

The defining term of the visited unit may include an *anonymous unit*, that is, a defining term without a name. For example, in the picture view of a diagram, the morphism attached to an arc may consist of a defining term rather than a named morphism; in the text view of a spec, the Klang defining of the spec may include the defining term of the spec being imported or translated.

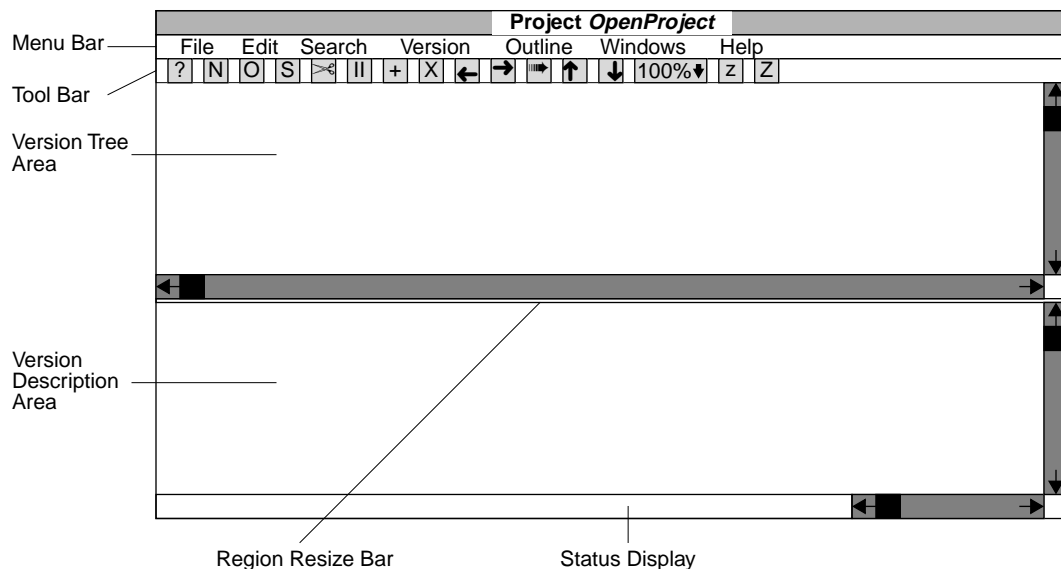
Whenever an anonymous unit appears in the defining term of the visited item, the user may shift-double-click the term to visit the anonymous unit. The unit is visited in the default view for its class, just as if it were a named unit. In the title of the browser, the unit is identified by the context in which it appears.

In a picture view, the user can shift-double-click the node or arc or its label, if shown. In text view, the user can shift-double-click the reserved word that introduces the term. The following table illustrates some anonymous units that might appear in text view and explains how to visit them and how these units are identified by the browser that visits them. The identifier for an anonymous unit includes the identifier for the unit whose defining term contains the anonymous unit. If the containing unit is a named unit, its identifier has the form *Class Name*; if the containing unit is itself anonymous, its identifier has one of the forms shown in the table.

Anonymous Unit	To Visit, Shift-Double-Click:	Identifier in Browser
nodes A: <i>spec-term</i>	The label A or the first word in the <i>spec-term</i> (spec, translate, or colimit)	Spec A: in <i>DiagramIdentifier</i> Translation Spec A: in <i>DiagramIdentifier</i> Colimit Spec A: in <i>DiagramIdentifier</i>
arcs AB: A->B: <i>sm-term</i>	The label AB or the first word in the <i>sm-term</i> (morphism, identity-morphism, translation-morphism, import-morphism, or cocone-morphism)	Morphism AB: in <i>DiagramIdentifier</i> Identity Morphism AB: in <i>DiagramIdentifier</i> Translation Morphism AB: in <i>DiagramIdentifier</i> Import Morphism AB: in <i>DiagramIdentifier</i> Cocone Morphism AB: in <i>DiagramIdentifier</i>
colimit of diagram ... end-diagram	The word diagram in the <i>diagram-term</i>	Defining Diagram of <i>SpecIdentifier</i>
translate <i>spec-term</i> ...	The word translate or the first word in the <i>spec-term</i> (spec, translate, or colimit)	Spec translated by <i>SpecIdentifier</i> Translation Spec translated by <i>SpecIdentifier</i> Colimit Spec translated by <i>SpecIdentifier</i>
import <i>spec-term</i>	The word import or the first word in the <i>spec-term</i> (spec, translate, or colimit)	Spec imported by <i>SpecIdentifier</i> Translation Spec imported by <i>SpecIdentifier</i> Colimit Spec imported by <i>SpecIdentifier</i>
translate ... by {...}	The word by	Translation Morphism of <i>SpecIdentifier</i>

3 Project Window

The project window is the main window of the SPECWARE UI. It is opened when the user starts SPECWARE and remains open throughout the entire interaction. (It can be minimized to free up space on the screen.) The project window allows the user to create a new project or open an existing project. If the project is opened from modification, the user can to create new versions, switch between existing versions, enter or modify the description of a version, and create a library from the active version.



When there is an open project, its name appears in the window title bar (page 35).

Operations are performed in the project window through:

- ☐ The menu bar (page 39)
- ☐ The tool bar (page 47)
- ☐ The version tree area (page 35)
- ☐ The version description area (page 38)

When there is an open project, the status display (page 39) gives information about that project.

The user can adjust the relative sizes of the version tree area and version display area as described in *Resizing Window Areas* on page 31.

Opening and Closing a Project

To open and close projects, the user chooses entries from the File menu.

When the user creates a new project, it is typically opened for modification. The only exception is when the user duplicates an existing project or library and enables the Open in Read-Only Mode checkbox in the Select Project dialog box (page 157).

When the user tries to open an existing project, the system checks the user's access privileges to the project.

- ☐ If the user has no access to the selected project the operation fails and the project window remains unchanged. A message window opens to explain that the user does not have access to the selected project.

➤ Ideally, the UI can check the access to items as it constructs the list of available projects in the Select Project dialog box. If that is done, the user cannot even try to open a project to which s/he does not have access.

- ☐ If the user has read-only (but not read-write) access to the selected project or if the user enables the Open in Read-Only Mode checkbox in the Select Project dialog box, the select project is opened for read-only access.
- ☐ Otherwise, the project is opened for modification.

Only one project can be opened at a time. If the user opens a project when a different project is already open, the open project is closed before the new project is opened.

Opening a Project

When a project is opened, its version tree appears in the project window.

Opening a project for modification makes it the *current project*, allowing its version history to be modified in the project window and allowing the project and the items it contains to be modified in browsers.

- ☐ Any open read/write browser whose visit area displays the project or a subproject or unit it contains updates its status display to show that the item can be modified.
- ☐ Any open browser with a visible project index is updated so that its index reflects the new current project.

Opening a project for read-only access allows the user to view its version history, but not to modify the project, its version history, or the items it contains. No changes are made to browsers when a project is opened for read-only access.

Closing a Project

Whenever the open project is closed, the project window is cleared.

If the project had been opened for modification:

- ☐ Any open read/write browser whose visit area displays the project or a subproject or unit it contains updates its status display to show that the item is read only.
- ☐ Any open browser with a visible project index is updated so that its index is empty.

Title Bar

When no project is open, the title of the window is **Project**. When a project is open, the title of the window is **Project *OpenProject***, where *OpenProject* is the name of the open project.

If the mouse pointer lingers over the title for a short time, a small box pops up next to the mouse pointer showing the name and full path name of the open project. This feature allows the user to see the entire project name even when the window size has been adjusted so that the name does not fit in the title bar. It also allows the user to see where in the universe hierarchy the open project is located.

Version Tree Area

The version tree area allows the user to see the tree of versions for the open project. When a project is opened, its version tree appears in the version tree area with the active version selected.

-
- Only a top-level project in a registry has a development history. The user can open a subproject, making it the current project. When that is done, however, the version tree area and version description area remain blank.
-

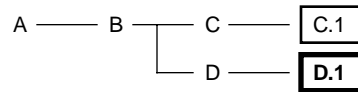
If the project was opened for modification, the user can edit the version tree, deleting versions that were found to be dead ends, and moving versions around within the hierarchy. For example, the user might get rid of all versions but four and arrange those versions into a single flat list although they were actually created along different development branches.

Version Tree

A project version tree is drawn with the parent version to the left of its child versions. Thus, the root version is at the left of the tree and the leaf versions are at the right end of branches.

In the tree, a frozen version is shown with its version name only; an active or dormant version is shown with its version name inside a box. The name of the active version is shown in boldface and its box has a heavier outline than the boxes of dormant versions.

To illustrate these conventions, the following version tree indicates a development history in which the user saved versions A, B, and C, continued development, reverted to version B, continued development, saved version D, and continued development.



Any active or dormant version appears as the leaf of its branch of the development tree. If desired, the user can freeze any dormant or active version. If that is done, a frozen version appears as the leaf of its development branch. If the user freezes the active version and then tries to make changes without switching to a different version, the frozen version is copied and the new copy is made the active version.

Creating a New Version

When the user creates a new version, its node is automatically selected in the version tree and the version description area is made active so that the user can enter a description of the new version.

Mouse-Controlled Actions

The user can select a node in the version tree as described in *Selecting in Non-Text Areas* on page 28. The text description of the selected version is displayed in the version description area. If the project was opened for modification, the mouse can be used to rename versions, move versions within the version tree, and revert to a different version.

The user can rename a version by editing its name in the version tree as described in *Editing Text Elements* on page 30.

The user can move nodes in the tree with the following actions:

- ☐ Select a single node.
- ☐ Point to the selected node, depress the mouse button, and drag. A shadow outline of the selected elements follows the mouse pointer. Release the mouse button to drop the node at its new location.
 - ☐ Drop on a node to make the selected node a new child of that node (after all its existing children).
 - ☐ Drop on an arc to insert the node into that arc.
 - ☐ Drop at a location outside the version tree to cancel the drag and drop operation.

The user can shift-double-click a node to revert to that version.

Keyboard-Controlled Actions

When the version tree area is the active window area, the keyboard can be used to select a version, to revert to a different version, and to move nodes of the tree.

Selecting

When the version tree area is active, the user can type the name of a version to select that version. The first version (from the root of the tree) whose name begins with the typed letters is selected.

- ❑ Pressing Space requires a full-name match; that is, it selects the first version whose full name matches the typed letters.
- ❑ Pressing Tab goes to the next version whose name matches the typed letters.

When a node in the tree is selected, the user can change the selected node using the arrow keys.

Up arrow (↑)

Selects the previous sibling of the selected version. This key is disabled when the selected version is the first child of its parent.

Down arrow (↓)

Selects the next sibling of the selected version. This key is disabled when the selected version is the last child of its parent.

Left arrow (←)

Selects the parent of the selected version. This key is disabled when the root of the version tree is selected.

Right arrow (→)

Selects the first child of the selected version. This key is disabled when the selected node is a leaf node.

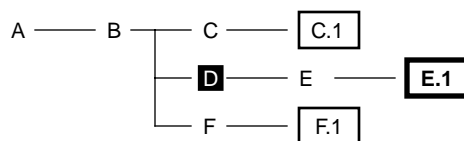
Reverting

If the project was opened for modification and the selected version is not the active version, the user can pressing the Return key to revert to the selected version.

Moving Nodes and Branches

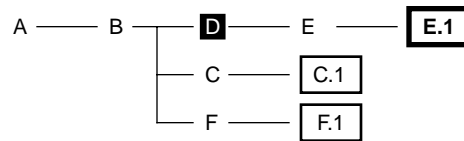
If the project was opened for modification, the user can move the selected version (and, in some cases, its subhierarchy) using the arrow keys with the Shift-Alt modifier.

To illustrate the operation of these keys, consider the following version history.

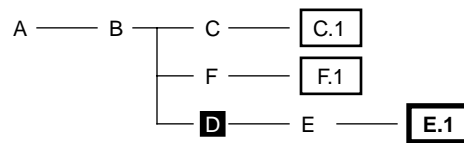


Following each move operation is an illustration of how this version tree would be affected by the move operation.

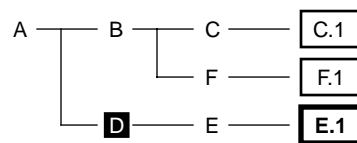
Shift-Alt ↑ Moves the selected version and its subhierarchy to before its previous sibling, moving its branch up in the version tree. This key is disabled when the selected version is the first child of its parent.



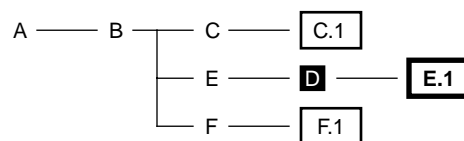
Shift-Alt ↓ Moves the selected version and its subhierarchy to after its next sibling, moving its branch down in the version tree. This key is disabled when the selected version is the last child of its parent.



Shift-Alt ← Moves the selected version and its subhierarchy to the previous level in the hierarchy, moving its branch left in the tree to make it a sibling of its parent. This key is disabled when the parent version is the root of the current version tree.



Shift-Alt → Moves the selected node to the next level in the hierarchy, moving it to the right in the tree and making it the child of its current first child. This key is disabled when the selected node is a leaf node or when the first child of the selected node is an active or dormant node. (Active or dormant nodes cannot be located at non-leaf positions within the tree.)



Version Description Area

When a version is selected in the version tree area, its text description appears in the version description area. The version description is displayed and edited in outline form. The controls and behavior in the version description area are the same as the controls and behavior of the notes subwindow of a browser in Information view. See *Outline Editing* on page 107.

Status Display

The status display below the version description area gives status information about the open project; it is blank when there is no open project. If the mouse pointer lingers over the status display for a short time, a small box pops up next to the mouse pointer showing the content of the status display; this feature allows the user to see the entire status display even when the window size has been adjusted so that all status information is not visible in the status display.

The status display can contain the following status indicators:

Export status; one of the following:

Exported

The open project is exported by its containing project.

(no indicator)

The open project is not exported by its containing project.

Modification status; one of the following:

Read only

The project was opened for read-only access.

Committed

The current project has not changed since it was last committed.

Modified

The current project has changed since it was last committed.

Version status; one of the following:

Version *VersionName*

The open project is a top-level project in its registry and its active version is *VersionName*.

Version *VersionName* of *TopLevelProjectName*

The open project is defined within the hierarchy of *TopLevelProjectName*, whose active version is *VersionName*.

The individual status indicators are separated by semicolons in the status display. For example:

Modified; Active Version Basic.1

Menu Bar

The menu bar of the Project window contains:

- ☐ File menu (page 40)
- ☐ Edit menu (page 42)
- ☐ Search menu (page 43)
- ☐ Version menu (page 44)
- ☐ Outline menu (page 45)
- ☐ Windows menu (page 46)
- ☐ Help menu (page 47)

In the illustrations of menus, italics in the label for a menu entry indicates that the exact label will vary from time to time. For example, *Rename Project CurrentProject* indicates that the label consists of the words *Rename Project* followed by the name of the current project.

File Menu

The File menu of the project window contains the following entries:

File	
New Project	Alt-N
Open Project...	Alt-O
New Project from File...	
Duplicate Project or Library...	
Rename Project <i>CurrentProject</i>	
Close Project <i>OpenProject</i>	
Delete Project <i>CurrentProject</i>	
Create Library from <i>CurrentProject</i> ...	
Page Setup...	
Print Version Tree...	
Print Version Description...	
Print Development History...	
Preferences	
BoringStuff	Recently opened projects and libraries
FunStuff	
Quit	Alt-Q

New Project Creates and opens a new, empty private project for the current user, making it the current project. The new project is named *Untitled-n*; its name can be changed by choosing *Rename Project Untitled-n* from the File menu.

Open Project Brings up the Select Project dialog box (page 157), which allows the user to select the project to be opened. Opens the selected project. |

New Project from File Creates a new private project for the current user from a project interchange file (page 16). Brings up the Open File dialog box (page 142) to prompt the user for the project interchange file. If the user does not select a project interchange file, the operation fails. After the project is created it is opened for modification. |

Duplicate Project or Library

Brings up the Select Project dialog box (page 157), which allows the user to select the project or library to be duplicated. Creates and opens a new private project for the current user by duplicating the selected project or library. The new project is named *Copy-of-SelectedProject* (or *Copy-of-SelectedProject-n*); its name can be changed by choosing *Rename Project CurrentProject* from the File menu.

Rename Project *CurrentProject*

Renames the current project. This entry brings up the Rename dialog box (page 151), which prompts the user for a new name for the project. This entry is greyed out if there is no current project.

Close Project *OpenProject*

Closes the open project, *OpenProject*. This entry is greyed out when there is no open project.

Delete Project *CurrentProject*

Closes the current project then deletes it and all its subprojects and units from the registry that contains it. This entry is greyed out if there is no current project.

Create Library from *CurrentProject*

Creates a library from the active version of the current project. Brings up the Create Library dialog box (page 137), in which the user enters information controlling the creation of the new library. This entry is greyed out when there is no current project.

Page Setup Brings up the Page Setup dialog box (page 137), in which the user can set any relevant printing attributes (for example, page orientation, output device, font, scaling).

Print Version Tree

Prints the content of the version tree area; brings up the Print dialog box (page 151), which allows the user to select printing options. This entry is greyed out when there is no open project.

Print Version Description

Prints the content of the version description area; brings up the Print dialog box (page 151), which allows the user to select printing options. This entry is greyed out when there is no open project.

Print Development History

Prints the entire development history of the open project, giving the name, status (frozen, dormant, or active), and description of each version. Brings up the Print dialog box (page 151), which allows the user to select printing options. This entry is greyed out when there is no open project.

Preferences Brings up the Preferences dialog box (page 144), which allows the user to customize the behavior of SPECWARE.

Recently opened projects and libraries

The section preceding Quit contains the name of the five most recently visited projects or libraries. Choosing any of these entries opens the project or library (closing any project or library that is already open). A chosen

project is opened for modification if the user has read/write access to it; otherwise, it is opened for read-only access. As always, a chosen library is opened for read-only access.

Quit Exits SPECWARE.

Edit Menu

The Edit menu of the project window contains the following entries:

Edit	
Undo <i>Command</i>	Alt-Z
Redo <i>Command</i>	Alt-Y
Cut	Alt-X
Copy	Alt-C
Paste	Alt-V
Select All	Alt-A

-
- All items in this menu are greyed out when there is no current project or when the current project is not at the highest level of its registry (because it has no development history to edit).
-

Undo Command

Undoes *Command*, the last undoable command performed in this window. The label changes to indicate what command will be undone; for example, it changes to Undo Cut after a cut operation; it changes to Can't Undo after an operation that can't be undone (if there are any).

This entry is greyed out before the user performs any undoable command on the open project.

Redo Command

Redoes *Command*, the command that was just undone. The label changes to indicate what command will be redone; for example, it changes to Redo Cut after a cut operation is undone.

This entry is greyed out before the user has undone any command.

Cut Cuts the selected elements from the active window area and places them on the clipboard for future Paste operations. Cutting nodes or subgraphs from the version tree deletes those versions from the project's version history.

This entry is greyed out when nothing is selected in the active area and when the open project is read only.

Copy Copies the selected element(s) from the active window area to the clipboard.

This entry is greyed out when nothing is selected in the active window area.

- Paste** Pastes the elements most recently cut or copied into the active window area. In the version description area, any pasted text replaces any selected text. In the version tree area, any pasted nodes replace any selected nodes; text pasted on a selected node renames the version.
- This entry is greyed out when nothing is selected in the active window area and when the open project is read only.
- Select All** Selects all elements in the active window area (all nodes in the version tree or all text in the version description).

Search Menu

The Search menu of the project window contains the following entries:

Search	
Find/Replace...	Alt-F
Find Next	Alt-G
Find First	
Find Selection	
Display Selection	
Find/Replace in Project...	Shift-Alt-F
Search for Items...	Shift-Alt-S

Find/Replace

Opens or activates the Find/Replace window (page 117), which lets the user find a given text string or pattern in the active window area and optionally replace some or all occurrences with a different string. This entry is greyed out if there is no open project.

Find Next

Finds and selects the next occurrence of the string (or pattern) that was searched for last and scrolls, if necessary, so that the selection is visible. This entry is greyed out if there is no open project or if no Find operation has been performed. In the version tree area, search begins at the root and proceeds top down, depth first.

Find First

Finds and selects the first occurrence in the active window area of the string (or pattern) that was searched for last and scrolls, if necessary, so that the selection is visible. This entry is greyed out if there is no open project or if no Find operation has been performed.

Find Selection

Finds and selects the next occurrence of the currently highlighted text in the active window area and scrolls, if necessary, so that the new selection is visible. This entry is greyed out if there is no open project or if no text is selected in the active window area.

Display Selection

Scrolls the active window area until the current selection is visible; positions the selection close to the top of the display. This entry is greyed out if there is no open project or if nothing is selected in the active window area.

Find/Replace in Project

Opens or activates the Find/Replace in Project window (page 120), which allows the user to search for a given string or pattern in all Klang definitions in the current project, replacing some or all occurrences with a different string. This entry is greyed out if there is no current project.

Search for Items

Opens or activates the Search for Items window (page 129), which allows the user to search an indicated portion of the universe hierarchy for items with particular characteristics.

Version Menu

The Version menu of the project window contains the following entries:

Version	
New Version...	Shift-Alt-V
Delete <i>SelectedVersions</i>	
Rename <i>SelectedVersion</i>	
Freeze <i>SelectedVersions</i>	
Revert to <i>SelectedVersion</i>	Shift-Alt-B

-
- All items in this menu are greyed out when there is no current project or when the current project is not at the highest level of its registry.
-

New Version

Creates a new version of the current project. Brings up the New Version dialog box (page 141) in which the user enters a name of the new version.

Delete *SelectedVersions*

Deletes the selected version(s) from the project development history.

- ☐ If a single version is selected in the version tree, its name appear in the label of this menu entry.
- ☐ If multiple versions are selected, the label is Delete Selected Versions.

Rename *SelectedVersion*

Renames *SelectedVersion*, the version currently selected in the version tree. Brings up the Rename dialog box (page 151) in which the user enters the new name for the version.

Freeze *SelectedVersions*

Freezes selected version(s).

- ❑ If a single version is selected in the version tree, its name appear in the label of this menu entry.
- ❑ If multiple versions are selected, the label is **Freeze Selected Versions**.

This entry is greyed out unless all the selected versions are either active or dormant.

Revert to *SelectedVersion*

Switches to *SelectedVersion*, the version currently selected in the version tree. This entry is greyed out unless a single version other than the active version is selected in the version tree.

Outline Menu

The Outline menu contains the following entries:

Outline	
Higher Level Heading	Shift-Alt ←
Lower Level Heading	Shift-Alt →
Heading to Text	Shift-Alt-T
Move Up	Shift-Alt ↑
Move Down	Shift-Alt ↓
Expand Heading to Section	Alt +
Contract Section to Heading	Alt -

- All items in this menu are greyed out when there is no current project or when the current project is not at the highest level of its registry.

The entries in the Outline menu are used to perform outline editing in the project description area (see *Outline Editing* on page 107).

Higher Level Heading

If all selected paragraphs are body-text paragraphs, converts each one to a heading at the next level below the heading that contains it. Otherwise, promotes each selected heading paragraph to the next higher level and leaves all selected body-text paragraphs unchanged.

Lower Level Heading

Demotes each selected heading paragraph to the next lower level heading and leaves all selected body-text paragraphs unchanged.

Heading to Text

Demotes all selected heading paragraphs to body-text paragraphs.

Move Up

Moves the selected paragraphs up (before the preceding paragraph).

- Move Down Moves the selected paragraphs down (after the following paragraph).
- Expand Heading to Section
 - Expands each selected heading paragraph to show its subheadings and text.
- Contract Section to Heading
 - Collapses the section of each selected heading paragraphs to show only the heading.

Windows Menu

The Window menu of the project window contains the following entries:

Windows

√ Show Tool Bar

New BrowserAlt-B

Command Window

Project Checkbook

Browser 1 - Spec Relation

Browser 3 - Spec Account

Items Meeting Search Criteria

Currently open windows:

The Project window

The first browser window to be opened, which is visiting the specification Relation

The third browser window to be opened, which is visiting the specification Account

The selection-list window; the most recent operation using this window was Search for Items

Show Tool Bar

Toggles between showing and hiding the tool bar (page 47). When the tool bar is being shown, a checkmark (√) appears in front of this entry. A user preference controls whether the tool bar is automatically shown when SPECWARE is started.

New Browser

Opens a new browser. If a project is open, the new browser opens visiting that project.

Command Window

Opens the command window (page 113); if the command window is already open, brings it forward.

Currently Open Windows

The menu entries following the vertical line in the menu correspond to the open windows, in the order in which they were opened. When the user chooses one of these entries, the corresponding window is brought to the front of the display and made the active window. The project window, browsers, dependency windows (page 115), and the selection-list window (page 135), are included in this section; other auxiliary windows are not included.

Help Menu

On-Line Help

The on-line help file should contain a comprehensive description of using SPECWARE with the interface. It should start with a hyperlinked Table of Contents and end with a hyperlinked index. It should support searching.

The help commands bring up the Help window (page 128) displaying the on-line help file; different commands jump to different sections of the file.

Menu entries

The Help menu of the project window contains the following entries:

Help
SPECWARE Help
Klang Grammar
Keyboard Shortcuts
Tutorial

SPECWARE Help

Brings up the Help window displaying Table of Contents for the on-line help file.

Klang Grammar

Brings up the Help window displaying the Klang grammar.

Keyboard Shortcuts

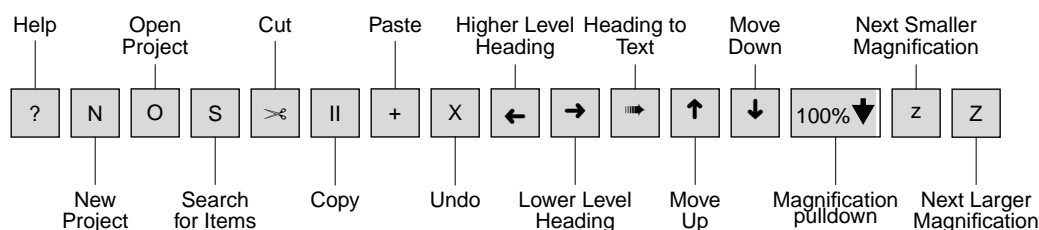
Brings up the Help window displaying the keyboard shortcuts and explaining how to replace the default shortcuts with shortcuts that the user prefers.

Tutorial

Starts an interactive tutorial to show the user how to use SPECWARE with the interface.

Tool Bar

The tool bar of the project window contains buttons that allow the user to perform common operations.



A user preference controls whether the tool bar is shown automatically when SPECWARE is started. Choosing Show Tool Bar from the Windows menu toggles between showing and hiding the tool bar.

If the mouse pointer lingers over a button in the tool bar for a short time, a small box pops up next to mouse pointer showing the meaning of the button (that is, the command it performs).

The tool bar of a browser contains:

- ☐ Buttons for frequently used commands
- ☐ Editing buttons
- ☐ Outline buttons
- ☐ Zoom controls to adjust the magnification of the active window area.

The buttons on the tool bar are greyed out under the same circumstances as their corresponding menu entries are greyed out.

Command Buttons

Current ideas for browser command buttons are:

- ☐ SPECWARE Help
- ☐ New Project
- ☐ Open Project
- ☐ Search for Items

The icons for these buttons are TBD.

Editing Buttons

The editing buttons issue the same commands as the corresponding entries in the Edit menu.

- ☐ Cut
- ☐ Copy
- ☐ Paste
- ☐ Undo

The icons for these buttons are TBD.

Outline Buttons

The outline buttons in the tool bar allow the user to perform the same operations at the entries in the Outline menu (page 45).

- ☐ Higher Level Heading
- ☐ Lower Level Heading
- ☐ Heading to Text
- ☐ Move Up
- ☐ Move Down

The icons for these buttons are TBD.

Zoom Controls

The zoom controls consist of a magnification pulldown and two buttons that change to the next smaller magnification and the next larger magnification, respectively. The zoom controls use a set of ten possibly magnifications. The user can customize the magnifications available in the project window by choosing **Set** from the magnification pulldown.

Magnification Pulldown

The Magnification pulldown menu contains the following options:

25%	Shows the content of the active area at 25% normal size.
50%	Shows the content of the active area at 50% normal size.
60%	Shows the content of the active area at 60% normal size.
70%	Shows the content of the active area at 70% normal size.
80%	Shows the content of the active area at 80% normal size.
90%	Shows the content of the active area at 90% normal size.
100%	Shows the content of the active area at normal size.
120%	Shows the content of the active area at 120% normal size.
150%	Shows the content of the active area at 75% normal size.
200%	Shows the content of the active area at twice the normal size.

Fit Content in Window

Scales the content of the active area as necessary to make it fit within the current size of that window area.

Fit Window to Content

Changes the window size as necessary to make the active area display its content at normal size (expand to full-screen size if the content is too big to display on the screen).

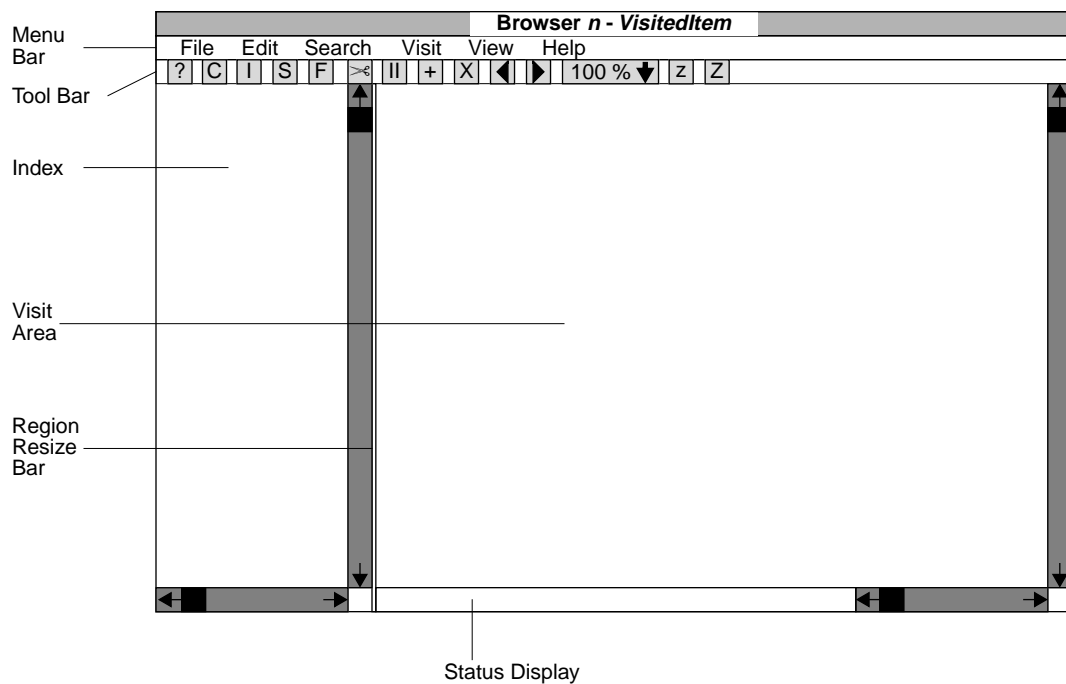
Set...

Opens the Zoom Options dialog box (page 162), which lets the user set the ten magnifications displayed in the pulldown menu.

✂ An alternative to **Set** and a separate dialog box is to make the menu an editable list. How often do we think people will want to customize magnification?

4 Browsers

A browser allows the user to peruse all accessible projects, libraries, and units, visit a particular item, and edit that item if it is, or belongs to, the current project.



When an item is being visited, its class and name appear in the window title bar (page 52).

Operations are performed in a browser through:

- ☐ The menu bar (page 57)
- ☐ The tool bar (page 65)
- ☐ The index (page 53)
- ☐ The visit area (page 54)

When an item appears in the visit area, the status display (page 54) gives information about that item.

The user can adjust the relative sizes of the index and project display area as described in *Resizing Window Areas* on page 31.

This chapter describes the general controls and behavior of browsers. Chapter 5 describes controls and behavior available when visiting items of particular classes. Chapter 6 describes controls and behavior available when particular views are chosen.

Opening a Browser

A browser is opened by the following actions:

- ☐ Choosing New Browser from the Windows menu of the project window.
- ☐ Peeling off a new browser window from an existing browser window.
- ☐ Clicking the Use As Browser Index button on the selection-list window.

Modes of Operation

A browser can operate in read-only or read/write mode. By default, a browser operates in read/write mode when visiting the current project or a unit or subproject of the current project. It operates in read-only mode when visiting a library or a unit of a library, a project other than the current project, or a unit or subproject of a project other than the current project.

The user can toggle the browser's mode by choosing Visit in Read-Only Mode from the File menu. When Visit in Read-Only Mode is enabled, the browser operates in read-only mode, even when visiting the current project or an item it contains.

➤ We may redefine Visit in Read-Only Mode to apply to the currently visited item, not to all items visited in the browser.

Title Bar

When no item is being visited, the title of the n -th browser to be opened in a session is Browser n . When the browser is visiting an item, the title is Browser n - *VisitedItem*, where *VisitedItem* is the class and name of the item being visited.

If the mouse pointer lingers over the title for a short time, a small box pops up next to the mouse pointer showing the class, name, and full path name of the visited item. This feature allows the user to see the entire item identifier even when the window size has been adjusted so that the identifier does not fit in the title bar. It also allows the user to see where in the universe hierarchy the visited item is located.

Index

The index is a scrollable read-only list in which the user can select items to be acted upon by commands performed in the browser (see *Selecting in Non-Text Areas* on page 28).

The user can see the index as a flat list or a hierarchical list by toggling Hierarchical Index in the View menu (see *Flat and Hierarchical Lists* on page 26).

The user can show or hide the index by toggling Show Index in the View menu.

Any change made to the current project (adding, deleting, or renaming subprojects or units; subscribing to libraries) is reflected in the index. Of course, the change may not be visible immediately.

✎ If the UI can check the access to items as it constructs the list, items to which the user does not have read access can be greyed out in the index. We could also differentiate items to which the user has read/write access from those to which s/he has read-only access, perhaps by putting (R) next to read-only items.

Entries in the View menu allow the user to choose any one of four index options, which control the content of index. The chosen option remains in effect whether or not the index is visible.

Universe The index shows the entire universe of accessible items. (This is the default index option when there is no current project.)

Project The index shows all items in the current project. (This is the default index option when there is a current project.)

Filtered The index shows the items that match search criteria entered in the Search for Items window (page 129).

File System The index shows the files and directories in a particular file-system directory.

Additional index options may be available, depending on the class of the visited item.

Mouse-Controlled Operations

The user can:

- ☐ Select items from the index as described in *Selecting in Non-Text Areas* on page 28.
- ☐ Double-click an entry in the index to visit the corresponding item in the browser's visit area.

Depending on the class and view of the item displayed in the visit area, the user may also be able to drag items from the index to the visit area. In most cases, a drag-and-drop operation is identical to a copy-and-paste operation. Any exceptions are described in Chapter 5, *Class-Specific Controls and Behavior in Browsers*, and Chapter 6, *View-Specific Controls and Behavior in Browsers*.

Keyboard-Controlled Operations

When the index is the active area, the user can type the name of an item to select that item. The first item (from the top of the list) whose name begins with the typed letters is selected.

- ☐ Pressing Space requires a full-name match; that is, it selects the first item whose full name matches the typed letters.
- ☐ Pressing Tab goes to the next item whose name matches the typed letters.
- ☐ Pressing Return visits the selected item.

Visit Area

The visit area displays the item currently being visited. A browser can display a number of different views of the visited item, depending on its class. The exact content and supported operations in this area depends on the class of the item and the chosen view.

- ☐ For details about working with the text view, see *Text View* on page 93.
- ☐ For details about working with the picture view, see *Picture View* on page 95.
- ☐ For details about working with the information view, see *Information View* on page 106.
- ☐ For details of the present-value view, see *Present-Value View* on page 109.
- ☐ For details about working with table views, see *Table Views* on page 109.

The class of the item determines the default view option that is displayed when the user first visits the item; a user preference can override this default. The user can switch from one view to another by selecting view options from the View menu.

Status Display

The status display below the visit area give status information about the item currently shown in the visit area; it is blank when the no item is being visited. If the mouse pointer lingers over the status display for a short time, a small box pops up next to mouse pointer showing the content of the status display; this feature allows the user to see the entire status display even when the window size has been adjusted so that all status information is not visible in the status display.

The status display can contain the following status indicators:

Export status; one of the following:

Exported

The visited item is exported by its containing project.

(no indicator)

The visited item is not exported by its containing project.

Modification status; one of the following:

Read only

The visited item cannot be modified (at least in this browser's current operating mode).

Committed

The visited item has not changed since it was last committed.

Modified

The visited item has changed since it was last committed.

Evaluation status (for units only); one of the following:

Unparsable

The Klang definition of the visited unit is a syntactically illegal.

Has Value

The visited unit has a present value.

Invalid The visited unit is parsable and has been evaluated since it was committed, but evaluation failed.

Not Evaluated

The visited unit is parsable but has not been evaluated since it was committed (or evaluation is in progress and has not completed).

Version status; one of the following:

Version *VersionName*

The visited item is the current project; it is a top-level project in its registry; and its active version is *VersionName*.

Version *VersionName* of *TopLevelProjectName*

The visited item is defined within the hierarchy of *TopLevelProjectName*, whose active version is *VersionName*.

The individual status indicators are separated by semicolons in the status display. For example, when the visited item is an exported unit that is modified and has not been evaluated, the status display contains:

Exported; Modified; Not Evaluated; Version Basic.1 of Project FunStuff

When the visited item is an exported unit in a library to which the current project subscribes, the status display contains:

Exported; Read only; Has Value; Version Basic.1 of Project FunStuff

When the current project is visited and has been changed since it was last committed, the status display contains:

Modified; Version Basic.1

Window Settings

The UI records the following window settings for each browser:

- ☐ The selected index option; for the filtered index option, the contents of the index
- ☐ Whether the index is displayed or hidden
- ☐ Whether the index is a hierarchical or a flat list
- ☐ The visible part of the index (scroll position)
- ☐ Any selection in the index
- ☐ Window size
- ☐ Window position
- ☐ Relative sizes of visit area and index

If the user closes all browsers, the UI remembers these settings for the last browser to be closed. If the user then opens another browser in the same session, it will have the settings of the most recently closed browser. At the end of the session, the UI remembers the window size, window position, and relative sizes of the visit area and index for the most recently active browser. In the user's next interaction with SPECWARE, the first browser to be opened will have the remembered window size and position.

Visit History

Each browser keeps a *visit history* of the items that have been visited in that browser. Each entry in the visit history records the following visit settings:

- ☐ The identity of the item which is being visited
- ☐ The selected view option
- ☐ The visible part of the canvas (scroll position) in the visit area
- ☐ Any selection in the visit area (or position of the insertion bar in text view)
- ☐ If the visited item is a project or a library (page 69) and a table view is selected, which columns and rows of the table that are currently being displayed
- ☐ If picture view (page 95) is selected:
 - ☐ Which labels (if any) are being displayed
 - ☐ Which defining diagrams are expanded and which are contracted
 - ☐ For if the node class of the diagram is a diagram class, which nodes and arcs are showing details and which are hiding details
 - ☐ Layout of the graph
- ☐ If information view (page 106) is selected, the relative sizes of the statistics area and the notes area
- ☐ If a table view (page 109) is selected:
 - ☐ How the table is sorted
 - ☐ Which table-display options are enabled

The UI saves most recent visit settings for each item. The next time the user visits an item that has been visited in the past, the visit settings will be the same as on the last visit to that item. In addition, the UI saves the most recent visit setting for each view of each item. If the user selects a view that has been selected before for the same item, the visit settings will be the same as on the last visit to the item in that particular view.

>& I believe the last decision was to save settings for the project, not on a user-by-user basis. So if two users are working on the project, and user B visited an item most recently, when user A visits the item, s/he will see it with user B's last visit settings. If we find that this causes problems, we can save user-specific settings in a future release.

Peeling Off a Browser

The user can peel off a new browser by selecting **Peel Off Browser** from the File menu or by using the mouse as follows:

- ☐ Press the Ctrl key, point to the window title bar, and depress the mouse button.
- ☐ With the Ctrl key and the mouse button both depressed, move the mouse. A shadow of the window follows the mouse pointer.
- ☐ Release the mouse button to create the new browser at the shadow's current location.

After the peel-off operation, the new browser opens visiting the item that was visited last in the original browser; its index, view, and visit history are as in the original browser. The original browser shows the previous item visited (as if the user has chosen **Back** from the Visit menu). The visit history of the original browser remains unchanged.

If the original browser has only visited one item, both browsers show that item. If the original browser had not visited any items, both browsers have empty visit areas.

Menu Bar

The menu bar of a browser always contains:

- ☐ File menu (page 58)
- ☐ Edit menu (page 60)
- ☐ Search menu (page 61)
- ☐ Visit menu (page 62)
- ☐ View menu (page 64)
- ☐ Help menu (page 65)

When an item is being visited in the browser, additional menus may be added that are specific to the class of that item or the selected view. Any such class-specific and view-specific menus are added between the View and Help menus so that the Help menu is always the last one in the menu bar.

In the illustrations of menus, italics in the label for a menu entry indicates that the exact label will vary from time to time. For example, **Commit *VisitedItem*** indicates that the label consists of the word **Commit** followed by identifier (class and name) of the item being visited.

File Menu

The File menu in a browser generally contains the following entries:

File	
Commit <i>VisitedItem</i>	Alt-S
Commit <i>VisitedItem</i> with Dependencies	
Roll Back <i>VisitedItem</i>	
Roll Forward <i>VisitedItem</i>	
Show Dependencies...	
Visit in Read-Only Mode	
Peel Off Browser	
Close Browser	Alt-W
Page Setup...	
Print One	
Print <i>VisitedItem</i> ...	Alt-P
Print Selection...	Shift-Alt-P
Write <i>VisitedItem</i> to File...	

When an item is being visited in the browser, additional entries may be added and other entries may be replaced or removed depending on the class of the visited item.

Commit *VisitedItem*

Commits the item in the visit area. If a project is being visited, commits any modified units in the project. If the visited item is an anonymous unit, this entry commits the named unit whose defining term contains this anonymous unit's defining term.

This entry is greyed out if the visit area is empty, if a library or file is being visited, if this browser is in read-only mode, or if the item in the visit area has not been modified since it was last committed.

Commit *VisitedItem* with Dependencies

Commits the item in the visit area plus any items on which it depends. If a project is being visited, commits any modified units in the project and any items on which they depend.

This entry is greyed out if the visit area is empty, if a file or anonymous unit is being visited, if this browser is in read-only mode, or if the item in the visit area has not been modified since it was last committed.

Roll Back *VisitedItem*

Rolls back to the previous committed state of the item in the visit area. This entry is greyed out if the visit area is empty, if a file or anonymous unit is being visited, if this browser is in read-only mode, or if there is no previous committed state of the item.

Roll Forward *VisitedItem*

Rolls forward to the next committed state of the item in the visit area. This entry is greyed out if the visit area is empty, if a file or anonymous unit is being visited, if this browser is in read-only mode, or if there is no next committed state of the item.

Show Dependencies

Shows dependencies of the visited item. If a dependency window (page 115) for the visited item exists, brings that window forward; otherwise, opens a new dependency window for the visited item. This entry is greyed out if the visit area is empty or if a file or anonymous unit is being visited.

Visit in Read-Only Mode

Toggles whether to enforce read-only mode for this browser. When read-only mode is enforced, a checkmark (✓) appears in front of this entry and any item visited in this browser is visited in read-only mode, even if it belongs to the current project.

➤ Should we change the meaning of Visit in Read-Only Mode to apply only to the currently visited item, not to all items visited in the browser?

Peel Off Browser

Opens a new browser by peeling off the most recently visited item from this browser (see *Peeling Off a Browser* on page 57).

Close Browser

Closes this browser.

Page Setup Brings up the Page Setup dialog box (page 137), in which the user can set any relevant printing attributes (for example, page orientation, output device, font, scaling).

➤ It is standard to include Page Setup in the File menu. We might also have a tab for Page Setup options in the Preferences dialog box (page 144), but I suspect that's overkill.

Print One Prints a single copy of the item in the visit area, with default printing options. This entry is greyed out if the visit area is empty.

Print *VisitedItem*

Prints the item in the visit area; brings up the Print dialog box (page 151), which allows the user to select printing options. This entry is greyed out if the visit area is empty.

Print Selection

Prints the selected portion of the visit area; brings up the Print dialog box (page 151), which allows the user to select printing options. This entry is greyed out if nothing is selected in the visit area is empty.

Write *VisitedItem* to File

Writes the item in the visit area to a file. If the visited item is modified, a message window opens informing the user of this fact and giving the user the options of continuing, cancelling, or committing the item before writing it. This entry brings up the Write to File dialog box (page 159), which allows the user to specify the desired view, file name, and file format. This entry is greyed out if the visit area is empty.

Edit Menu

The Edit menu of a browser always contains the following entries:

Edit	
Undo <i>Command</i>	Alt-Z
Redo <i>Command</i>	Alt-Y
Cut	Alt-X
Copy	Alt-C
Paste	Alt-V
Select All	Alt-A

When an item is being visited in the browser, additional menu entries may be added that are specific to the class of the visited item or the chosen view.

Undo *Command*

Undoes *Command*, the last undoable command performed (in any browser) on the item being visited. The label changes to indicate what command will be undone; for example, it changes to Undo Cut after a cut operation; it changes to Can't Undo after an operation that can't be undone (if there are any).

This entry is greyed out:

- ☐ When the visit area is empty
- ☐ When this browser is in read-only mode
- ☐ Before any change has been made (in any browser) to the visited item.

Redo *Command*

Redoes *Command*, the command that was just undone. The label changes to indicate what command will be redone; for example, it changes to Redo Cut after a cut operation is undone.

This entry is greyed out before the user has undone any command.

Cut

Cuts the selected elements in the visit area (placing them on the clipboard). The meaning of a cut in a non-text area varies with the class being visited (but is the expected behavior of removing the element from the context in which it appears).

Cutting the name of an item places its name and defining term on the clipboard so that it could be pasted into a context that expects a defining term of the item's class. Cutting an item in a context that removes it from a project or diagram saves the item's full path name on the clipboard so that undoing the operation restores the correct item to the project or diagram.

This entry is greyed out:

- ☐ When the visit area is empty.
- ☐ When this browser is in read-only mode.
- ☐ When nothing is selected in the visit area.

Copy Duplicates the selected element(s) in the active window area, placing the duplicate elements on the clipboard.

Copying the name of an item places its name and defining term on the clipboard so that it could be pasted into a context that expects a defining term of the item's class.

This entry is greyed out when nothing is selected in the active window area.

Paste Pastes the elements that were placed on the clipboard by the most recently cut or copy operation into the visit area, replacing any selected elements. The meaning of a paste operation and the way in which elements are replaced depends on the class of the item in the visit area and the chosen view.

This entry is greyed out unless a read/write item is being visited. In some contexts, it is greyed out if nothing is selected in the visit area.

The Paste operation fails (causing a beep) if the element(s) on the clipboard are not of a class that can be pasted in the current context.

Select All Selects all elements in the active area.

Search Menu

The Search menu of a browser always contains the following entries:

Search	
Find/Replace...	Alt-F
Find Next	Alt-G
Find First	
Find Selection	
Display Selection	
Find/Replace in Project...	Shift-Alt-F
Search for Items...	Shift-Alt-S

When an item is being visited in the browser, additional menu entries may be added that are specific to the class of the visited item or the chosen view.

Find/Replace

Opens or activates the Find/Replace window (page 117), which lets the user find a given text string or pattern in the visit area and optionally replace some or all occurrences with a different string. This entry is greyed out if the visit area is empty.

Find Next

Finds and selects the next occurrence of the string (or pattern) that was searched for last and scrolls, if necessary, so that the selection is visible. This entry is greyed out if the visit area is empty or if no Find operation has been performed.

Find First

Finds and selects the first occurrence in the visit area of the string (or pattern) that was searched for last and scrolls, if necessary, so that the selection is visible. This entry is greyed out if the visit area is empty or if no Find operation has been performed.

Find Selection

Finds and selects the next occurrence of the currently highlighted text in the visit area and scrolls, if necessary, so that the new selection is visible. This entry is greyed out if the visit area is empty or if no text is selected in the visit area.

Display Selection

Scrolls the visit area until the current selection is visible; positions the selection close to the top of the display. This entry is greyed out if the visit area is empty or if nothing is selected in the visit area.

Find/Replace in Project

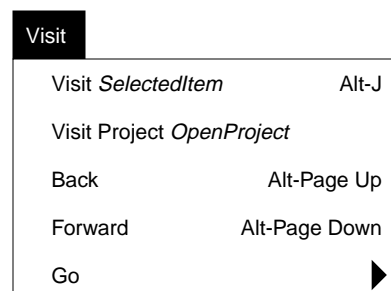
Opens or activates the Find/Replace in Project window (page 120), which allows the user to search for a given string or pattern in all Klang definitions in the current project, replacing some or all occurrences with a different string. This entry is greyed out if there is no current project.

Search for Items

Opens or activates the Search for Items window (page 129), which allows the user to search an indicated portion of the universe hierarchy for items with particular characteristics.

Visit Menu

The Visit menu of a browser contains the following entries:



Visit *SelectedItem* adds a new node to this browser's visit history; the other menu entries go to nodes already in the visit history.

Visit *SelectedItem*

Visits *SelectedItem*, the currently selected item in the active window area.

If visit area is active and the visited item is or belongs to the current project, the name of a non-existent item may be selected; this menu entry creates an empty defining term for an item of the appropriate class and visits that newly defined item. The new item is put in the same (sub)project as the item from which it was visited.

This entry is greyed unless a single item is selected in the active area.

Visit Project *OpenProject*

Visits *OpenProject*, the current project. This entry is greyed out if there is no open project.

Visiting a project this way is identical to selecting the project from the universe index and choosing Visit *SelectedItem*. That is, a particular project is visited (which happens to be the open project now). That project remains in the browser's visit history even if that project is closed and a different project is later opened.

Forward Visits the next node in this browser's visit history. This entry is greyed out if no items have been visited or if the visit area displays the last node in this browser's visit history.

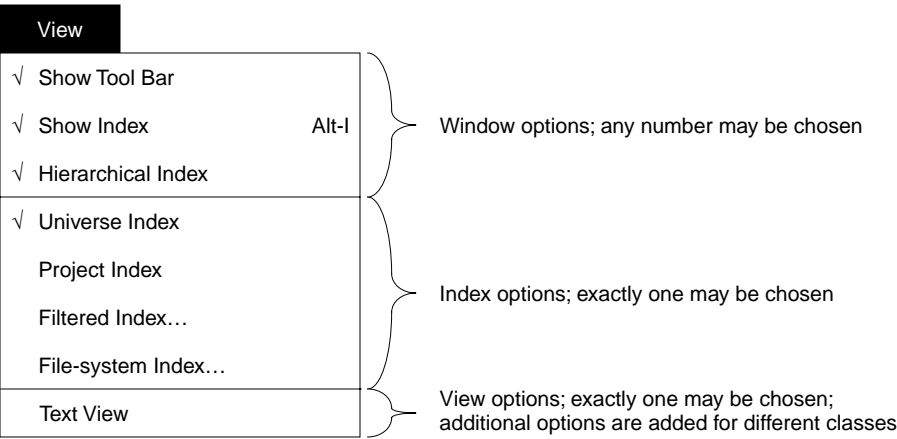
Back Visits the previous node in this browser's visit history. This entry is greyed out if no items have been visited or if the visit area displays the first node in this browser's visit history.

Go Goes to a node in this browser's visit history. The submenu contains the browser's visit history, with the most recently visited item at the top of the submenu. Each entry in the submenu contains the class, name, and pathname of the visited item. Any item that was visited more than once appears only once in the visit history (at the location corresponding to its most recent visit).

➤ I assume that the user will most often use the Go submenu to jump to recently visited items, so it is most useful to order it showing the reversed order in which items were visited **not** the order in which they were added to the visit history. A user who knows he/she wants to retrace steps would probably use Forward and Back instead. An alternative would be to keep the list in the order in which items were added to the visit history and simply highlight the item currently being visited.

View Menu

The View menu of a browser allows the user to control how the browser appears and what view of the visited item is shown. Entries in this menu allow the user to select window options, index options, and view options.



The window options are the same for all visited items but the index and view options may change. Depending on the class of the item being visited, additional sections may be added to the menu following the view options. Default settings for window and index options can be overridden by user preferences.

Window Options

- Show Tool Bar
Toggles between showing and hiding the tool bar (page 65). When the tool bar is being shown, a checkmark (✓) appears in front of this entry.
- Show Index
Toggles between showing and hiding the index (page 53). When the index is being shown, a checkmark (✓) appears in front of this entry.
- Hierarchical Index
Toggles between displaying the index as a hierarchical list and displaying it as a flat list. When the index is displayed in hierarchical form, a checkmark (✓) appears in front of this entry.

Index Options

- Universe Index
Sets the index to contain all items in the universe.
- Project Index
Sets the index to contain all items in the current project.
- Filtered Index
Sets the index to contain only those items that match some search criteria. Opens or activates the Search for Items window (page 129). The user enters search criteria in that window and performs the search; this browser's index displays the results of the search.

File-System Index

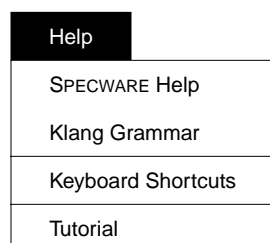
Sets the index to contain the files in some file-system directory. Brings up the Select Directory dialog box (page 155). The user selects a directory in that dialog box; this browser's index displays the contents of the selected directory. If the user cancels the Select Directory dialog box, this browser's index option is not changed.

View Options

The collection of view options depends on the class of the visited unit. Choosing any view option shows the visited item in that view.

Help Menu

The Help menu in a browser is identical to the Help menu in the project window. The menu is duplicated in browsers to ensure that on-line help is readily available regardless of the window in which the user is working.

**SPECWARE Help**

Brings up the Help window displaying Table of Contents for the on-line help file.

Klang Grammar

Brings up the Help window displaying the Klang grammar.

Keyboard Shortcuts

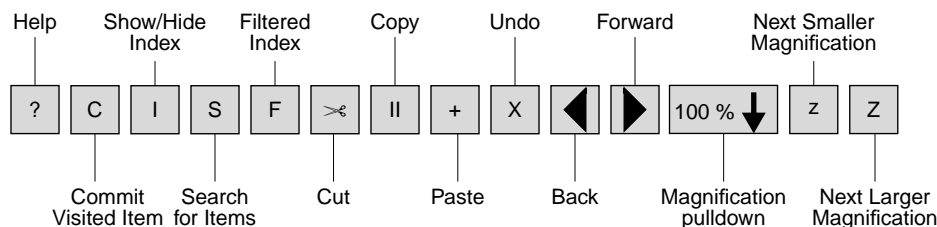
Brings up the Help window displaying the keyboard shortcuts.

Tutorial

Starts an interactive tutorial to show the user how to use SPECWARE with the interface.

Tool Bar

The tool bar of a browser contains buttons that allow the user to perform common operations.



A user preference controls whether the tool bar is shown automatically when a browser opens. Choosing **Show Tool Bar** from the View menu toggles between showing and hiding the tool bar.

If the mouse pointer lingers over a button in the tool bar for a short time, a small box pops up next to mouse pointer showing the meaning of the button (that is, the command it performs).

The tool bar of a browser contains:

- ☐ Buttons for frequently used commands
- ☐ Editing buttons
- ☐ Navigation buttons to move around in the browser's visit history.
- ☐ Zoom controls to adjust the magnification of the visit area.

Additional buttons may be added to the tool bar in different views.

The buttons on the tool bar are greyed out under the same circumstances as their corresponding menu entries are greyed out.

Command Buttons

Current ideas for browser command buttons are:

- ☐ SPECWARE Help
- ☐ Commit Visited Item
- ☐ Show/Hide Index
- ☐ Search for Items
- ☐ Filtered Index

The icons for these buttons are TBD.

Editing Buttons

The editing buttons issue the same commands as the corresponding entries in the Edit menu.

- ☐ Cut
- ☐ Copy
- ☐ Paste
- ☐ Undo

The icons for these buttons are TBD.

Navigation Buttons

The navigation buttons issue the same commands as the corresponding entries in the Visit menu.

- ☐ Forward
- ☐ Back

These buttons use icons that are standard in web browsers and other readers for hyperlinked documents.

Zoom Controls

The zoom controls consist of a magnification pulldown and two buttons that change to the next smaller magnification and the next larger magnification, respectively. The zoom controls use a set of ten possibly magnifications. The user can customize the magnifications available in a particular browser by choosing **Set** from the magnification pulldown.

Magnification Pulldown

The Magnification pulldown menu contains the following options:

- 25% Shows the item at 25% normal size.
- 50% Shows the item at 50% normal size.
- 60% Shows the item at 60% normal size.
- 70% Shows the item at 70% normal size.
- 80% Shows the item at 80% normal size.
- 90% Shows the item at 90% normal size.
- 100% Shows the item at normal size.
- 120% Shows the item at 120% normal size.
- 150% Shows the item at 75% normal size.
- 200% Shows the item at twice the normal size.

Fit Item in Window
Scales the item as necessary to make it fit within the current size of the visit area.

Fit Window to Item
Changes the window size as necessary to make the visit area contain the item at normal size (expand to full-screen size if item is too big to display on the screen)

Set... Opens the Zoom Options dialog box (page 162), which lets the user set the ten magnifications displayed in the pulldown menu.

✂ An alternative to Set and a separate dialog box is to make the menu an editable list. How often do we think people will want to customize magnification?

5 Class-Specific Controls and Behavior in Browsers

This chapter describes the controls that are added to a browser when an item of a particular class is being visited in that window. It provides details about the meaning of common actions (such as cut and paste) for various classes:

- ☐ Project and Library classes (page 69)
- ☐ Klang spec (page 81)
- ☐ Klang spec morphism (page 84)
- ☐ Diagram classes (page 90)
- ☐ File

This chapter lists the available views for each class and indicates which of these is the default view for the class. The user can change the default view for any class in the Preferences dialog box (page 144).

Project and Library Classes

When a browser visits a project or library, the user can select any one of the following views:

- ☐ Hierarchical table view (default)
- ☐ Flat table view
- ☐ Text
- ☐ Information

✂ Lambert has suggested that we should also support a 2D view with icons similar to a Macintosh Finder window in Icon or Small Icon view. I don't believe this adds anything except, perhaps, the ability to see more subprojects in the visit area because they can appear in columns. In particular, it does not allow the user to see any information about the subprojects. I suggest that we do not include such a view in the first release, but add it later if lots of users ask for it.

The following changes are made to the browser's standard controls:

- ☐ Class-specific entries are added to the File menu (page 73).
- ☐ The supported views are added to the View menu (page 75).
- ☐ When either of the table views is chosen:
 - ☐ A section containing column-display options and a section containing table-display options are added to the View menu.
 - ☐ An Order menu (page 77) is added to the menu bar.
 - ☐ If the visited item is a project (not a library), a Project menu (page 78) is added to the menu bar.

Table Views

In the two table views, each row corresponds to an item in the project or library.

- ☐ The hierarchical table view of a project or library is a hierarchical table (page 111). A container row in the table is a row describing a subproject or sublibrary; a noncontainer row is a row describing a unit.

The text in the Class and Name columns (if shown) is indented to show the item's level within the project hierarchy.

- ☐ The flat table view of a project or library is a non-hierarchical table.

The table also contains a row for every library or sibling subproject to which the visited item (or any of its subprojects or sublibraries) subscribes. The text in the Class and Name columns (if shown) is in italics to indicate that the item is not defined in the visit project or library. In a hierarchical table view, the row appears as noncontainer row; the user cannot look at the hierarchy of the subscribed-to item within the table. (The user can, of course, visit the subscribed-to item to see its hierarchy.)

The column-display options chosen from the View menu (page 75) control what information columns (status, class, name, path name, commit date) appear in the table. The ordering options chosen from the Order menu (page 77) control how rows at the same level in the hierarchy are ordered from top to bottom.

Only cells in the Name column may be edited.

For details about working with the table views, see *Table Views* on page 109.

Target Project

If the user is visiting the current project or a subproject it contains, various operations add subprojects, libraries, or units to the visited project. In all cases, the new items are put in the *target project*, which is identified based on what is selected in the visit area when some user action causes items to be added.

- ☐ If a subproject folder is selected in the visit area, that subproject is the target project.
- ☐ If a library or unit is selected in the visit area, the (sub)project containing that library or unit is the target project.
- ☐ If a sublibrary is selected in the visit area, the target project is the (sub)project containing the library to which the selected sublibrary belongs.
- ☐ If more than one item is selected in the visit area, the target project is the largest project containing all selected items (possibly the visited project itself).

- ❑ If nothing is selected in the visit area, the target project is the visited project.

Cut-and-Paste Operations

If the user is visiting the current project or a subproject it contains in a table view, the user can modify the visited project's contents using cut-and-paste operations to rename items, delete items from the visited project, and add items to the visited project.

Renaming Items. The user can enter text-edit mode on a cell in the Name column as described in *Editing Text Elements* on page 30; after that, Cut and Paste can be used to modify the items's name as in any text area.

Deleting Items. If one or more rows of the table are selected, Cut removes the corresponding items from the project. Cutting a subproject removes that subproject and all its contents from the visited project.

Adding Items. If the last copy or cut operation placed one or more items on the clipboard, the user can add those items to the target project by pasting.

Drag-and-Drop Operations

If the user is visiting the current project or a subproject it contains in a table view, drag-and-drop operations allow the user to rearrange the visited project's hierarchy and add items to it.

Dragging in the Table. The user can drag within the table to move items around within the visited project's hierarchy. Dragging the row for a unit moves just that unit; dragging the row for a subproject moves that subproject and its contents to the new location.

The user performs the following actions to drag and drop within the table.

- ❑ Select the row of the table corresponding to the item to be moved. A shadow box containing the item's class and name appears and moves with the mouse pointer.
- ❑ With the mouse button still depressed, move the mouse pointer inside the visit area.
- ❑ Release the mouse button to drop the dragged item:
 - ❑ Point to a subproject; that subproject is highlighted. Release the mouse button to move the dragged item into that subproject.
 - ❑ Release the mouse button without pointing to any subproject to move the dragged item to the top level of the visited project.
 - ❑ Release the mouse button outside the visit area or on the original row to cancel the drag-and-drop operation.

Dragging from the Index. The user can drag from the index to the table to add items to the visited project or to rearrange items within the project. If a dragged item was not already part of the current project, the drag-and-drop operation is equivalent to a copy-and-paste operation. It duplicates the item and adds a deep copy to the visited project at the indicated location. Duplicated items may be renamed to avoid name conflicts, just as occurs when the user selects *Duplicate Selected Items* from the Project menu (see page 79).

If a dragged item was already part of the current project, the drag-and-drop operation is equivalent to a cut-and-paste operation; the item is effectively moved from its original location within the current project to the indicated location. Moving a subproject moves that subproject and its contents to the new location.

The user performs these actions to drag and drop from the index to the table:

- ☐ Select any number of items to be dragged.
- ☐ Point to a selected item in the index and depress the mouse button. A shadow outline of the selected items appears and moves with the mouse pointer.
- ☐ With the mouse button still depressed, move the mouse pointer into the visit area.
- ☐ Release the mouse button to drop the selected items:
 - ☐ Point to a subproject; that subproject is highlighted. Release the mouse button to add the items to that subproject.
 - ☐ Release the mouse button without pointing to any subproject to add the items to the target project.
 - ☐ Release the mouse button outside the visit area to cancel the drag-and-drop operation.

Text View

The text view of a project or library is its Klang definition (see *Defining Terms* on page 16). See *Text View* on page 93 for a description of the editing capabilities available in this view.

Information View

The information view of a project or library contains statistics and user notes about the item.

The statistics include:

- ☐ The item's path in the universe hierarchy
- ☐ The item's creation date
- ☐ For a project, the name and modification date of the its active version
- ☐ The total number of items of each class that the project or library contains, both as immediate elements and at all levels of the hierarchy.
- ☐ The names of the items that the project or library exports.
- ☐ (Possibly) the users or groups who have read and write access to the project or who have read access to the library.

The user can enter and modify notes about the project as described in *Information View* on page 106.

File Menu

When a project or library is being visited, entries are added to the File menu:

File	
Commit <i>VisitedItem</i>	Alt-S
Commit <i>VisitedItem</i> with Dependencies	
Commit <i>SelectedItems</i>	Class-specific menu entry for Project (not Library)
Commit <i>SelectedItems</i> with Dependencies	Class-specific menu entry for Project (not Library)
Roll Back <i>VisitedItem</i>	
Roll Forward <i>VisitedItem</i>	
Show Dependencies...	
Visit in Read-Only Mode	
Peel Off Browser	
Close Browser	Alt-W
Page Setup...	
Print One	
Print <i>VisitedItem</i> ...	Alt-P
Print Selection...	Shift-Alt-P
Print <i>SelectedItems</i>	Class-specific menu entry
Write <i>VisitedItem</i> to File...	
Write <i>SelectedItems</i> to File...	Class-specific menu entry

- Because libraries are read only, all menu entries in the first section are greyed out when the visited item is a library.

All class-specific menu entries apply to items selected in the visit area. If a single item is selected, its class and name appear in the labels of the class-specific menu entries; if multiple items are selected, *Selected Items* appears in the labels.

The class-specific menu entries are:

Commit *SelectedItems*

Commits the items selected in the visit area. This entry is greyed out if the browser is in read-only mode, if no items are selected in the visit area, or if none of the selected items needs to be committed.

Commit *SelectedItems* with Dependencies

Commits the items selected in the visit area and any items on which they depend. This entry is greyed out if the browser is in read-only mode, if no items are selected in the visit area, or if none of the selected items needs to be committed.

Print *SelectedItems*

Prints the item(s) selected in the visit area as if they had been visited in the default view for their class. This entry brings up the Print dialog box (page 151), which allows the user to select printing options.

Write *SelectedItems* to File

Writes the item(s) selected in the visit area to a file. If any of the selected items are modified, a message window opens informing the user of this fact and giving the user the options of continuing, cancelling, or committing the items before writing them. This entry brings up the Write to File dialog box (page 159), which allows the user to specify the desired view, file name, and file format.

This entry is greyed out if nothing is selected in the visit area.

-
- Lambert: Is it sufficient to support Print *SelectedItems* and Write *SelectedItems* only when the user is visiting a project or library? I think it would be overkill to support these commands when any item is visited.
-

View Menu

When a project or library is being visited, entries for the supported views appear in the view options section. When either of the table views is chosen, column- and table-display options are added to the View menu. Default settings for window, index, view, and column-display, and table-display options can be overridden by user preferences.

View	
√ Show Tool Bar	
√ Show Index	Alt-I
√ Hierarchical Index	
Universe Index	
√ Project Index	
Filtered Index...	
File-system Index...	
√ Hierarchical Table View	View options for projects and libraries; exactly one may be chosen
Flat Table View	
Text View	
Information View	
√ Status	Class-specific column-display options (added in either table view); any number may be chosen
√ Class	
√ Name	
Path Name	
Commit Date	
Show Lines	Table-display options (added in either table view); any number may be chosen
Include Time with Date	
Filter Rows of Table...	

Column-Display Options

Status	Displays the status of each item in the project. The status column contains three subcolumns with symbols that indicate the status of the item. The first subcolumn gives the item's export status. It can contain the following symbols: E The item is exported by its containing project. (blank) The item is internal to its containing project. The second subcolumn gives the item's modification status. It can contain the following symbols: R The item is read-only (because it is or belongs to a library).
--------	--

M The item has been modified since it was last committed.

(blank) The item has been committed.

The third subcolumn applies to units only; it gives the unit's evaluation status

U The Klang definition of the unit is a syntactically illegal (unparsable).

V The unit has a present value.

I The unit is invalid; that is, its Klang definition is parsable and the unit has been evaluated since it was committed, but evaluation failed.

(blank) The unit's Klang definition is parsable but the unit has not been evaluated since it was committed (or evaluation is in progress and has not completed).

If the mouse pointer lingers over a cell in the status column for a short time, a small box pops up next to mouse pointer showing the verbose form of the status information. The verbose form is the same form that is shown in the status display of a browser visiting the item; see *Status Display* on page 54.

Class Displays the class of each item.

Name Displays the name of each item.

Path Name Displays the path name of each item.

Commit Date
Displays the last commit date of each item.

✂ I didn't include size because I couldn't think of a meaningful measure of size that would be valid for all items.

Table-Display Options

Show Lines Toggles whether to draw a pale horizontal line between each row in the table.

Include Time with Date

Toggles whether to includes time (hour and minute) when displaying the commit date.

Filter Rows of Table

Toggles whether the table should include rows for all items in the visited project or library, or only those items that meet some search criteria. When the table is filtered, a checkmark (✓) appears in front of this entry and the window title includes the word Filtered in parentheses before the item identifier. For example:

Browser 2 - (Filtered) Project FunStuff

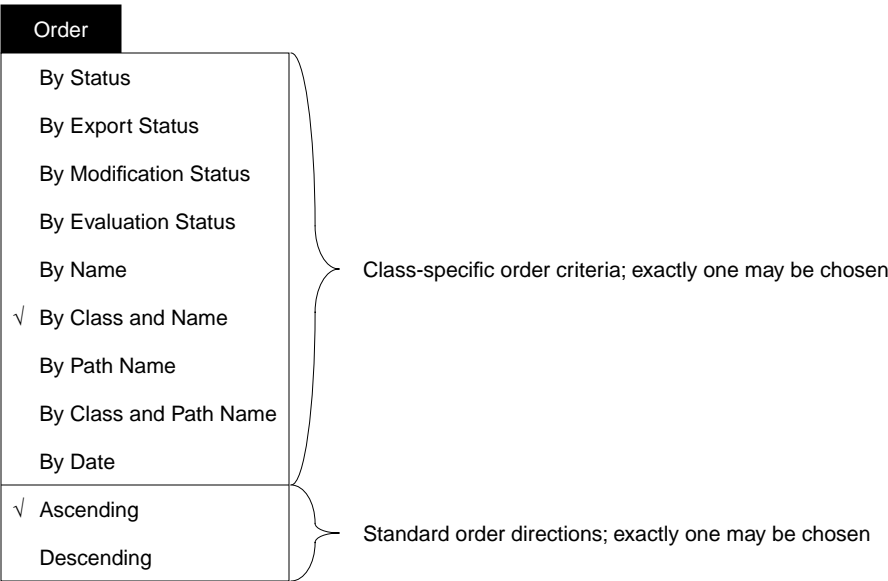
When the user enables filtering, the Search for Items window (page 129) is opened or activated. The user enters search criteria in that window and performs the search; the table is updated to show the search results. When the user disables filtering, the table is updated to show all items in the project.

Order Menu

When either of the table views is chosen, the table-specific Order menu is added to the menu bar. This menu contains a section of class-specific order criteria and the standard section of order directions.

The Order menu lets the user control how to order items in the table when flat table view is chosen and how to order items at the same level of the hierarchy when hierarchical table view is chosen. Ordering by the chosen order criteria is alphabetical and case-insensitive.

Default settings for the order criteria and order direction can be overridden by user preferences. Only one option in each section may be enabled; choosing an option from this menu disables the other options in the same section.



The class-specific order criteria are:

- By Status** Orders items by the symbols in status field. That is: first order by export status; order items of the same export status by modification status; order items of the same export and modification status by evaluation status.
- By Export Status** Orders items by the symbols in export status subcolumn.
- By Modification Status** Orders items by the symbols in modification status subcolumn.
- By Evaluation Status** Orders items by the symbols in evaluation status subcolumn.
- By Name** Orders items by name.

- By Class and Name

Orders items by class and within class by name.
- By Path Name

Orders items by path name. (In hierarchical table view, this is identical to ordering by name because items are already grouped by path.)
- By Class and Path Name

Orders items by class and within class by path name.
- By Date

Orders items by commit date.
- I suspect that the ordering options listed here will be sufficient. If users want more control, we can replace this menu with a dialog box (similar to the Sort dialog box in Excel) that allows the user to sort by a number of different columns.

Project Menu

When the user visits a project in either table view, a Project menu is added to the browser’s menu bar; this menu contains the following entries:

Project		
Add		▶
Add Items from File...		
Duplicate <i>SelectedItems</i>		Shift-Alt-C
Subscribe to <i>SelectedItems</i>		Alt-L
Unsubscribe to <i>SelectedItems</i>		
√ Export <i>SelectedItem</i>		
Rename <i>SelectedItem...</i>		
Delete <i>SelectedItems</i>		Alt-X

- All items in this menu are greyed out when this browser is in read-only mode.

- Add

Adds a new item to the target project (page 70). A user preference controls whether this operation automatically visits the newly created unit. The submenu contains the following entries in alphabetical order:

Subproject

Creates a new, empty subproject named Subproject-*n* and adds it to the target project.

ClassName

Creates a new, empty unit of class *ClassName* named *ClassName-n* and adds it to the target project.

The Add submenu may have additional levels of submenus. For example, instead of listing all diagram classes, it may contain a Diagram submenu of the diagram classes.

As SPECWARE evolves, new classes (possibly including user-defined classes) will be available on the submenu.

After the item is added, its row (and no other) is selected. The name of the new items can be changed in the Name column of this row (see *Editing Text Elements* on page 30) or by choosing Rename *SelectedItem* from the Project menu.

Add Items from File

Reads a text file containing Klang definitions; creates new items corresponding to those definitions, and adds the new items to the target project. Brings up the Open File dialog box (page 142) to prompt the user for the text file. If the user does not select a file of Klang definitions, the operation fails and the Open File dialog box reopens.

After the items are added, their rows (and no others) are selected.

Duplicate *SelectedItem*s

Creates new items by duplicating the items selected in the active area and adds the new items to the target project.

The name of each duplicate item is determined as follows:

- ☐ If the target project does not contain an item of the same class with the original name, the duplicate item has the same name as the original item.
- ☐ If the target project does not contain an item of the same class named *Copy-of-OriginalName*, that name is used.
- ☐ Otherwise, the duplicate is named *Copy-of-OriginalName-n*.

If a single item is selected, that item's class and name appear in the label for this menu entry. Otherwise, the label is Duplicate Selected Items.

This entry is greyed out if nothing is selected in the active area or if the index is active and the current project is selected in the index. (The user cannot duplicate the project into itself or a descendent subproject.)

Subscribe to *SelectedItem*s

Subscribes to the libraries and/or sibling projects that are selected in the index. (The subscribed-to items are added to the target project.) This entry is greyed out unless at least one item is selected in the index and all selected items are either top-level libraries, sublibraries that are exported or re-exported by top-level libraries, or sibling subprojects of the visited subproject.

If a single item is selected, its class and name appear in the label for this menu entry. Otherwise, the label is Subscribe to Selected Items.

Unsubscribe to *SelectedItem*s

Unsubscribes to the items that are selected in the visit area. This entry is greyed out unless at least one item is selected in the visit area and all selected items are subscribed-to libraries or sibling projects.

If a single item is selected, its class and name appear in the label for this menu entry. Otherwise, the label is Unsubscribe to Selected Items.

Export *SelectedItem*

Toggles whether to export the item whose row is selected in the visit area from the subproject that contains it. When the selected item is exported, a checkmark (✓) appears in front of this entry. This entry is greyed out unless a single subproject or read/write unit is selected in the visit area. (A read/write unit is one that is not inside a library.)

-
- Because Export toggles the state, it applies to only one item. If users will want to export lots of items at one time, we could add entries for Export *SelectedItems* and Don't Export *SelectedItems* (either instead of or in addition to the current Export entry).
-

Rename *SelectedItem*

Renames *SelectedItem*; brings up the Rename dialog box (page 151), which prompts the user for a new name for the selected item. If a single read/write item is selected in the visit area, that item's class and name appear in the label for this menu entry. This entry is greyed out unless exactly one item, which is read/write, is selected in the visit area.

If the user enters a name that already exists in the selected item's containing (sub)project, the user is shown an error message window; after clicking OK in that window, the Rename dialog box reopens.

Delete *SelectedItems*

Deletes *SelectedItems*:

- ❑ If a single read/write item is selected in the visit area, that item is deleted; its class and name appear in the label for this menu entry. (A read/write item is a subproject or a unit that is not inside a library.)
- ❑ If more than one read/write items and no read-only items are selected in the visit area, all selected items are deleted; the label is Delete Selected Items.

Deleting a subproject also deletes the subprojects and units it contains.

This entry is greyed out if nothing is selected in the visit area, or if any read-only item is selected in the visit area.

-
- Do we need this entry? It is identical to Cut. I favor leaving it in just to make it clear to new users that they can delete items from the project.
-

Klang Spec

When a browser visits a Klang spec, the user can select any one of the following views:

- ☐ Text (default)
- ☐ Picture
- ☐ Information
- ☐ Present Value

The only change to the browser's standard controls is that the supported views are added to the View menu.

Text View

The text view of a Klang spec is its Klang definition. See *Text View* on page 93 for a description of the editing capabilities available in this view.

Picture View

The picture view of a Klang spec shows the spec as a node in a graph representing a diagram of the category whose nodes are Klang specs and whose arcs are Klang spec morphisms. When a spec's picture view is first displayed, that spec's node is expanded (if possible); all other nodes and arcs that could be expanded are contracted.

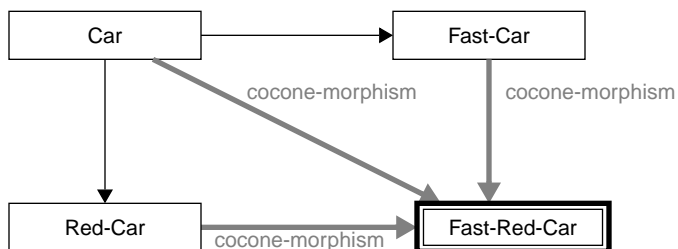
See *Picture View* on page 95 for a description of the editing capabilities available in this view.

As always, if spec's the graph contains any nodes and arcs other than the node for the spec itself, the user can shift-double-click the node or arc to visit the attached unit (which may be an anonymous unit).

✂ I would like to use the term “defining diagram” for the diagram that is displayed for a spec in picture view. Unfortunately, this conflicts with the was that term is currently used for colimit specs. I believe the picture view for a spec should include the visited spec itself, so I have chosen the cocone diagram as the picture view of a colimit spec.

Colimit Specs

Picture view for a colimit spec shows the spec's cocone diagram. For example::



The arcs for the cocone morphism are drawn as heavy grey arrows because the user may not attach different morphisms to those arcs. (See *Drawing Conventions in Graphs* on page 96.)

The colimit spec has a heavy box because it is the spec being visited; its box is double because the node can be expanded to the illustrated graph or contracted to the node by itself.

-
- There is no command to visit the diagram whose colimit is taken to create this spec. To do this, the user could visit the spec in text mode and shift-double-click to visit the diagram. (If the diagram is not named, the user could visit it as described in *Visiting Anonymous Units* on page 32.)
-

Translation Specs

Picture view for a translation spec, has the form:



The arc for the translation morphism is a heavy grey arrow because the user may not attach different morphism to this arcs. Note, however, that the user could shift-double-click the arrow (or its label) to visit (and possibly edit) the translation morphism.

As always visited spec has a heavy box; its box is double because the node can be expanded to the illustrated graph or contracted to the node by itself.

Import Specs

If the visited spec imports a different spec, its picture view has the form:



The arc for the import morphism is a heavy grey arrow because the user may not attach different morphism to this arcs. As always visited spec has a heavy box; its box is double because the node can be expanded to the illustrated graph or contracted to the node by itself.

If the imported spec is a translation or colimit spec (or other spec that can be expanded), its node also has a double box. For example:



Other Specs

If the visited spec is not a colimit or translation spec and does not import another spec, its picture view shows only the node for the spec itself. For example:



As always visited spec has a heavy box; its box is drawn with a single line because the node cannot be expanded to a more complex graph.

Defining a New Spec

The picture view of a newly created empty spec, looks like the picture view for a spec that is not a colimit or translation spec and does not import another spec.

Define an Import Spec. To make the new spec into an import spec, the user can add a node for the imported spec and the arc for the import morphism.

Define a Translation Spec: To make the new spec into a translation spec, the user can:

- ☐ Add a node for the translated spec and the arc for the translation morphism.
- ☐ Visit the translation morphism and set its mappings.

Define a Colimit Spec: To make the new spec into a colimit spec, the user can:

- ☐ Add a diagram in either of the following ways:
 - ☐ Copy and paste an existing diagram.
 - ☐ Drag an existing diagram from the index.
 - ☐ Create a new (anonymous) diagram by adding its nodes and arcs.
- ☐ Use the Graph Tools Palette (page 123) or equivalent entries in the Graph menu (page 104) to:
 - ☐ Take the diagram's colimit.
 - ☐ Merge the node for the colimit spec (added by the colimit operation) with the node for the spec being defined.

Information View

The information view of a Klang spec contains statistics and user notes about the spec. The statistics include the following information about the spec:

- ☐ Path in the universe hierarchy
- ☐ Class (Is this necessary?)
- ☐ Creation date (Is this necessary?)
- ☐ Date and time of last commit
- ☐ Names of units referenced by this spec.
- ☐ Names of units that reference this spec.

The user can enter and modify notes about the spec as described in *Information View* on page 106.

Present-Value View

The present-value view of a Klang spec is its full Klang definition with references to other units resolved, that is, replaced by the present values of those units. For additional information, see *Present-Value View* on page 109.

Klang Spec Morphism

When a browser visits a Klang spec morphism, the user can select any one of the following views:

- ☐ Mapping table (default)
- ☐ Text
- ☐ Picture
- ☐ Information
- ☐ Present Value

The following changes are made to the browser's standard controls:

- ☐ The supported views are added to the View menu (page 89).
- ☐ When the user selects mapping table view:
 - ☐ An entry is added to the Edit menu (page 88).
 - ☐ A section containing the standard table-display options is added to the View menu.
 - ☐ An Order menu (page 89) is added to the menu bar.
 - ☐ A Possible Mappings list (page 90) is added to the visit area to the right of the mappings table.
 - ☐ A region resize bar between the mapping table and the Possible Mappings list allows the user to adjust the relative sizes of the table and list as described in *Resizing Window Areas* on page 31.

If the browser is in read-only mode, the Possible Mappings list and region resize bar are omitted.

Mapping Table View

The mapping table view is a hierarchical table with a few additional controls.

Columns in the table show the element type, the source, and the target to which the source is mapped.

The table heading consists of three rows.

- ☐ The first row is read-only; it contains column titles.
- ☐ The second row contains the source and target spec. In this row, the Source and Target cells can be edited.
- ☐ The third row is read-only. It can contain the following warnings in the Source and Target columns:

Modified	The spec has been modified since it was last committed.
----------	---

Not Parsable	The spec's last committed defining term is syntactically illegal.
Invalid	The spec's last committed defining term is invalid.
Being Evaluated	The spec is currently being evaluated. (Depending on how quick evaluation is, this warning may never appear)

In the body of the table, the container rows correspond to the different element types (sort, operation, axiom, theorem). Noncontainer rows correspond to elements of the source spec. The Type column is filled only in container rows; the Source and Target columns are filled only in noncontainer rows. Only cells in the Target column can be edited.

The ordering options chosen from the Order menu (page 89) control how rows at the same level in the hierarchy are ordered from top to bottom.

For details about working with table views, see *Table Views* on page 109.

➤ The mapping table view provides the functionality of the current morphism editor.

Browser 1 - Morphism Acct-to-Dollar															
File			Edit		Search		Visit		View		Order		Help		
?	C	I	S	F	≧	II	+	X	◀	▶	100 %	▼	z	Z	
						≧	II	+	X	◀	▶	100 %	▼	z	Z
	≧	II	+	X	◀	▶	100 %	▼	z	Z					
	≧	II	+	X	◀	▶	100 %	▼	z	Z					
	≧	II	+	X	◀	▶	100 %	▼	z	Z					
	≧	II	+	X	◀	▶	100 %	▼	z	Z					
	≧	II	+	X	◀	▶	100 %	▼	z	Z					
	≧	II	+	X	◀	▶	100 %	▼	z	Z					
	≧	II	+	X	◀	▶	100 %	▼	z	Z					
	≧	II	+	X	◀	▶	100 %	▼	z	Z					
	≧	II	+	X	◀	▶	100 %	▼	z	Z					
	≧	II	+	X	◀	▶	100 %	▼	z	Z					
	≧	II	+	X	◀	▶	100 %	▼	z	Z					
	≧	II	+	X	◀	▶	100 %	▼	z	Z					
	≧	II	+	X	◀	▶	100 %	▼	z	Z					
	≧	II	+	X	◀	▶	100 %	▼	z	Z					
	≧	II	+	X	◀	▶	100 %	▼	z	Z					
	≧	II	+	X	◀	▶	100 %	▼	z	Z					
	≧	II	+	X	◀	▶	100 %	▼	z	Z					
	≧	II	+	X	◀	▶	100 %	▼	z	Z					
	≧	II	+	X	◀	▶	100 %	▼	z	Z					
	≧	II	+	X	◀	▶	100 %	▼	z	Z					
	≧	II	+	X	◀	▶	100 %	▼	z	Z					
	≧	II	+	X	◀	▶	100 %	▼	z	Z					
	≧	II	+	X	◀	▶	100 %	▼	z	Z					
	≧	II	+	X	◀	▶	100 %	▼	z	Z					
	≧	II	+	X	◀	▶	100 %	▼	z	Z					
	≧	II	+	X	◀	▶	100 %	▼	z	Z					
	≧	II	+	X	◀	▶	100 %	▼	z	Z					
	≧	II	+	X	◀	▶	100 %	▼	z	Z					
	≧	II	+	X	◀	▶	100 %	▼	z	Z					
	≧	II	+	X	◀	▶	100 %	▼	z	Z					
	≧	II	+	X	◀	▶	100 %	▼	z	Z					
	≧	II	+	X	◀	▶	100 %	▼	z	Z					
	≧	II	+	X	◀	▶	100 %	▼	z	Z					
	≧	II	+	X	◀	▶	100 %	▼	z	Z					
	≧	II	+	X	◀	▶	100 %	▼	z	Z					
	≧	II	+	X	◀	▶	100 %	▼	z	Z					
	≧	II	+	X	◀	▶	100 %	▼	z	Z					
	≧	II	+	X	◀	▶	100 %	▼	z	Z					
	≧	II	+	X	◀	▶	100 %	▼	z	Z					
	≧	II	+	X	◀	▶	100 %	▼	z	Z					
	≧	II	+	X	◀	▶	100 %	▼	z	Z					
	≧	II	+	X	◀	▶	100 %	▼	z	Z					
	≧	II	+	X	◀	▶	100 %	▼	z	Z					
	≧	II	+	X	◀	▶	100 %	▼	z	Z					
	≧	II	+	X	◀	▶	100 %	▼	z	Z					
	≧	II	+	X	◀	▶	100 %	▼	z	Z					
	≧	II	+	X	◀	▶	100 %								

Table Header

Visiting a morphism in mapping table view causes the evaluation, if necessary, of its Source and Target specs.

The user can enter or change the morphism's source or target spec in the appropriate cell in the table header by typing, pasting, or dragging from the index and dropping in the cell. If the user changes the source or target spec, the new spec is evaluated, if necessary.

A Warnings cell in the Source or Target column is updated whenever the corresponding spec is committed or evaluated.

- If the Warnings row indicates that either the source spec or target spec is modified, the user may want to visit that spec and commit it before proceeding to define or modify the visited morphism.

Table Body

In the body of the table the Type and Source columns are filled in whenever the morphism has a source spec that has a present value.

The Target column is filled in whenever the morphism has a target spec that has a present value.

- ❑ If the morphism is being defined, the Target column contains suggested mappings generated by SPECWARE.
- ❑ If the morphism already exists, the Target column contains the mappings specified in the morphism's current defining term. (If the source and/or target specs have changed, some these mappings may be illegal.) If the defining term does not contain a mapping from a particular source element, the Target column in that row contains a suggested mapping generated by SPECWARE.

Any suggested mappings are shown in italics; mappings entered by the user are shown in normal font.

The user can enter or modify any target element in the table by typing, pasting, or replacing the existing element with an element in the Possible Mappings list (page 90). If the user deletes an existing target element and does not replace it, it is replaced by a suggested mapping generated by SPECWARE.

Until the user commits the morphism, only mappings entered by the user are included in the morphism's defining term. When the user commits the morphism in mapping table view, any remaining suggested mappings are accepted and become part of the morphism's defining term.

Text View

The text view of a Klang spec morphism is its Klang definition. See *Text View* on page 93 for a description of the editing capabilities available in this view.

Picture View

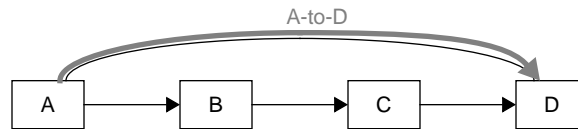
The picture view of a Klang spec morphism shows the morphism as an arc in a graph representing a diagram of the category whose nodes are Klang specs and whose arcs are Klang spec morphisms. The graph contains nodes for the morphism's source and target specs. See *Picture View* on page 95 for a description of the editing capabilities available in this view.

When a morphism's picture view is first displayed, that morphism's arc is expanded (if possible); all other nodes and arcs that could be expanded are contracted.

The arc for the visited morphism is drawn with a heavy grey arrow to indicate that the user may not change the morphism attached to that arc. (See *Drawing Conventions in Graphs* on page 96.)

Composition Morphisms

If the visited morphism is the sequential composition of other morphisms, the expanded picture view contains arcs for the component morphisms and intermediate specs as well. For example:



The visited arc is drawn as a double line to indicate that it can be expanded and contracted. When the visited arc is contracted, the graph appears as follows:



Other Morphisms

If the visited morphism is not a composition, its picture view shows nodes for the source and target specs and an arc for the morphism itself. For example:



As always visited morphism arc is a heavy grey line; it is drawn with a single line because the arc cannot be expanded to a more complex graph.

Defining a New Morphism

In the picture view of a newly created empty morphism, no specs are attached to the node for the source and target. For example:



Information View

The information view of a Klang spec morphism contains statistics and user notes about the morphism. The statistics include the following information about the morphism:

- ☐ Path in the universe hierarchy
- ☐ Class (Is this necessary?)
- ☐ Creation date (Is this necessary?)
- ☐ Date and time of last commit
- ☐ Names of the units referenced by this morphism (its source and target specs).
- ☐ Names of the units that reference this morphism.

The user can enter and modify notes about the morphism as described in *Information View* on page 106.

Present-Value View

The present-value view of a Klang spec morphism is its full Klang definition with references to other units resolved, that is, replaced by the present values of those units. For additional information, see *Present-Value View* on page 109.

Edit Menu

When a Klang spec morphism is visited in mapping table view, an item is added to the Edit menu:

Edit	
Undo <i>Command</i>	Alt-Z
Redo <i>Command</i>	Alt-Y
Cut	Alt-X
Copy	Alt-C
Paste	Alt-V
Select All	Alt-A
Replace Mapping	

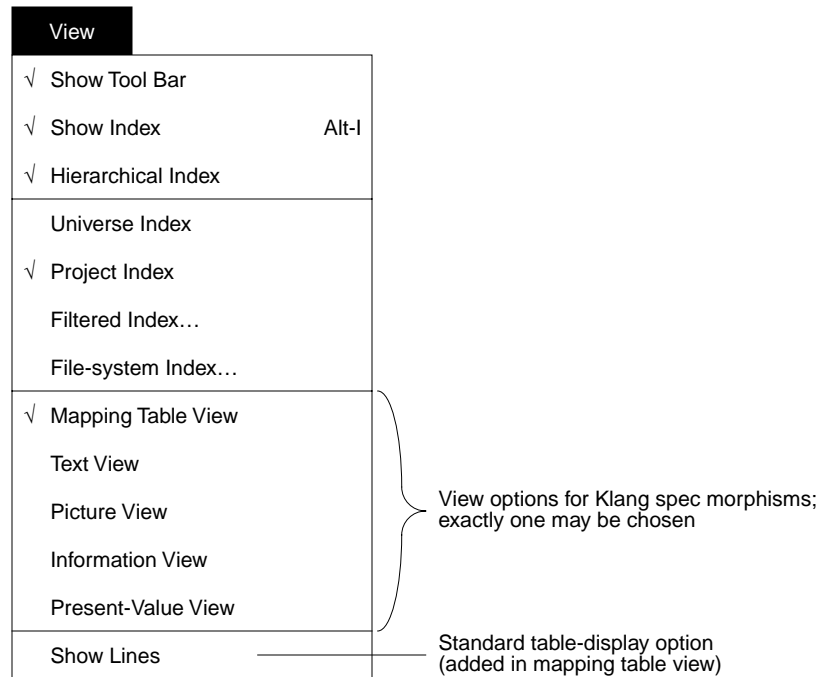
Class- and view-specific entry

Replace Mapping

Replaces the target element in the active row by the element selected in the Possible Mappings list. This entry is greyed out unless a target element is selected in the Possible Mappings list.

View Menu

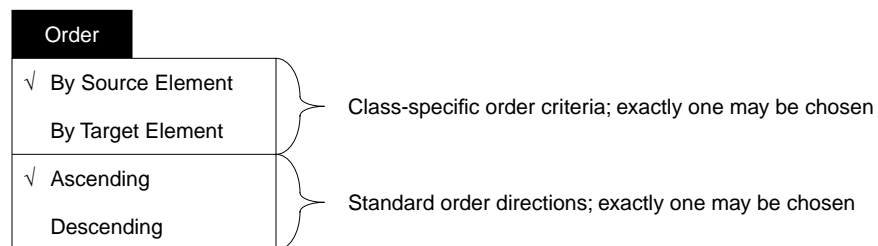
When a Klang spec morphism is visited, class-specific options appear in the view options section. When the mapping table view is chosen, a section containing the standard table-display option is added to the View menu.



Order Menu

When a Klang spec morphism is visited in mapping table view, an Order menu is added to the menu bar. This menu contains a section of class-specific order criteria and the standard section of order directions.

Checkmarks indicate the default settings for the order options. Only one option in each section may be enabled; when you choose an option from this menu, you enable it and disable the other options in the same section.



The class-specific order criteria are:

By Source Element

Orders mapping rows of the same type by the source element.

By Target Element

Orders mapping rows of the same type by the target element.

Possible Mappings List

The Possible Mappings list is used when the morphism has source and target spec both of which have present values. When the user selects a cell in the body of the table, the Possible Mappings list is filled with all the target elements that could be mapped to the row's source element. The user can click a target element in the list to select it or double-click a target element to use it in the row's Target column.

If an element is selected in the Possible Mappings list, the user can replace the existing element with the selected element by choosing Replace Mapping from the Edit menu.

Diagram Classes

When visiting a diagram of any class, the user can select any one of the following views:

- ☐ Picture (default)
- ☐ Text
- ☐ Information
- ☐ Present Value

The only change to the browser's standard controls is that the supported views are added to the View menu.

➤ Following our convention that the user should be able to learn what operations can be performed in a window by looking at its menus, I think an entry **Generate Code** should be added to the File menu when this operation is appropriate. This item would bring up the Write to File dialog box (page 159) to prompt the user for the name of the resulting file.

Text View

The text view of a diagram is its Klang definition. See *Text View* on page 93 for a description of the editing capabilities available in this view.

Picture View

The picture view of a diagram is a two-dimensional graph of the nodes and arcs that constitute the diagram. See *Picture View* on page 95 for a description of the editing capabilities available in this view.

Information View

The information view of a diagram contains statistics and user notes about the morphism. The statistics include the following information about the morphism:

- ☐ Path in the universe hierarchy
- ☐ Category name (Is this necessary?)
- ☐ Class of units for nodes in the diagram
- ☐ Class of units for arcs in the diagram
- ☐ Creation date (Is this necessary?)
- ☐ Date and time of last commit
- ☐ Names of the units referenced by this diagram (attached to its nodes and arcs).
- ☐ Names of the units that reference this diagram.

The user can enter and modify notes about the diagram as described in *Information View* on page 106.

Present-Value View

The present-value view of a diagram is its full Klang definition with references to other units resolved, that is, replaced by the present values of those units. For additional information, see *Present-Value View* on page 109.

File Class

When visiting a file, only text view is available. The text view for a file is the content of that file. (Users shouldn't visit non-ASCII files; if they do, they probably won't see anything useful.)

The following changes are made to the File menu:

- ☐ The entry Commit *ItemName* is replaced with Save *FileName*. The file is not updated on disk until the user chooses this entry.
- ☐ The Commit *ItemName* with Dependencies entry is replaced with Save As. This entry brings up the Write to File dialog box (page 159), which allows the user to specify the name of the new file. Save As leaves the visited file unchanged (since the last Save) and continues visiting the new file. The original file is still in the browser's visit history.
- ☐ The entry Write *ItemName* to File is removed from the menu.

6 View-Specific Controls and Behavior in Browsers

This chapter describes the controls that are added to a browser when particular views are chosen in that window. It provides details about the meaning of common actions (such as cut and paste) in the various views:

- ☐ Text view (page 93)
- ☐ Picture view (page 95)
- ☐ Information view (page 106)
- ☐ Present-value view (page 109)
- ☐ Table views (page 109)

-
- When new menus are added to a browser's menu bar, they are added between the View and Help menus so that the Help menu is always the last one on the menu bar.
-

Text View

Text view allows the user to see and edit text, either the text representation of an item (its Klang definition), or the content of a text file. In this view, a Text menu (page 94) is added to the window's menu bar.

Text-Editing Capabilities

Text view supports the standard text-editing operations:

- ☐ Move cursor forward or backward by one character or one word.
- ☐ Move cursor up or down by one line.
- ☐ Delete forward or backward one character, one word, or one line.
- ☐ Transpose two characters, two words, or two lines (on either side of the cursor).
- ☐ Copy selected text.

Text is selected as described in *Selecting in Text Areas* on page 28.

- ☐ Cut selected text.

❑ Paste from clipboard, replacing any selected text.

➤ I have never liked drag and drop in text area. Usually if I do drag text it's by accident. However, we could add drag and drop if others find it useful.

By default, a text view operates in insert mode. That is, if the user places the insertion point before existing text and starts to type, the new text is inserted at the insertion point.

➤ Do we need to support overwrite mode, too? In overwrite mode, if the user places the insertion point (more appropriately called the cursor) before existing text and starts to type, the new text is overwrites the existing text.

Keyboard Equivalents

By default, the keyboard equivalents are the same as are used in Emacs. In addition, Arrow keys can also be used to move the cursor. Home, End, Page Up, Page Down can be used to scroll the visit area. User preferences can reset the keyboard equivalents.

Syntax-Based Editing

By default, if a SPECWARE item (and not a file) is being visited, syntax-based editing is enabled. A user preference can disable this feature.

See paragraph 4.9.3 through 4.9.6 in Lambert's *Design Decisions* for more information about syntax-based editing.

Text Menu

➤ Do we need this menu? Should it contain other entries?

The Text menu contains the following entries

Text	
Apply Fonts	
Check Syntax	
Show Line Numbers	
Reformat	
Go To Line...	Ctrl-G
Go To Last Insertion Point	

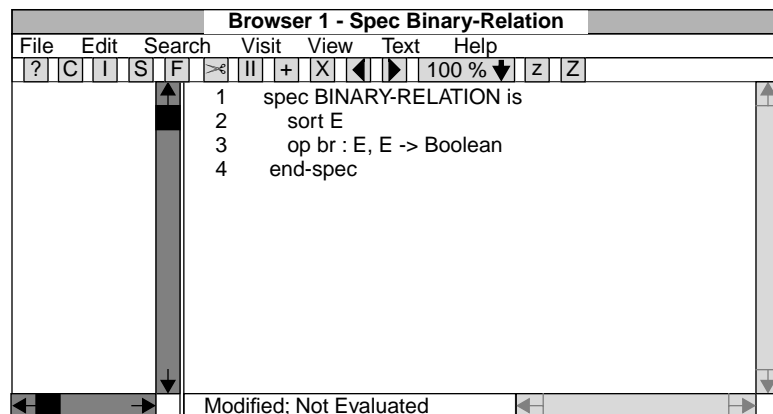
Apply Fonts Applies different fonts and colors to various Klang syntactic entities. This item is greyed out when a file (and not a SPECWARE item) is being visited.

Check Syntax

Checks the Klang syntax of the displayed item. If syntax errors appear, the selection-list window (page 135) is opened or brought forward. Each entry in the selection list contains an error description including the line number where the error occurred; the user can double-click an error description to jump to the line with the error. This entry is greyed out when a file (and not a SPECWARE item) is being visited.

Show Line Numbers

Adds a column of line numbers to the left of the text in the visit area:



Reformat Reformats the text, indenting as appropriate for the Klang syntax. This item is greyed out when a file (and not a SPECWARE item) is being visited.

Go To Line Brings up the Go To Line dialog box (page 140), which lets the user jump to a particular line. Puts the insertion point at the beginning of the indicated line; if the line is not currently visible, scrolls the visit area until the indicated line is close to the top.

Go To Last Insertion Point

Puts the text-insertion bar at the point where text was last inserted. If no text has been inserted since text view was chosen, puts the text-insertion bar at the end of the text. If the new location of the text-insertion bar is not currently visible, scrolls the visit area until the line with the text-insertion bar is close to the top.

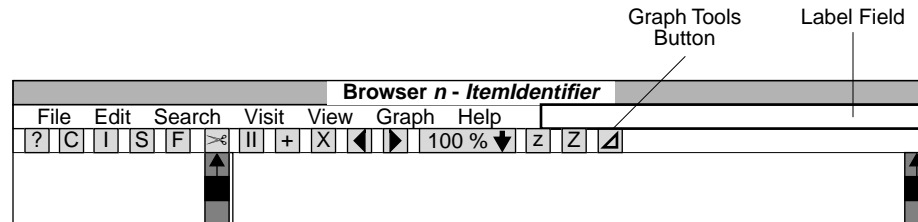
Picture View

Picture view allows the user to see and edit a two-dimensional graph representation of a diagram in some category.

In picture view, the following changes are made to the browser's standard controls:

- ☐ A search command is added to the Search menu (page 102).
- ☐ An index option and a section containing label options are added to the View menu (page 103).
- ☐ A Graph menu (page 104) is added to the menu bar.

- ☐ A Label field (page 105) is added next to the window's menu bar.
- ☐ A Graph Tools button (page 106) is added to the tool bar. The button will have a suitable icon (TBD).



Drawing Conventions in Graphs

The following conventions are used in the graph:

- ☐ A node or arc with no attached unit is drawn with a dotted line.
- ☐ A node or arc with an attached unit is drawn with a solid line.
- ☐ If the visited item is a spec, its node is drawn with a heavier box than other nodes.
- ☐ An arc is drawn with a heavy grey arrow if the user may not change the morphism attached to that arc.
- ☐ If the visited unit is a spec or morphism:
 - ☐ The node for a spec is drawn with a double line if the spec's picture view is more complex than the single node for the spec itself. Such a node can be expanded or contracted within the displayed graph.
 - ☐ The arc for a morphism is drawn with a double line if the morphism's defining diagram is more complex than the arc between its source and target specs. Such an arc can be expanded or contracted within the displayed graph.

➤ These conventions are subject to change; they were chosen so that the differences would be apparent in this black-and-white document. The important issues are to have different renderings for an element with no attached unit, an element with an attached unit, the unit being visited (if applicable), an element whose attached unit cannot be changed, and an element that can be expanded or contracted.

Layout

The layout algorithm tries to layout the graph so that the majority of arcs are either horizontal or vertical. When the user adds a node to the graph, it may be "snapped" to a nearby location to align it horizontally or vertically with an existing node.

Labels

Each node or arc in the graph has a *label* that identifies the node or arc and describes its attached unit or term (anonymous unit).

Syntax for Labels

The label for a node or arc has one of the forms:

- ☐ *ID:Name* where *Name* is the name of the attached unit and *ID* is a token that uniquely identifies the node or arc within the graph.
- ☐ *Name* where *Name* is the name of the attached unit; this is a shorthand for *Name:Name*; that is, the label's ID is *Name*.
- ☐ *Name:Name* where *Name* is the name of the attached unit.
- ☐ *ID:Term* where *Term* is the defining term of the anonymous unit attached to the node or arc and *ID* is a token that uniquely identifies the node or arc within the graph.
- ☐ *ID*: where *ID* is a token that uniquely identifies the node or arc within the graph; this form indicates that the node or arc does not have an attached unit (or term).
- ☐ *MorphismOp* where *MorphismOp* is the Klang token that identifies the morphism operation used to construct the morphism attached to the labeled arc. One of: identity-morphism, translation-morphism, import-morphism, cocone-morphism, inferred-morphism
- ☐ *ID:MorphismOp* where *MorphismOp* identifies a morphism operation and *ID* is a token that uniquely identifies the arc within the graph.

The nodes in a graph must all have different IDs in their labels and the arcs in the graph must all have different IDs in their labels. Although the same ID could be used for both a node and an arc, it is more typical for all IDs used in a graph to be unique.

SPECWARE-Supplied Label IDs

In either picture or text view the user may:

- ☐ Enter the defining term for the anonymous unit attached to an element without giving a label ID.
- ☐ Attach the same (named) unit to two or more different elements without giving label IDs to disambiguate those elements.
- ☐ Give different nodes the same label.
- ☐ Give different arcs the same label.

In these situations, SPECWARE generates unique label IDs automatically when labels are displayed in picture view, when the user switches from picture view to text view, when SPECWARE generates the Klang definition from a graph constructed in picture view, or when the user commits the visited item. Generated label IDs for nodes are typically single letters A, B, C, and so on. (The user who does not like the system's choice of label ID can always change it.) Generated label IDs for arcs are typically composed of the label IDs for the source and target nodes, for example AB, BC.

-
- ✂ Lambert's *Design Decisions* paragraph 4.10.15 says that the SPECWARE will supply the label ID for a proper term in picture view. Is there any reason not to extend this to text view as well? That is, let the user enter just the term and have SPECWARE add the label ID when the user commits or displays
-

Viewing Labels

Label options in the View menu (page 103) control whether the labels of nodes or arcs are displayed in the graph. When displayed, a node's label appears in the node's box; an arc's label appears next to the arc's arrow.

The user can see the label of a node or arc in the following ways (independent of whether labels are displayed).

- ☐ If a single node or arc is selected, its label appears in the Label field (page 105).
- ☐ If the mouse pointer lingers over a node or arc for a short time, a small box pops up next to the mouse pointer showing the label of that node or arc. This feature allows the user to see the entire label, which may be too long to be readable in the Label field.

➤ Do we need a way to selectively show labels on some (but not all) nodes or arcs?

Displaying Labels on page 167 describes how a label of each form appears in the graph and in the Label field.

Modifying Labels in the Graph

When labels are displayed in the graph, the user can select a label and then modify it as described in *Editing Text Elements* on page 30. *Editing Labels* on page 168 describes the effect of replacing a syntactically legal label with a different syntactically legal label.

Selection Operations

In picture view, the selectable elements are the nodes and arcs of the graph. Elements can be selected as described in *Selecting in Non-Text Areas* on page 28.

Cut and Paste Operations

The user can cut or paste nodes, arcs, subgraphs, or the entire graph.

Cutting

The user can cut any collection of nodes and arcs.

- ☐ When the user cuts a node, that node and all arcs from or to the node are removed from the graph.
- ☐ When the user cuts an arc, only that arc is removed from the graph.

Pasting Nodes

The user can paste a node when the last cut or copied element is any of the following:

- ☐ A unit that belongs to (or can be coerced to) the node class of the category of the diagram in the visit area.
- ☐ Text that is the name of a unit that belongs to (or can be coerced to) the node class; the named unit is pasted.

- ❑ Text that is a legal defining term for a unit that belongs to (or can be coerced to) the node class; an anonymous unit with that defining term is pasted.

If nothing is selected when the Paste command is issued, the shadow of a node starts following the mouse pointer; the user clicks the mouse button to drop the node. The dropped node has that pasted unit attached.

If anything is selected, the selection is replaced by a node with the pasted unit attached; see *Replacing Subgraphs* on page 99.

Pasting Arcs

The user can paste an arc when the last cut or copied element is any of the following:

- ❑ A unit that belongs to (or can be coerced to) the arc class of the category of the diagram in the visit area.
- ❑ Text that is the name of a unit that belongs to (or can be coerced to) the arc class; the named unit is pasted.
- ❑ Text that is a legal defining term for a unit that belongs to (or can be coerced to) the arc class; an anonymous unit with that defining term is pasted.

If an arc is selected when the Paste command is issued, the pasted unit is attached to that arc, replacing any unit that was attached to the arc.

Otherwise, a subgraph that consists of two nodes connected by an arc is added to the visit area. The pasted unit is attached to the arc. If that unit's defining term specifies its source unit, that source unit is attached to the arc's source node. If the pasted unit's defining term specifies its target unit, that target unit is attached to the arc's target node. If nothing is selected, a shadow of the subgraph starts following the mouse pointer; the user clicks the mouse button to drop the subgraph. If anything is selected, the selection is replaced by the subgraph as described in arc *Replacing Subgraphs* on page 99.

Pasting Graphs

The user can paste a graph when the last cut or copied element is any of the following:

- ❑ A diagram of (or that can be coerced to) the same category as the diagram in the visit area.
- ❑ The entire graph or a subgraph cut or copied from a picture view showing a diagram of (or that can be coerced to) the same category as the diagram in the visit area.

If nothing is selected when the Paste command is issued, the shadow of a graph starts following the mouse pointer; the user can drop shadow just as in a drag-and-drop operation.

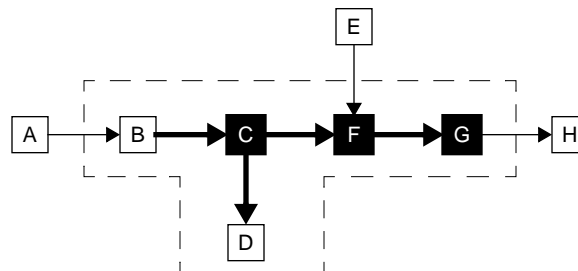
If anything is selected, the selection is replaced by the pasted graph; see *Replacing Subgraphs* on page 99.

Replacing Subgraphs

If one or more nodes and arcs are selected when a graph is pasted, the closure of the selection is replaced by the closure of the nodes and arcs to be pasted.

- ❑ The closure of the selected nodes and arcs is the smallest subgraph of displayed graph that contains all selected nodes and arcs. It is the union of the selected nodes and arcs and the source and target nodes of all selected arcs. This subgraph is called the *original subgraph*.

For example, in the following graph of a visited unit, the original subgraph is the enclosed by dotted line. Nodes B and D are added to the selected nodes because B is the source of a selected arc and D is the target of a selected arc.



- ☐ If the collection of the nodes and arcs to be pasted includes any arcs without source or target nodes, the closure is constructed by adding the missing source or target nodes. Units are attached to an added source or target if a morphism is attached to the arc and that morphism specifies its source or target, respectively.

The closure of the nodes and arcs to be pasted is called the *replacement graph*.

During the replacement process, each boundary node of the original subgraph is matched with a node of the replacement subgraph. A *boundary node* is any node that is the source or target of some arc that is not in the original subgraph. In the preceding illustration, B and F are boundary nodes because they are targets of arcs not in the original subgraph; G is a boundary node because it is the source of an arc not in the original subgraph.

The Replace Selection with Graph dialog box (page 152) opens to let the user guide the matching. (This dialog box is not opened if automatic matching is enabled, and all boundary nodes in the original subgraph have unique compatible nodes in the replacement graph.)

-
- See the appendix on graph-replacement matching in Lambert's *Design Decisions* for a description of the algorithm that is used for automatic matching.
-

Drag-and-Drop Operations

The user can drag and drop to move or copy parts of the graph, to add to the graph, and to make arcs curve.

Moving Parts of the Graph

The user can move individual nodes and subgraphs with the following steps:

- ☐ Selected nodes or subgraphs to be moved.
- ☐ Point to any selected element, and begin the drag operation. (Do not click the mouse button; that action deselects the element.)

- ❑ Drag the selection and drop in the desired location. Any arcs between nodes that are not moved and nodes being moved are redrawn continually as the selection is moved.

As always, dropping outside the visit area or at or near the original location cancels the operation.

Copying Parts of the Graph

If the user depresses the Ctrl key while moving a node or subgraph, the selected node or subgraph is copied and the copy is moved to the desired location. Only the arcs in the selection are copied, not arcs between nodes that weren't copied and nodes that were copied.

Adding to the Graph

The user can drag a unit from the index and drop it in the graph to add it to the graph. The dragged unit can be:

- ❑ A unit that belongs to (or can be coerced to) the node class of the category.
- ❑ A unit that belongs to (or can be coerced to) the arc class of the category.
- ❑ A diagram of (or that can be coerced to) the same category as the diagram in the visit area.

Dropping a dragged unit in the visit area is identical to copying it from the index and pasting it in the visit area. Dropping the unit on a node or arc is equivalent to selecting that node or arc and pasting the unit. Dropping the unit in the visit area not on a node or arc is equivalent to pasting the unit when nothing is selected in the visit area.

As usual, dropping outside the visit area cancels the drag-and drop operation.

Making Arcs Curve

The user can point to an arc and drag. The end-points of the arc will remain attached to the source and target nodes. The point of the arc under the mouse pointer will move with the mouse pointer as the user drags. A smooth curve is fitted from the source node, through the dragged point, to the target node. The user can drop the arc to curve the arc through the desired point. As usual, dropping outside the visit area or close to the original arc cancels the operation.

Search Menu

When picture view is selected, a search command is added to the search menu.

Search	
Find/Replace...	Alt-F
Find Next	Alt-G
Find First	
Find Selection	
Display Selection	
Find/Replace in Project...	Shift-Alt-F
Search for Items...	Shift-Alt-S
Search for Morphisms...	Alt-M

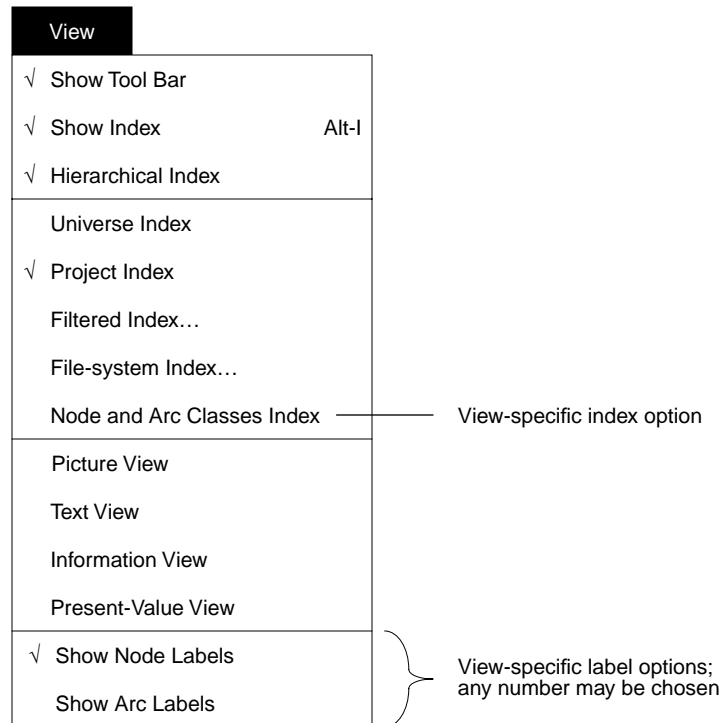
View-specific search command

Search for Morphisms

Opens a Search for Morphisms window (page 132), which allows the user to search for morphism that can be attached to arcs in the graph of the visited item.

View Menu

When picture view is selected, an index option and a section containing label options is added to the view menu. Checkmarks indicate the default settings for label options; these defaults can be overridden by user preferences.



View-Specific Index Option

Node and Arc Classes Index

Sets the index to a list of the units that can be attached to nodes or arcs in the graph. The index contains those units that belong to the node class or the arc class for the category of diagram being edited and that are accessible to the visited unit (see *Accessibility of Units* on page 15). The index also includes any morphism operations supported by the category (like identity-morphism, translation-morphism, import-morphism, cocone-morphism, inferred-morphism). When the index is shown as a hierarchical list, these morphism operations appear at the top level of the hierarchy.

Label Options

Show Node Labels

Toggles between showing and hiding labels on all nodes in the graph. When node labels are being shown, a checkmark (✓) appears in front of this entry.

Show Arc Labels

Toggles between showing and hiding labels on all arcs in the graph. When arc labels are being shown, a checkmark (✓) appears in front of this entry.

Graph Menu

The Graph menu contains the following entries:

Graph	
New Node	Shift-Alt-N
New Arc	Shift-Alt-A
New Colimit	
Sequential Composition	
Merge Selected Nodes	
Expand Defining Diagram	Alt +
Contract Defining Diagram	Alt -
Show Details	Shift-Alt +
Hide Details	Shift-Alt -
Neaten Graph	

The entries in the Graph menu correspond to operations that can be performed using the Graph Tools palette. For a detailed description of these operations, see *Graph Tools Palette* on page 123.

➤ All entries in this graph are greyed out if this browser is in read-only mode.

Adding Nodes and Arcs

- New Node** Creates a new node.
- New Arc** Creates a new arc.
- New Colimit** Creates a new colimit from the selected subgraph; adds the new colimit node and the cocone arcs to the graph. This entry is greyed out if nothing is selected.
- Sequential Composition**
Composes the selected arcs sequentially; adds the corresponding arc to the graph. This entry is greyed out if no arcs are selected or if the selected arcs do not form a path.
- Merge Selected Nodes**
Replaces the selected nodes with a single node. This entry is greyed out if no nodes are selected or if the selected nodes are not compatible.

Expanding and Contracting Subgraphs

The commands that expand and contract subgraphs are available only in the defining diagrams of the visited unit (for example, a spec or a morphism). These entries are greyed out when the visited unit is a diagram.

Expand Defining Diagram

Expands the selected node or arc to show the defining diagram for its attached unit. This entry is greyed out unless a single node or arc is selected and that element's attached unit has a nontrivial defining diagram that is currently contracted.

Contract Defining Diagram

Contracts the defining diagram of the unit attached to the selected node or arc to show only the selected node or arc. This entry is greyed out unless a single node or arc is selected and that element's attached unit has a nontrivial defining diagram that is currently expanded.

Showing and Hiding Details

The commands that show and hide details are available only in the categories in which nodes are themselves diagrams. These entries are greyed out other categories.

Show Details

Shows details of the selected element.

- ☐ If a node is selected, this entry replaces the selected node with a box containing a graph of the diagram attached to the node.
- ☐ If an arc is selected, this entry is greyed out unless a morphism in the category is a collection of morphisms between units attached to nodes in the source and target diagrams. This entry shows details of the arc's source and target nodes (if necessary) and shows the component arcs between nodes in the source and target diagrams.

This entry is greyed out unless a single node or arc is selected whose details are currently hidden.

Hide Details

Hides details of the selected element, replacing details with a single node or arc. If an node is selected, and that node is the source or target of an arc whose details are being shown, the arcs's details are also hidden.

This entry is greyed out unless a single node or arc is selected whose details are currently being shown.

Controlling Layout

Neaten Graph

Makes the alignment and spacing of the nodes in the graph more even so that the majority of its arcs can be horizontal or vertical.

Label Field

When a single node or arc is selected, the Label field shows the label of the selected element.

When a node or arc with no label is selected, the user can enter the label in this field by typing or pasting. The user may enter a label on any of the forms described in *Labels* on page 96. Alternatively, the user may enter just the defining term of the attached anonymous unit and let SPECWARE supply a unique label ID.

When the Label field displays the label of a node or arc, the user can click in that field to place the text insertion bar and edit the existing label. *Editing Labels* on page 168 describes the effect of replacing a syntactically legal label with a different in syntactically legal label.

When no node or arc is selected, the Label field is inactive. That is, the user cannot place the insertion point in the Label field (and so cannot enter text there).

Graph Tools Button

The Graph Tools button opens or brings forward the Graph Tools palette (page 123), which allows the user to add new nodes and arcs and to control the layout of nodes in the graph.

Information View

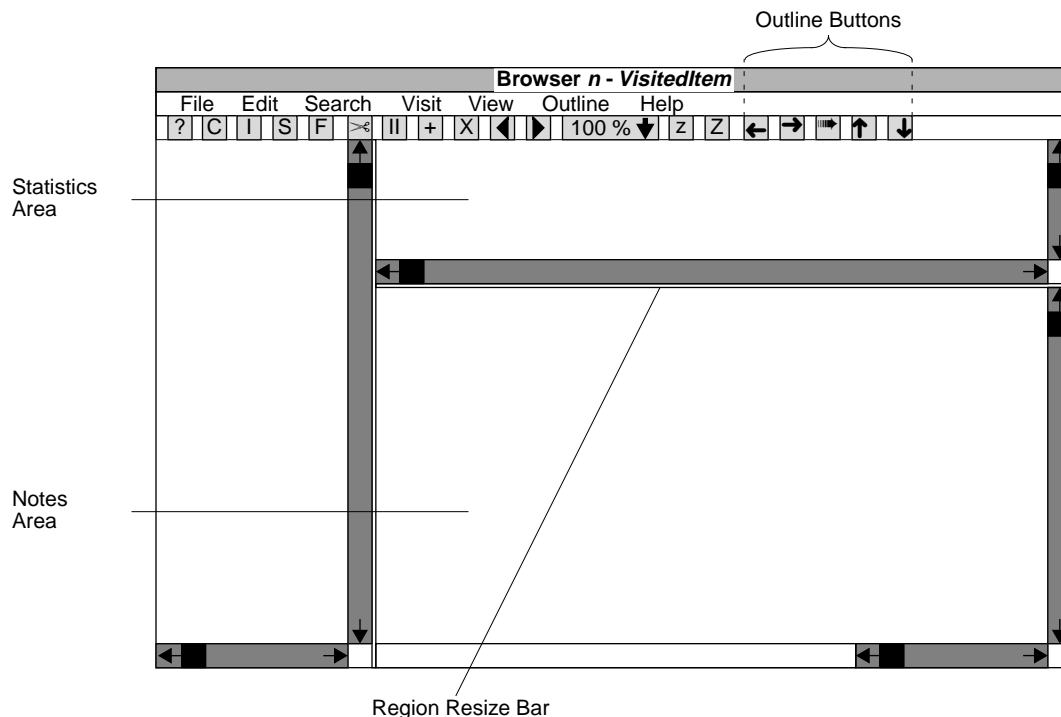
Information view allows the user to see statistics about a unit and to maintain notes about the unit and its development process.

In information view, the following changes are made to the browser's standard controls:

- ☐ An Outline menu (page 108) is added to the menu bar (after any class-specific menus).
- ☐ Outline buttons (page 109) are added to the tool bar.
- ☐ The visit area is divided into two parts:
 - ☐ A read-only statistics area
 - ☐ A read/write notes area containing the user's notes, which are displayed and edited in outline form

A region resize bar marks the movable boundary between the two window areas. The user can change the relative size of the window areas as described in *Resizing Window Areas* on page 31.

-
- For now, assume that statistics area is read-only. Make it editable in a future release if users ask for this change.
-



Text-Editing Capabilities

The information view supports all the standard text-editing capabilities that are available in text view (page 93), but not the syntax-based edition capabilities.

In information view, you can select text as in any text area; see *Selecting in Text Areas* on page 28.

Outline Editing

In information view, each paragraph (identified by an end-of-line character) is considered an element of an outline, either a heading at some level or body text.

- ☐ A heading paragraph is shown in **boldface**; body text is shown in a regular weight font.
- ☐ Each subheading is indented under the heading that contains it.
- ☐ Each paragraph of body text is indented under the heading that contains it.

A triangle appears in the margin of any line that begins a heading paragraph. The triangle behaves like the triangle on a line in an Acrobat Reader index:

- ☐ The triangle points right when the paragraphs in the header's section are not shown; it points down when they are shown.
- ☐ The user clicks the triangle to toggle between showing and hiding the paragraphs in the header's section. Showing the paragraphs is called *expanding* the header to its section; hiding the paragraphs is called *contracting* the section to its header.

The outline-editing commands allow the user to change the level of a heading, to change a heading to body text and vice versa, and to move sections or paragraphs around within the outline. These commands are available through the Outline menu and the tool bar.

Selected Paragraphs

The outline-editing commands act on the “selected paragraphs”, but the selection need not include entire paragraphs. For example, the selection could go from the middle of one paragraph to the middle of another paragraph. The selected paragraphs are defined as follows:

- ☐ If the selection or insertion point is in a body-text paragraph, that paragraph is the (only) selected paragraph.
- ☐ If selection or insertion point is in a heading paragraph whose section is currently expanded, that heading paragraph is the (only) selected paragraph.
- ☐ If the selection or insertion point is in a heading paragraph whose section is currently contracted, that heading and all paragraphs in its section are the selected paragraphs.
- ☐ If the selection spans more than one paragraph, the selected paragraphs are all paragraphs that contain part of the selection. If this includes any contracted headings, those headings and all paragraphs in their sections are selected paragraphs.

Outline Menu

The Outline menu contains the following entries:

Outline	
Higher Level Heading	Shift-Alt ←
Lower Level Heading	Shift-Alt →
Heading to Text	Shift-Alt-T
Move Up	Shift-Alt ↑
Move Down	Shift-Alt ↓
Expand Heading to Section	Alt +
Contract Section to Heading	Alt -

Higher Level Heading

If all selected paragraphs are body-text paragraphs, converts each one to a heading at the next level below the heading that contains it. Otherwise, promotes each selected heading paragraph to the next higher level and leaves all selected body-text paragraphs unchanged.

Lower Level Heading

Demotes each selected heading paragraph to the next lower level heading and leaves all selected body-text paragraphs unchanged.

Heading to Text

Demotes all selected heading paragraphs to body-text paragraphs.

Move Up Moves the selected paragraphs up (before the preceding paragraph).

Move Down Moves the selected paragraphs down (after the following paragraph).

Expand Heading to Section

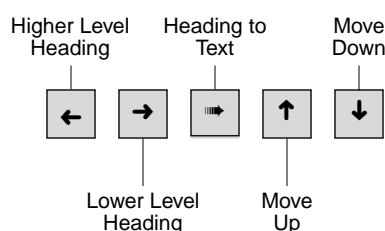
Expands each selected heading paragraph to show its subheadings and text.

Contract Section to Heading

Collapses the section of each selected heading paragraphs to show only the heading.

Outline Buttons

The outline buttons in the tool bar allow the user to perform the same operations at the entries in the Outline menu.



The icons for these buttons are TBD.

Present-Value View

The present-value view allows users to see a unit's present value or the reason why evaluation failed. This view is read only, the only available operations are to select and copy text.

In present-value view, no changes are made to the browser's standard controls, but the Cut and Paste entries of the Edit menu are greyed out.

Showing this view causes evaluation, if necessary, of the visited unit and, recursively, evaluation all units that the visited unit references. While evaluation is in progress, this view displays a message to that effect; any progress messages or warnings produced by evaluation are added to the visit area.

When evaluation terminates, if the unit is valid, any output from the evaluation process is cleared and the unit's present value is displayed. If the unit is invalid, output from the evaluation process remains. This output should include an explanation why the evaluation failed. For example:

*** No value because the referenced spec "WARE" has no value

Table Views

Table views contain a tabular presentation of the visited item. Each row in the table represent an element of the visited item (the meaning of *element* depends on the item's class). Each column contains a class-specific attribute (particular piece of information) for each element of the item.

Currently, the following table views are available:

- ☐ Hierarchical table view for the Project and Library classes.
- ☐ Flat table view for the Project and Library classes.
- ☐ Mapping table view for the Klang Spec Morphism class (a hierarchical table view).

Table views of other classes may be supported in the future.

The following window illustrates a browser visiting a project in hierarchical table view with column-display options status, class, and name; rows are in ascending order by class and name.

Header

Status	Class	Name
R	Library	Relations
▼	M	Project
	V	Diagram
E	Spec	SetOfPairs
	V	Bag
E	Spec	Collection
E	Spec	Sack
E M I	Spec	Seq
E	Spec	Set
	Spec	SetOfPairsColimit
	Spec Morphism	SetToSeq
E R	Project	Misc
▶	Project	Shapes

Current Project; Modified; Version Initial

Table-Specific Controls

In table views, the following changes are made to the browser's standard controls:

- ☐ If the user may select different columns to be included in the table, a section containing column-display options is added to the View menu. The column-display options are class specific; they indicate information that can appear in columns of the table.
- ☐ A section containing table display options is added to the View menus. All classes that support table views include the Show Lines table-display option, which toggles whether to draw a pale horizontal line between each row in the table. Certain classes may have additional table-display options.
- ☐ An Order menu is added to the menu bar. The Order menu contains a section for order criteria entries and a section for order direction entries.
 - ☐ The order criteria entries are class specific; they indicate which column(s) should be used to order the rows of the table.
 - ☐ The order direction for all classes contains two entries: Ascending and Descending.

Table Rows and Columns

The table in the visit area contains at least one header row and one body row for each element of the item being visited.

The table contains one column for each attribute (piece of information about) an element of the visited item. If the table view added column-display options to the View menu, the information columns correspond to the enabled column-display options. A hierarchical table view contains an initial column that indicates whether the row describes a container.

The header remains at the top of the screen when the table is scrolled; only the body rows are scrolled. The first header row contains titles for all information columns (but not for the first column in a hierarchical table). The column used as the order criteria has an underlined title in the header. Some table views include additional class-specific header rows. The first row of the header cannot be edited but editing may be allowed in some cells of class-specific header rows.

Hierarchical Tables

In a hierarchical table, the first column contains an icon indicating whether the row describes a container element (that is, one that contains other elements). This column allows the user to show or hide the contents of a container. A row for a container element has a triangle in this column; a row for a noncontainer element has a blank cell in this column.

Cells in the first column of a hierarchical table are not selectable. Click a cell containing a triangle toggles between showing and hiding the elements in the container.

- ☐ The triangle points right when the container's elements are not shown; it points down when they are shown.
- ☐ Showing element in a container adds rows for those element to the table, directly below the container row. Hiding items in a container removes their rows from the table.

Selecting

The user can select cells or rows of the table as described in *Selecting in Non-Text Areas* on page 28.

In addition, the user can type text to select the first row (from the top of the table) whose ordering column contains that text. For example, suppose the browser is visiting a project in flat table view and the Name field is the ordering criteria. When the visit area is the active area, typing selects the select the first row in the table for an item whose name begins with the typed characters.

- ☐ Pressing Space requires a full-word match; that is, it selects the first row whose ordering column contains exactly the typed letters.
- ☐ Pressing Tab goes to the next row whose ordering column matches the typed letters.

Mouse-Controlled Actions

In the table header, the user can:

- ☐ Click the title of a column to order the rows by that column. For example, in the browser illustrated on page 110, clicking Name would reorder the rows by name (independent of class); they would be displayed in ascending order.
- ☐ Click the title for the column that is the current order criteria to toggle between ascending and descending order.

- ❑ Move the mouse pointer across the header. When the pointer passed the boundary between two columns, it changes from the normal arrow to a resize icon (see *Resizing Window Areas* on page 31). The user can then depress the mouse button and drag in the header to move the boundary between the two adjacent columns, thus resizing them.

Some table views support drag and drop operations; others do not.

Keyboard-Controlled Actions

If a cell of the table is selected, the user can change the selected cell using the arrow keys.

Up arrow (↑)

Selects the cell above the selected cell. This key is disabled when the selected cell is in the first row of the header.

Down arrow (↓)

Selects the cell below the selected cell. This key is disabled when the selected cell is in the last row of the table.

Left arrow (←)

Selects the cell to the left of the selected cell. This key is disabled when the selected cell is in the first column of the table.

Right arrow (→)

Selects the cell to the right of the selected cell. This key is disabled when the selected cell is in the last column of the table.

Tab

Selects the cell to the right of the selected cell. This key is disabled when the selected cell is in the last column of the table.

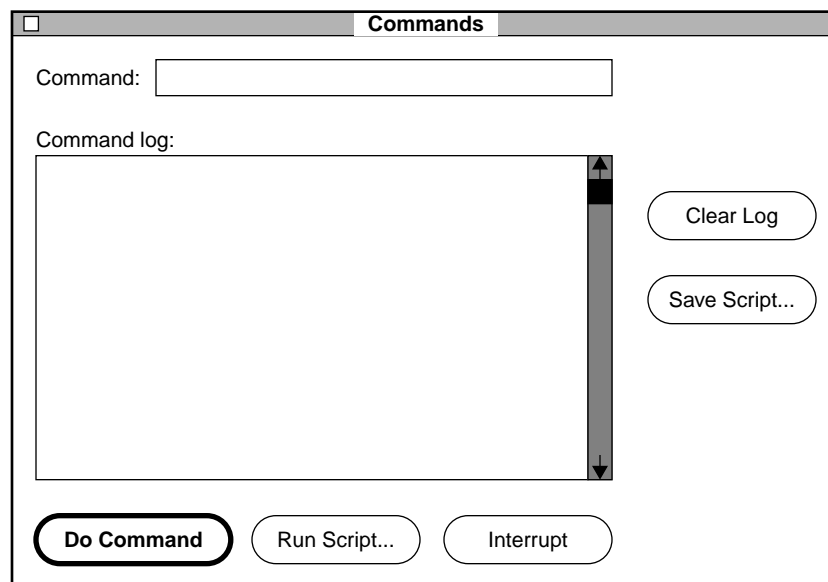
7 Auxiliary Windows

This chapter lists the UI auxiliary windows, that is, windows other than the project window and browsers. The windows are described in alphabetical order.

➤ Perhaps some of these windows are really non-modal dialog boxes.

Command Window

The command window allows the user to issue commands in text form either by typing or by executing a script. A series of commands entered in the window can be saved to a script file for later replay.



-
- The main purpose of this window is to provide an easy way for developers who extend SPECWARE to test SPECWARE features that are not yet ready to be integrated into the interface. Assuming we do that, it might be useful to provide a text form of all commands and the ability to save and rerun a script of commands.
-

Opening the Command Window

The command window is opened when the user chooses Command Window from the Windows menu of the project window. The window remains open until the user clicks its close box. If the command window is already open when the user chooses Command Window, it is brought forward.

Controls

Command Field

The user enters a command to be executed in the Command field, by typing, pasting, or selecting from the command log.

Command Log

The command log is a scrollable read-only text area. Each command that is issued by clicking the Do Command button or by running a script appears in this area in boldface. Any SPECWARE output from the command immediately follows it (in normal weight font).

The user can select and copy in this area. If the user selects a command line (by triple-clicking it), the command is written to the Command field.

Command Buttons

Clear Log Clears the command log.

Save Script Writes the commands in the command log to a script file. Brings up the Write to File dialog box (page 159) to prompt the user for the name of the file to write.

Do Command
Issues the command in the Command field.

Run Script
Runs a sequence of commands from a script file. Brings up the Open File dialog box (page 142) to prompt the user for the script file. As each command in the script file is issued to SPECWARE, it is written to the command log (in bold). Any output from the command is also written to the command log (in normal weight font). When one command finishes executing, the next command is issued.

While a script is running, the Command field and all buttons except Interrupt are greyed out.

Interrupt Asks SPECWARE to interrupt the currently executing command. If the command was part of a script, the remaining commands in the script are not issued. This button is greyed out unless a SPECWARE command is being executed. It is ungreyed when the user clicks the Do Command or Run Script button and it returns to grey when the command or script finishes or when the user clicks Interrupt.

Dependency Windows

A dependency window allows the user to see the items on which a particular item depends and the items that depend on it. Any number of dependency windows can be open at a time; each one shows the dependencies of a different item.

☐ **Dependencies of *ClassName ItemName***

Dependencies of *ItemPathName*

☐ Items that depend on *ClassName ItemName*

☐ Hierarchical List

☐ Items that *ClassName ItemName* depends on

☐ Hierarchical List

☒ Use last committed state of all units
☐ Commit any modified unit

Show Dependencies

Opening a Dependency Window

A dependencies window is opened when the user chooses Show Dependencies from the File menu of a browser. If a dependency window for the visited item already exists, that window is brought forward; otherwise, a new dependency window is opened.

The title of each dependency window identifies the items whose dependencies it shows, which is called the *window's item*. If the mouse pointer lingers over the title for a short time, a small box pops up next to the mouse pointer showing the full path name of the window's item.

Controls

Item Path Name

A label at the top of the window gives the full path name of the item whose dependencies appear in the window. If the full path name is too long to fit in the window, it is truncated on the left and begins with ellipses (...) to indicate that the beginning of the path is not shown. (The user can see all components of the path by pointing to the window title bar.)

Checkboxes and Lists

The user clicks checkboxes to toggle whether to show:

- ☐ Items on which the window's item depends
These items are computed by following References links, starting with the window's item.
- ☐ Items that depend on the window's item
These items are computed by following Referenced By links, starting with the window's item.

Under each checkbox is a list scrollable where the corresponding items are displayed. The lists are blank until the user clicks the Show Dependencies button. Then the list for each enabled checkbox is filled with the corresponding items. Immediately below each list is a the Hierarchical List checkbox, which toggles between displaying a hierarchical list and displaying a flat list. See *Flat and Hierarchical Lists* on page 26.

A list is greyed out when any change is made to the window's item or to one of the items in the lists.

Radio Buttons

Two radio buttons control whether dependencies are computed based on the last committed value of each item encountered while following References and Referenced By links. The user can choose between two options:

- ☐ Use References and Referenced By links based on the last committed value of each item.
- ☐ Commit any modified item and use its updated References and Referenced By links.

Show Dependencies Button

The user clicks the Show Dependencies button to (re)compute and display the selected dependencies. This button is greyed out if the affected lists already show the dependencies and no changes have been made that might invalidate the lists of dependencies.

Find/Replace

The Find/Replace window allows the user to search the text (including text elements in non-text views) in a particular window, possibly replacing some or all occurrences of the search string.

Opening the Find/Replace Window

The Find/Replace window is opened when the user chooses Find/Replace from the Search menu of the project window or a browser. The window remains open until the user clicks its close box.

If this window is open, it is brought forward whenever:

- ☐ The user next chooses Find/Replace from a Search menu.
- ☐ The user performs some standard windowing-system operation to bring it forward (for example, by clicking in the window or its title bar).

Search Area

The *search area* for a Find operation is the window area whose content is searched for the indicated text or pattern.

- ☐ If the window is opened or brought forward from a browser, the search area is the browser's visit area. If the browser is in information view, the search area can be either the statistics area or the notes area, whichever was active.
- ☐ If the window is opened or brought forward from the project window, the search area is that window's active area.
- ☐ If the window was brought forward by some windowing-system operation, the search area is the active area of the most recently active major window (that is, browser or project window).

Controls

Controls in the Find/Replace window describe the search to be performed and the replacement text. When the window opens, the controls are set as of the last search in this session even if the window has been closed since then. Command buttons perform the Find and Replace operations.

Describing the Search and the Replacement Text

The following controls describe the search to be performed:

- ☐ The user enters the search string in the Find field by typing, pasting, or selecting from the Previous Searches pulldown menu.

Space characters in the Find field are significant, including space characters at the beginning and end of the search string.

- ☐ The Whole Word checkbox toggles whether the search string must match whole sequences of whole tokens.

When this checkbox is enabled, the search string must be separated by space characters or other delimiters from any other text on the same line. (The search string itself may include delimiter characters.)

-
- We need to decide what delimiters are used. I assume hyphen is not a delimiter; anything that can be a delimiter in Klang is a delimiter.
-

- ☐ The Regular Expression checkbox toggles whether the search string is text to be matched exactly or a regular expression to be used in pattern-matching search.

- ☐ The Case Sensitive checkbox toggles whether the search is case insensitive (the default) or case sensitive.

Because Klang is case insensitive, the user typically performs a case-insensitive search when the search area contains a Klang definition. However, case-sensitive search might be useful when the user visits a text file or searches a version description or an item's information attribute.

- ☐ The Find Backward checkbox toggles whether to search from the current location forward (the default) or backward.

- ☐ The Wrap-Around Search checkbox toggles whether the search wraps from the end of the search area to the beginning (when searching forward) or from the beginning of the search area to the end (when searching backward).

- ☐ The user enters the replacement text in the Replace With field by typing, pasting, or selecting from the Previous Replacements pulldown menu.

Command Buttons

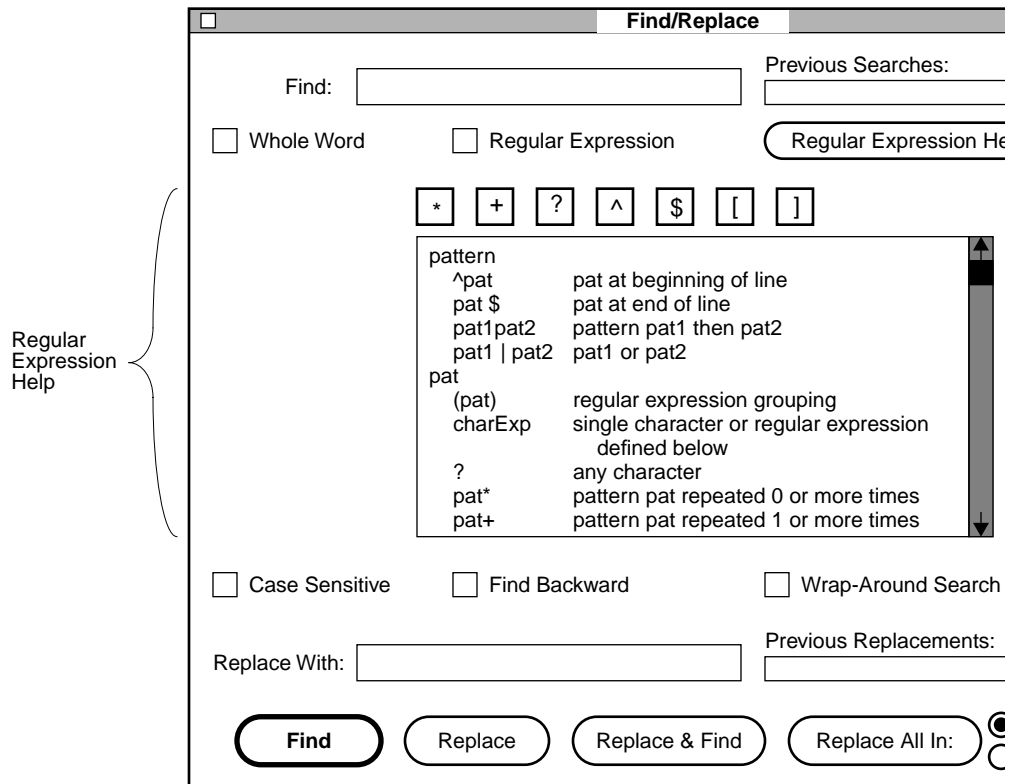
The Find/Replace window contains the following command buttons:

Regular Expression Help

Toggles whether to expand the Find/Replace window to include a scrollable description of the regular expression syntax and buttons for the symbols that

are used in regular expressions. The user can click one of these buttons to insert the indicated character into the Find field (instead of having to search the keyboard for keys that are not used commonly).

The expanded window would look something like the following:



Find Finds and highlights (selects) the next occurrence of the search string. (In a picture view, labels are searched in some arbitrary order.) This button is greyed out when the Find field is blank; otherwise, it is the default button.

Replace Replaces the selection with the replacement text. If the Replace With field is blank, simply deletes the selected text. This button is greyed out if no text is selected or if the Find operation is performed in a read-only area. (Typically the user clicks this button immediately after performing a Find operation.)

Replace & Find Replaces the selection with the replacement text and finds the next occurrence of the search string. This button is greyed out if no text is selected or if the Find operation is performed in a read-only area. (Typically the user clicks this button immediately after performing a Find operation.)

Replace All In Replaces all occurrences of the search string as indicated by the adjacent radio buttons. If the Window Area button is selected, replaces all occurrences in the search area. If the Selection button is selected, replaces all occurrences in the selection; this option does nothing if no text is selected in the search area. This button is greyed out if the Find operation is performed in a read-only area.

If a Find operation fails, a message window opens saying “Not Found”.

Find/Replace in Project

The Find/Replace in Project window allows the user to search all Klang definitions in the current project for a particular string or pattern, possibly replacing some or all occurrences of the search string. Because Klang is case insensitive, the search is case insensitive.

Opening the Find in Project Window

The Find/Replace in Project window is opened when the user chooses Find/Replace in Project from the Search menu of the project window or a browser. The window remains open until the user clicks its close box.

If this window is open, it is brought forward whenever:

- ☐ The user next chooses Find/Replace in Project from a Search menu.
- ☐ The user performs some standard windowing-system operation to bring it forward (for example, by clicking in the window).

Target Browser

Any Find operation performed in this window visits items containing the search string in a browser called the *target browser*.

- ☐ If the window is opened or brought forward from a browser, that browser is the target browser.
- ☐ If the window is opened or brought forward from the project window, a new browser is opened and used as the target browser.
- ☐ If the window was brought forward by some windowing-system operation:
 - ☐ If the most recently active major window is a browser, that browser is the target browser.

- ❑ If the most recently active major window is the project window, a new browser is opened with a hierarchical project index. This new browser is used as the target browser.

Controls

Controls in the Find/Replace in Project window describe the search to be performed, the items to be searched, and the replacement text. When the window opens, the controls are set as of the last search in this session even if the window has been closed since then. Command buttons perform the Find and Replace operations.

Search String

The following controls describe the string to be searched for:

- ❑ The user enters the search string in the Find field by typing, pasting, or selecting from the Previous Searches pulldown menu.
Space characters in the Find field are significant, including space characters at the beginning and end of the search string.
- ❑ The Whole Word checkbox toggles whether the search string must match whole sequences of whole tokens.
When this checkbox is enabled, the search string must be separated by space characters or other delimiters from any other text on the same line. (The search string itself may include delimiter characters.)

✎ We need to decide what delimiters are used. I assume hyphen is not a delimiter; anything that can be a delimiter in Klang is a delimiter.

- ❑ The Regular Expression checkbox toggles whether the search string is text to be matched exactly or a regular expression to be used in pattern-matching search.

Items to be Searched

The following controls describe the *search list*, that is, the list of items whose Klang definitions are to be searched for the indicated string or pattern:

- ❑ The Search Items In pulldown can limit the search to particular subprojects of the current project. The pulldown is hierarchical list of the current project and the items it contains.
 - ❑ If the target browser is visiting a project or library in a table view, the pulldown contains an additional entry, Currently Displayed Rows. The latter entry appears at the top level of the hierarchy (the same level as the current project) and limits the search to the items currently displayed in the table view.
 - ❑ Otherwise, this pulldown contains an additional entry, Index of Browser *n*. This entry appears at the top level of the hierarchy and limits the search to the items currently in the target browser's index. If the browser's index shows the entire universe (or includes items that are not in the project), this entry enables the user to perform the Find operation beyond the boundaries of the current project. Note, however, that the Replace operation can be performed only on items within the current project.

- ☐ The Search Classes pulldown can limit the search to items of a particular class.

Replacement Text

The user enters the replacement text in the Replace With field by typing, pasting, or selecting from the Previous Replacements pulldown menu.

Command Buttons

The Find in Project window contains the following command buttons:

Regular Expression Help

Toggles whether to expand the window to include a scrollable description of the regular expression syntax and buttons for the symbols that are used in regular expressions. See illustration on page 119.

Find First Finds the first item (in some arbitrary order) that matches the search criteria, visits that item in text view in the target browser, and highlights the search string in the browser's visit area. This button is greyed out when the Find field is blank.

Find Next Finds and highlights the next occurrence of the search string, either in the item being visited by the previous Find operation or in the next item that matches the search criteria. This button is greyed out until the user has visited the first item containing the search string, either by clicking the Find First button or by jumping from the Selection-List window (page 135).

List All Finds all items in the search list whose Klang definitions match the search string. Opens or brings forward the list-selection window (page 135) and fills that window with the results of the search. This button is greyed out when the Find field is blank. This is the default button.

Replace Replaces the selection with the replacement text. If the Replace With field is blank, simply deletes the selected text. This button is greyed out if no text is selected or if the visited item is read only. (Typically the user clicks this button immediately after performing a Find operation.)

Replace & Find

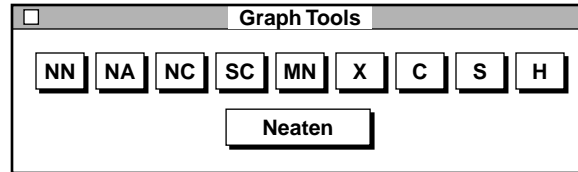
Replaces the selection with the replacement text and finds the next occurrence of the search string, either in the item being visited by the previous Find command or in the next item that matches the search criteria. This button is greyed out if no text is selected or if the visited item is read only. (Typically the user clicks this button immediately after performing a Find operation.)

Replace All In

Replaces all occurrences of the search string as indicated by the adjacent radio buttons. If Current Item is selected, replaces all occurrences in the item currently being visited. If All Searched Items is selected, replaces all occurrences in all read/write items in the search list. The Current Item radio button is greyed out if the visited item is read-only.

Graph Tools Palette

The graph tools palette contains controls for creating and arranging nodes and arcs in a graph. The controls on the palette affect the active browser. If the active window is not a browser or if the active browser is not in picture view, clicking buttons on the palette has no effect.



✂ The buttons in this palette should contain icons; for simplicity I'm currently using initials as button labels.

Buttons on the palette add nodes and arcs to a graph and control the layout of nodes in the graph. All buttons are greyed out when the active browser is in read-only mode.

If the mouse pointer lingers over a button on the palette for a short time, a small box pops up next to the mouse pointer showing the meaning of the button (that is, the command it performs).

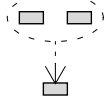
Opening the Graph Tools Palette

The graph tools palette is opened when the user clicks the Graph Tools button in a browser (in picture view). If the palette is already open, clicking the Graph Tools button in a browser brings the palette to the front of the screen.

Changing the Nodes and Arcs

The first five buttons on the palette change the nodes and arcs in the graph. The following table describes the buttons and suggests icons for them.

Label	Icon	Meaning	Changes to Graph
NN		New node	Adds a node
NA		New arc	Adds an arc
NC		New colimit from selected nodes and arcs	Adds colimit node and arcs for cocone morphisms
SC		Compose the selected arcs sequentially	Adds arc for composition of morphisms

Label	Icon	Meaning	Changes to Graph
MN		Merge nodes	Replaces selected nodes with a single node

Common Functionality

After a node or arc is added with any of the buttons in the palette:

- ☐ That item (and no other) is selected in the browser's visit area.
- ☐ The browser's label field is cleared.
- ☐ If labels are being displayed for that kind of item (node or arc), the new item's label is automatically put into text-edit mode (see *Editing Text Elements* on page 30). The user may enter a label for the new item or press Return to exit text-edit mode without giving the item a label.

If the user modifies the label, the modifications are shown both in the label itself and in the browser's label field.

- ☐ If labels are not being displayed, a text insertion bar appears in the (empty) label field. The user can enter a label for the new item (but need not do so). The user can type a label; alternatively, the user can paste in the label field if text or an item of the appropriate class was most recently cut or copied. The label field remains active (accepting keyboard input) until the user presses Return or clicks outside the label field.

NN: New Node

This button creates a new node in the graph. When the button is clicked, the mouse pointer changes to cross-hairs. The user can click a location in the browser visit area to indicate the desired location of the new node. Clicking outside the visit area cancels the operation. The new node appears at or near the specified location. (It may be snapped to align it horizontally or vertically with an existing node.)

NA: New Arc

This button creates a new arc.

- ☐ If exactly two nodes were selected in the browser's visit area when the user clicked this button, the arc is drawn from the node that was selected first to the node selected second.
- ☐ If exactly one node is selected in the browser's visit area, the source (non-arrow) end of the new arc is attached to that node. The mouse pointer changes to cross-hairs. As the user moves the mouse, the target (arrow) end of the arc moves with the cross-hairs. When the user clicks the mouse button inside the visit area, the target end of the arc is positioned at the indicated location. If the user clicks the mouse button outside the visit area, the operation is cancelled.
- ☐ Otherwise, the mouse pointer changes to cross-hairs. When the user clicks the mouse button inside the visit area, the source end of the new arc is placed at the indicated location. As the user moves the mouse, the target end of the arc moves with the cross-hairs. When the user clicks the mouse button inside the visit area a second time,

the target end of the arc is positioned at the indicated location. If the user clicks the mouse button outside the visit area before completing the operation, the operation is cancelled.

When placing the ends of the arc, if the user clicks a node, the arc is connected to that node; if the user clicks a location with no node, a new node is added at that location and the arc is connected to the new node.

NC: New Colimit

This button creates a new colimit node from the selected subgraph; it is greyed out unless at least one node is selected. In addition to the colimit node, cocone morphism arcs are added automatically. Cocone arcs are added from nodes of strongly connected components that are sinks and from leaf nodes of the subgraph. At the end of the operation, only the colimit node itself is selected, not the cocone morphism arcs.

✎ Lambert: Check the phrasing above; I'm not sure I interpreted your comment correctly.

If some nodes and/or arcs don't have attached units, the colimit can still be created. It is defined in Klang as the colimit of an anonymous diagram, which is defined by the Klang equivalent of the selected subgraph.

SC: Sequential Composition of Arcs

This button creates a new arc by sequentially composing the selected arcs; it is greyed out unless:

- ☐ Two or more arcs that can form a path are selected
- ☐ The path does not contain cycles except for self loops (arcs whose source and target are the same).
- ☐ No other arcs are selected.
- ☐ The only selected nodes (if any) are sources or targets of selected arcs.

✎ Notes from Lambert not incorporated into preceding list: Selected arcs could include sequential compositions of parts of the path.

If some nodes and/or arcs don't have attached units, the composition can still be created. It's Klang definition includes the labels of the arcs in the selected subgraph (even if the labels contain only IDs and not names or defining terms).

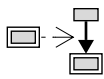
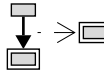
✎ Klang will include a Compose statement to construct a morphism from the sequential composition of other morphisms.

MN: Merge Nodes

This button replaces the selected nodes with a single node; it is greyed out unless two or more compatible nodes are selected. Nodes with attached units are *compatible* if they all have the same attached unit; a node with no attached unit is compatible with any single node and with any collection of compatible nodes.

Expanding and Contracting Subgraphs

The next two buttons expand and contract subgraphs. They are available only in the defining diagrams of the visited unit (for example, a spec or a morphism). These entries are greyed out when the visited unit is a diagram. The following table describes the buttons and suggests icons for them.

Label	Icon	Meaning	Changes to Graph
X		Expand subgraph	Expands a node or arc into the defining diagram for its attached unit.
C		Contract subgraph	Contracts the defining diagram of the unit attached to the selected node or arc to show only the selected node or arc.

X: Expand Subgraph

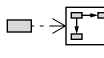
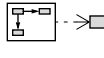
This button expands the selected node or arc to show the defining diagram for its attached unit. This entry is greyed out unless a single node or arc is selected and that element's attached unit has a nontrivial defining diagram that is currently contracted.

C: Contract Subgraph

This button contracts the defining diagram of the unit attached to the selected element. After this operation, only the selected node or arc is shown. This entry is greyed out unless a single node or arc is selected and that element's attached unit has a nontrivial defining diagram that is currently expanded.

Showing and Hiding Details

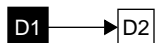
The next two buttons show and hide details of the selected element. They are available only in the categories in which nodes are themselves diagrams; they are greyed out other categories. The following table describes the buttons and suggests icons for them.

Label	Icon	Meaning	Changes to Graph
S		Show details	Show details of the selected node or arc.
H		Hide details	Hide details of the selected node or arc.

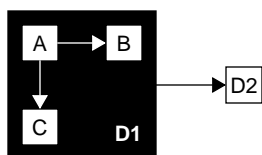
S: Show Details

This button shows details of the selected element; it is greyed out unless a single node or arc is selected whose details are currently hidden.

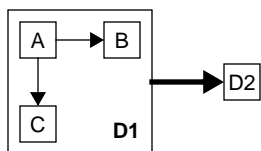
- If a node is selected, this entry replaces the selected node with a box containing a graph of the diagram attached to the node. For example, if the user clicks this button when the visit area of the active browser shows the following graph:



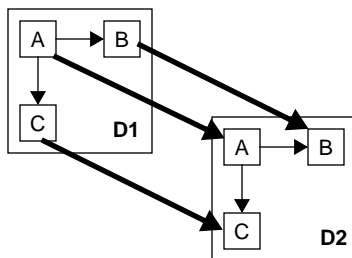
The graph will be modified to appear as follows:



If an arc is selected, this button is greyed out unless a morphism in the category is a collection of morphisms between units attached to nodes in the source and target diagrams. This button shows details of the arc's source and target nodes (if necessary) and shows the component arcs between nodes in the source and target diagrams. For example, suppose the arc is selected in the preceding diagram:



If the user click the Show Details button, the graph will be modified to appear as follows:



When a details are shown, the user cannot select the component nodes or arcs of nodes or arcs in the graph being displayed.

- If the user clicks an element of an internal diagram, the entire diagram node is selected. In the preceding example, if the user clicks the internal node B or the internal arc between A and C, the node D1 is selected.
- If the user selects one of the component arcs of an arc whose details are shown, the latter arc is selected; all of its component arcs are highlighted. In the preceding diagram, the arc between D1 and D2 is selected.

H: Hide Details

This button contracts the defining diagram of the unit attached to the selected element. After this operation, only the selected node or arc is shown. This entry is greyed out unless a single node or arc is selected and that element's attached unit has a nontrivial defining diagram that is currently expanded.

-
- I did not combine Show Details and Hide Details with Expand and Contract, respectively, because I assume that a diagram (for example D1) might have a defining diagram that could be displayed. In that case, both Expand and Show are valid commands, and each has a different meaning.
-

Controlling Layout

The Neaten button makes the alignment and spacing of the nodes in the graph more even so that the majority of its arcs can be horizontal or vertical. If a subgraph is selected, only neatens that subgraph.

-
- Do we need a dialog box to ask the user for how much effort to expend and how far elements may be moved?
-

Help Window

-
- Placeholder
-

Opening the Help Window

The Help window can be opened by choosing SPECWARE Help, Klang Grammar, or Keyboard Shortcuts from the Help menu of the project window or a browser.

Search for Items

The Search for Items window allows the user to enter search criteria for items of interest and to find all items meeting those criteria.

✎ If users find that they want to enter more complex search criteria (constructed with And, Or, and Not), we could extend the functionality of this window in future releases.

Opening the Search for Items Window

The User Can Open the Search for Items window is opened when the user:

- ☐ Chooses Search for Items from a Search menu.
- ☐ Chooses Filtered Index from the View menu of a browser.
- ☐ Enables filtering of a table view in a browser that is visiting a project or library (by toggling Filter Rows in Table in the browser's View menu).

The window title changes to indicate what operation is being performed:

Operation	Window Title
Search for Items	Search for Items
Filtered Index	Set Index of Browser <i>n</i>
Filter Rows in Table	Filter Rows in Table of <i>VisitedItem</i>

Controls

The Search for Items windows has controls to describe the search criteria and to perform the search. When the window opens, the controls are set as of the last similar search or filtering operation in this session even if the window has been closed in the mean time.

Filtering by Hierarchy

The user can limit the search to items in a particular container. To do so, the user selects the desired container from the Items In pulldown.

- ☐ If this window is being used to filter the rows in the table view of a project or library, this pulldown is a hierarchical list of that project or library and the items it contains plus an additional entry, Currently Displayed Rows. The latter entry appears at the top level of the hierarchy (the same level as the visited project or library) and limits the search to the items currently displayed in the table view; it can be used to perform a series of filtering operations in which each operation further filters the result of the previous operation.
- ☐ Otherwise, this pulldown contains a hierarchical list of all available items, with Universe at the top of the hierarchy. If the window was opened or brought forward from a browser, the list contains an additional entry, Index of Browser *n*. This entry appears at the top level of the hierarchy (the same level as Universe) and limits the search to the items currently in the browser's index. When used with the Filtered index option in the browser, this entry can be used to perform a series of filtering operations in which each operation further filters the result of the previous operation.

-
- Should we allow the user to select more than one container? If so, it would mean to find items in any of the selected containers.
-

The first time this window is used to filter the rows of the table view for a particular project or library, this pulldown is set to that project or library; if the user later filters the same project or library, this pulldown is set to the last selected container for that item.

The first time this window is used for any other purpose, this pulldown is set to Universe; thereafter:

- ☐ If the last selected entry is Index of Browser *n*:
 - ☐ If this window was activated by Browser *n*, the pulldown is set to Index of Browser *n*.
 - ☐ Otherwise, the pulldown is set to Universe.
- ☐ If the last selected entry is anything else, the pulldown is set to the last selected entry.

Filtering by Class

The user can limit the search to items of a particular class. To do so, the user selects the desired class from the **Items in Class** pulldown. This pulldown contains at least the names of all supported SPECWARE classes and **Any**; it may contain commonly used combinations of classes.

-
- Should we allow the user to select more than one class? If so, it would mean to find items in any of the selected classes.
-

The first time this window opens in a session, this pulldown is set to **Any**; thereafter, it is set to the last selected class (even if the window has been closed since it was used last).

Filtering by Name

The user can limit the search to items whose names match some search string. To do so, the user enters enables the **With Name** checkbox and enters the search string in the adjacent field. To treat the name as a pattern, the user can click the **Regular Expression** check box.

The user can click the **Regular Expression Help** button to toggle whether to expand the window to include a scrollable description of the regular expression syntax and buttons for symbols that are used in regular expressions. See illustration on page 119.

Filtering by Status

The user can limit the search to items with a particular status. Three groups of radio buttons allow the user to specify the desired export status, modification status, and evaluation status.

Because only units have evaluation status, limitations on evaluation status is ignored when testing whether a given project or library meets the search criteria.

Filtering by Dependency Network

The user can limit the search to items within particular parts of the dependency network. All such filtering follows **References** or **Referenced By** links based on the last committed value of all items.

-
- Should we let the user choose whether to commit modified units and recompute links as in dependency windows (page 115)?
-

Checkboxes allow the user to search for items:

- ☐ That are referenced by a particular item
- ☐ That reference a particular item
- ☐ That depend on a particular item
- ☐ That a particular item depends on

When one of these checkboxes is enabled, the user can select the corresponding item from the adjacent pulldown. Each pulldown contains Any and a hierarchical list of all available items.

Search Button

After entering the search criteria, the user clicks the Search button. If no items meet the search criteria, a message window informs the user of this fact and control returns to the Search for Items window. Otherwise, the resulting items are displayed as follows:

- ☐ If the Search for Items window was just opened or made active because the user chose Filtered Index from the View menu of a browser, the list of items found by the search replaces the index of that browser. If the list contains only one item, that item is displayed automatically in the browser's visit area.
- ☐ If the Search for Items window was just opened or made active because the user chose Filter Rows in Table from the View menu of a browser, the list of items found by the search determines which rows are displayed in the table view of that browser's visited project or library.
- ☐ In all other cases, the list of items is displayed in the selection-list window (page 135).

Search for Morphisms

A Search for Morphisms window allows the user to see the morphisms that can be attached to arcs in the picture view of a particular unit. Any number of Search for Morphism windows can be open at a time; each one finds suitable morphisms for the graph of a particular unit.

☐ **Morphisms Accessible to *ClassName UnitName***

Morphisms accessible to *UnitPathName*

Source

Target

Available Morphisms

☐ Hierarchical List

Show Morphisms

Opening a Search for Morphisms Window

A Search for Morphisms window is opened when the user chooses Search for Morphisms from the Search menu of a browser in picture view. If a Search for Morphisms window for the visited unit already exists, that window is brought forward; otherwise, a new Search for Morphisms window is opened.

The title of each Search for Morphisms window identifies the unit in whose graph the morphisms may be used; that unit is called the *window's unit*. If the mouse pointer lingers over the title for a short time, a small box pops up next to the mouse pointer showing the full path name of the window's unit.

-
- A Search for Morphisms window finds morphisms that are accessible to the window's unit. Those morphisms can be referenced by the window's unit, so they can legally be attached to an arc in the unit's picture view. See *Accessibility of Units* on page 15.
-

Controls

Unit Path Name

A label at the top of the window gives the full path name of the window's unit. If the full path name is too long to fit in the window, it is truncated on the left and begins with ellipses (...) to indicate that the beginning of the path is not shown. (The user can see all components of the path by pointing to the window title bar.)

Source and Target

The source and target pulldowns allow the user to indicate the source and target units for the morphisms. Each pulldown contains:

- ☐ The entry any, which indicates no restriction on the source or target.
- ☐ An alphabetical list of units currently attached to nodes in the unit's diagram.
- ☐ An alphabetical list of units that could be attached to nodes in the diagram (given their class and accessibility).

These pulldowns are initially set based on what is selected in the visit area of the browser from which the window is opened or brought forward. Nodes in the diagram are identified as the source and target for the morphisms of interest. The unit attached to the source node appears in the Source pulldown; the unit attached to the target node appears in the Target pulldown. If a source or target node has no attached unit, any appears in the corresponding pulldown. The source and target nodes are identified as follows:

- ☐ If the selection consists of a single arc and, optionally, either or both of the nodes it joins, the arc's source and target nodes are used to set the pulldowns.
- ☐ If a single node is selected, it is used to set the source pulldown.
- ☐ If two nodes are selected, the node that was selected first is used to set the source pulldown and the node that was selected second is used to set the target pulldown.
- ☐ If the selection does not fall into one of the preceding categories, the source and target pulldowns are set as they were the last time the window was used. Both are set to any if this is the window's first use.

If the user performs some standard window-system operation to bring forward a Search for Morphisms window, the source and target pulldowns are set as they were the last time the window was used.

Morphism List

The scrollable morphism list is initially blank. When the user clicks the Show Morphisms button, accessible morphisms with the indicated source and target units are displayed in the morphism list. The Hierarchical List checkbox toggles between displaying a hierarchical list and displaying a flat list. See *Flat and Hierarchical Lists* on page 26.

If the user changes either the Source or the Target pulldown, the list is cleared.

Selecting from the List. Once a list of morphisms is displayed, the user can click an entry to select the morphism and use the keyboard short-cut to copy it. After doing so, the user could paste the morphism in the diagram.

Updating the Diagram. A short-cut is available when the browser that opened the Search for Morphisms window is still visiting the window's unit in picture view and the selection in the browser's visit area was used to set the Source and Target pulldown. In that case, the user can double-click an entry in the morphism list to use it in the browser's diagram (and return control to the browser).

- ☐ If an arc is selected, the morphism is attached to that arc.
- ☐ If a single node is selected, an arc is drawn from that node to a new node. The morphism is attached to the arc and the morphism's target (if specified) is attached to the new node.
- ☐ If two nodes are selected, an arc is drawn between them (if necessary) and the morphism is attached to that arc.

For example, a user could click an arc in a graph, type Alt-M, click Show Morphisms in the resulting Search for Morphisms window, then double-click a morphism in the list to attach that morphism to the selected arc in the browser.

Visiting from the List. The user can shift-double-click an entry in the list to open a new browser visiting the corresponding morphism.

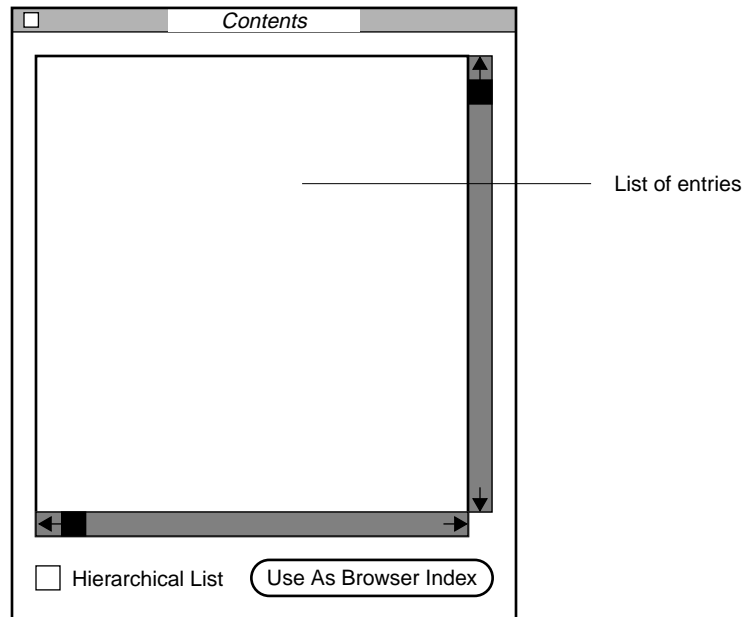
Show Morphisms

The user clicks the Show Morphisms button to search for the accessible morphisms with the indicated source and target units.

As long as the window remains open, the user can enter different source and target units and click Show Morphisms to search for suitable morphisms.

Selection-List Window

The selection-list window displays a scrollable list of entries to the user.



Opening the Selection-List Window

The Selection-List window is opened when the user:

- ☐ Performs a search from the Find in Project window (page 120) that finds more than one item containing the search string or pattern.
- ☐ Performs a search from the Search for Items window (page 129) except when that window is used to create a filtered index for a browser or to filter the rows of a table.
- ☐ Chooses Check Syntax from the Text menu in a browser in text view.

The window remains open until the user clicks its close box. If the window is already open when the user performs an operation that writes to the window, the window is brought forward; the old content of the window is replaced with the entries that result from the current operation. (A user who wants to save the results of the last search can use that last list as the index of a new browser before performing the next search.)

The window title changes to indicate what operation generated the list:

Operation	Window Title
Find in Project	Items Containing Search String
Search for Items	Items Meeting Search Criteria
Check Syntax	Syntax Errors in <i>VisitedItem</i>

Controls

List of Entries

The list of entries displays the results of performing the operation that opened this window or brought it forward. The entries that are results of performing the operation are called *result entries*. The user can double-click an entry in the list to jump to the location associated with that entry. The entries in the list and their associated locations depend on the operation that created the list

Operation	Result Entries	Associated Location
Find/Replace in Project	The items whose Klang definitions include at least one match for the search string.	New browser visiting the relevant item in text view with the first occurrence of the search string selected. (The user can click the Find Next button in the Find/Replace in Project window to jump to the next occurrence.)
Search for Items	The items that satisfy the search criteria.	New browser visiting the item.
Check Syntax	Syntax errors in the Klang definition that was checked.	Line with the error in the same window in which syntax was checked.

When the list entries are items, each item appears as its class, name, and full path name. When the list entries are syntax errors, each entry gives the line number where the error occurred and a brief description of the error.

Hierarchical List Checkbox

The Hierarchical List checkbox toggles between displaying the list of entries as a hierarchical list and displaying them as a flat list.

Use As Browser Index

The Use As Browser Index button opens a new browser in filtered index mode; the browser's index contains the items in the selection list. The index is shown hierarchically if the hierarchical checkbox in this window is enabled.

This feature allows the user to jump to all the items in the same browser instead of opening a new browser for each one. This button is greyed out when the selection-list window was opened by checking syntax (because all items already jump to within one item in one window).

8 Dialog Boxes

Dialog Boxes are opened when the user chooses various menu entries or clicks buttons in various windows. This chapter describes the dialog boxes in alphabetical order.

Create Library

The Create Library dialog box prompts the user for information controlling the creation of a new of a library from the current project.

Create Library from *CurrentProject*

Library Name:

Naming sublibraries:

- ☒ Give each sublibrary same name as the corresponding subproject
- ☐ Add "Library-" to front of the corresponding subproject name
- ☐ Ask for name of each sublibrary

Creating libraries from projects to which *CurrentProject* subscribes:

- ☒ Create top-level libraries from all subscribed-to projects
- ☐ Create sublibraries from all subscribed-to projects
- ☐ Ask about each subscribed-to project

Create **Cancel**

-
- Lambert: Please check this dialog box. I believe it captures the spirit of the proposal in your email of 9 Dec 98 23:09:10.
- Differences:
- I thought it would be less confusing to leave the command meaning “create library from current project” instead of requiring the user to select projects to convert. The Project menu is only present in a table view, so your proposal would preclude creating a library when visiting the project in text view. The entry “Convert <SelectedProjects>” doesn’t make it clear that the current project will be converted along with the selected subscribed-to projects or that there is no need to select subprojects because they are converted automatically in any case.
 - A library will be identified as such by an icon (in hierarchical lists) or by the identifier Library *Name* in flat lists and tables, so the “Library” our default name Library-*ProjectName* is redundant. To save the user from having to rename each library and sublibrary, this dialog box provides a way for the user to specify different names when the libraries are created.
 - Simplifies the interaction: SPECWARE doesn’t have to ask the user about each subscribed-to sibling project individually if all are to be treated the same way.
-

Opening the Create Library Dialog Box

The Create Library dialog box is opened when the user chooses Create Library from the File menu of the project window. The name of the current project is filled into the title of the dialog box and various text labels. It is also used to generate a default library name.

Controls

The Create Library dialog box has a Library Name field, radio buttons that control how sublibraries are named and how subscription between sibling projects is handled, and command buttons.

Library Name Field

When the dialog box opens, the Library Name field is initialized with a name of the form:

Library-*CurrentProject*

The name is selected, so the user can enter a different name by typing or pasting.

Naming Sublibraries

The first group of radio buttons controls how names are assigned to sublibraries corresponding to the subprojects in the hierarchy of the current project. This group is greyed out if the current project has no subprojects.

The user can select from the following options:

- ☐ Give each sublibrary the same name as its corresponding subproject.
- ☐ Give each sublibrary a name of the form Library-*SubprojectName*, where *SubprojectName* is the name of its corresponding subproject.

- ☐ Ask the user for the name for each sublibrary. If the user selects this option, SPECWARE brings up the following dialog box for each subproject in the hierarchy of the current project.

Name Library Created from *Subproject*

Creating sublibrary of *ContainingLibrary* from subproject *Subproject* of *ContainingProject*.

Sublibrary Name:

OK

If the user enters an illegal library name, a message window informs the user of this fact and this dialog box reopens to allow the user the modify the name.

Creating Libraries from Subscribed-To Projects

The second group of radio buttons controls the creation of libraries from any sibling projects to which the current project subscribes. They are greyed out if the current project is a top-level project in its registry or if the current project does not subscribe to any sibling projects.

The user can select from the following options:

- ☐ Create a top-level library from each subscribed-to sibling project. Each top-level library is given the default name *Library-SubscribedToProjectName*.
- ☐ Create a sublibrary from each subscribed-to sibling project. The names of any sublibraries are set as indicated by the first group of radio buttons.
- ☐ Ask the user how to handle each individual subscribed-to sibling project. If the user selects this option, SPECWARE brings up the following dialog box for each subscribed-to project encountered while resolving subscription relationships.

Library Created from *SubscribedToProject*

SubscribingProject subscribes to it sibling project *SubscribedToProject*.

Make the library created from *SubscribedToProject*:

☒ A top-level library

☐ A sublibrary of the library created from *SubscribingProject*

Library Name:

OK

If the user selects the second radio button before modifying the Library Name field and the Create Library dialog box indicates that sublibraries should have the same name as their corresponding subproject, the default name in the Library Name field is automatically changed to *SubscribedToProject*.

If the user enters an illegal library name, a message window informs the user of this fact and this dialog box reopens to allow the user the modify the name.

The selected option is applied recursively. That is, if a subscribed-to sibling project A subscribes to another sibling project B, a library is created from B according to the selected option.

If the user selects Ask about each subscribed-to project, after the user has said that a particular project should be converted to a top-level library, that library is used by any sibling that subscribes to the corresponding project, eliminating the need to ask about the project again. Thus if A subscribes to the current project, the library created from A will subscribe to the library created from the current project. If A is converted to a top-level project and B also subscribes to A, the library created from B will subscribe to the top-level library created from A.

Command Buttons

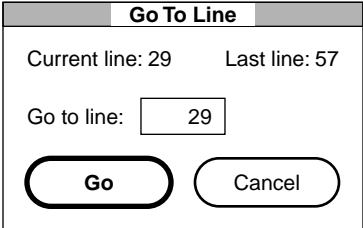
The Create Library dialog box has the following command buttons

- | | |
|--------|--|
| Create | Creates the library and exits the dialog box. This is the default button.

If the a library already exists with the indicated name (or the name is illegal), a message window informs the user of this fact and the Create Library dialog box reopens to allow the user the modify the name. |
| Cancel | Exits the dialog box without creating a library. |

Go To Line

The Go To Line dialog box lets the user enter the line where the text insertion bar should be placed.



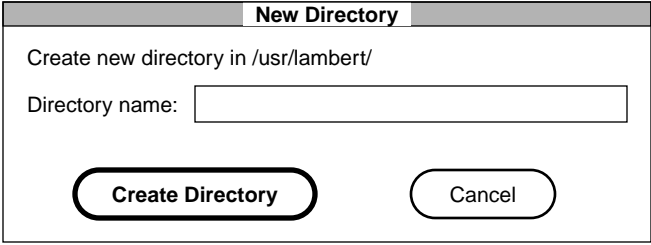
The screenshot shows a dialog box titled "Go To Line". It contains the following elements:

- Title bar: Go To Line
- Text: Current line: 29 Last line: 57
- Text: Go to line:
- Buttons: Go, Cancel

The Go To Line dialog box is opened when the user chooses Go To Line from the Text menu of a browser (in text view).

New Directory

The New Directory dialog box allows the user to enter the name of a new directory to be created.

The dialog box has a title bar with the text "New Directory". Inside, it says "Create new directory in /usr/lambert/". Below this is a label "Directory name:" followed by a text input field. At the bottom, there are two buttons: "Create Directory" and "Cancel".

Create new directory in /usr/lambert/

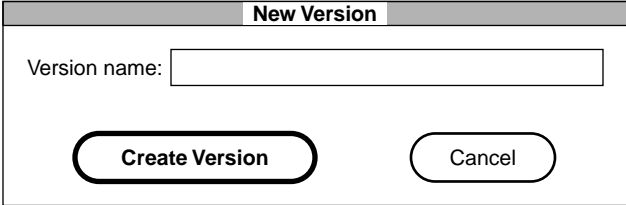
Directory name:

Create Directory Cancel

The New Directory dialog box is opened when the user clicks the New Directory button in the Select Directory dialog box or the Write to File dialog box.

New Version

The New Version dialog box allows the user to enter the name of a new version of the current project.

The dialog box has a title bar with the text "New Version". Inside, it says "Version name:" followed by a text input field. At the bottom, there are two buttons: "Create Version" and "Cancel".

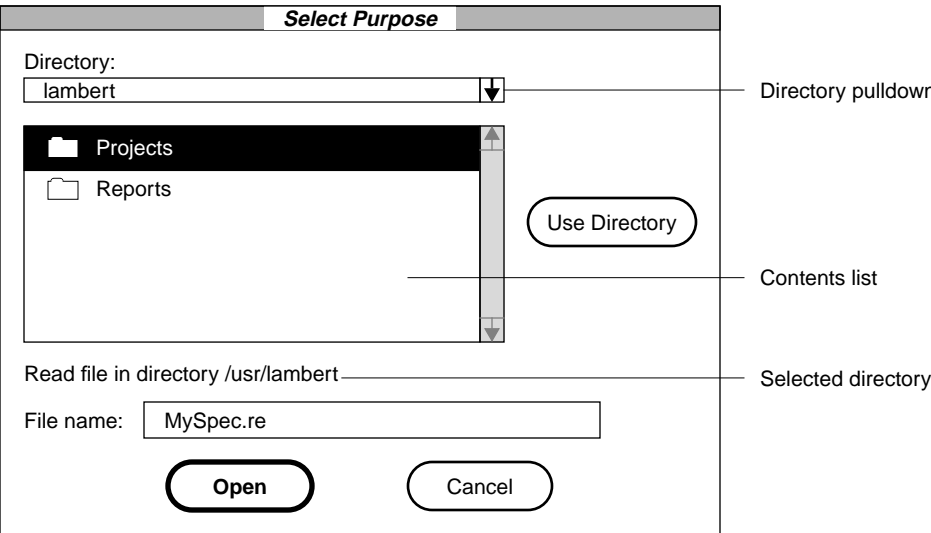
Version name:

Create Version Cancel

The New Version dialog box is opened when the user chooses New Version from the Version menu of the project window.

Open File

The Open File dialog box prompts the user for the name of a file to be opened for some purpose.



Opening the Open File Dialog Box

The Open File dialog box is opened when the user:

- ☐ Chooses New Project from File from the File menu in the project window.
- ☐ Chooses Add Items from File from the Project menu of a browser visiting the current project or one of its subprojects.
- ☐ Clicks the Run Script button in the Command window.

The title of the dialog box indicates the operation that opened it.

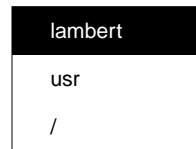
Operation	Title of Dialog Box
New Project from File	Create Project From File
Add Items from File	Add Items Defined in File
Run Script	Run Script in File

Controls

The Open File dialog box has controls that allow the user to navigate through the file-system hierarchy.

Directory Pulldown

The directory pulldown shows the name of the currently selected directory from which the file will be read. Entries in the menu show the names of each directory in the path from the selected directory to the root of the file-system hierarchy. For example:



The full pathname of the selected directory appears below the contents list.

When the dialog box opens for the first time in a given session, the directory pulldown shows the connected directory from which the user started SPECWARE. Thereafter, the pulldown shows the directory selected most recently in this dialog box, the Select Directory dialog box, or the Write to File dialog box. The user can move up in the file-system hierarchy by selecting an entry from the directory pulldown.

Contents List

The contents list shows the subdirectories and files in the directory that appears in the directory pulldown. Names of read-protected files and subdirectories are greyed out in the list. The user can:

- ☐ Click a subdirectory entry to select it.
- ☐ Double click a subdirectory entry to make that subdirectory appear in the directory pulldown, moving down in the file-system hierarchy.
- ☐ Click a file entry to make its name appear in the File Name field.
- ☐ Double click a file entry to select and open that file and exist the dialog box.

Selected Directory

A label between the contents list and the File Name field shows the full path name of the selected directory. If the full path name is too long to fit in the dialog box, it is truncated on the left and begins with ellipses (...) to indicate that the beginning of the path is not shown. (The user can see all components of the path by opening the directory pulldown.)

File Name Field

The File Name field contains the name of the file that will be read from the selected directory. When the dialog box opens, this field is blank.

The user can enter a file name by typing, pasting, or selecting from the contents list.

Command Buttons

The Open File dialog box contains the following command buttons:

Use Directory

If a subdirectory entry is selected in the contents list, makes that subdirectory appear in the directory pulldown, moving down in the file-system hierarchy. Otherwise, this button is greyed out.

Open	Opens the file of the indicated name in the selected directory and exists the dialog box. This is the default button. If the indicated file does not exist or is not of the expected format, a message window informs the user of this fact and reopens the Open File dialog box.
Cancel	Exits the dialog box, cancelling the operation that brought up this dialog box.

Page Setup

The Page Setup dialog box allows the user to set any relevant printing attributes (for example, page orientation, output device, font, scaling).

> Add picture

The Page Setup dialog box is opened when the user chooses Page Setup from the File menu of the project window or a browser.

Preferences

> Add picture.

The Preferences dialog box allows the user to control the behavior of SPECWARE. This window will have tabs for setting:

- ☐ Project window options (page 145)
- ☐ Browser options (page 145)
- ☐ Text format options (page 147)
- ☐ Color use (page 148)
- ☐ Keyboard shortcuts (page 148)
- ☐ Interface options (page 148)
- ☐ SPECWARE options (page 149)
- ☐ Prover options (page 150)
- ☐ Printing options (page 150)
- ☐ Miscellaneous options (page 150)

Opening the Preferences Dialog Box

The Preferences dialog box is opened when the user chooses Preferences from the File menu in the project window.

Project Window Options

✂ Add picture.

Ideas for display options that affect the project window:

- ☐ Whether the tool bar is initially shown when SPECWARE starts.
Default: yes.
- ☐ Which commands have corresponding buttons on the tool bar of the project window.
Default: See *Tool Bar* on page 47.

Browser Options

✂ Add picture.

Ideas for display options that affect browsers:

- ☐ Whether the tool bar is initially shown when a browser opens.
Default: yes.
- ☐ Which commands have corresponding buttons on the tool bar of a browser.
Default: See *Tool Bar* on page 65.
- ☐ Whether the index is initially shown when a browser opens.
Default: yes.
- ☐ Whether the index is initially displayed in hierarchical form when a browser opens.
Default: yes.
- ☐ The initial index option for new browsers (other than browsers opened from the selection-list window).
Default: project index when there is a current project; otherwise, universe index.
- ☐ The initial view when an item of each class is visited in a browser.

Class	Default Initial View
Project	Hierarchical table
Library	Hierarchical table
Any specification class	Text
Any morphism class	Mapping table
Any diagram class	Picture
File	Text

✂ Add other classes to the preceding table as we discuss what views they support.

Project and Library Options

✂ Add picture.

Ideas for display options that affect browsers visiting a project or library:

- ☐ The information columns that are included in either table view.
Default: Status, Class, and Name.
- ☐ The ordering criteria in either table view.
Default: By Class and Name
- ☐ The order direction in either table view.
Default: Ascending
- ☐ Table-display options that are enabled in either table view.
Default: none.

Picture-View Options

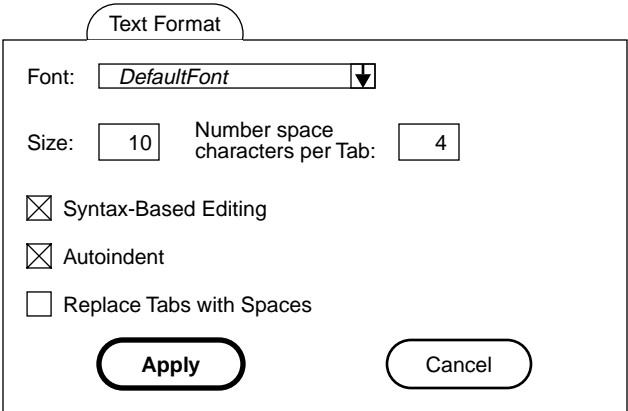
✂ Add picture.

Ideas for display options that affect browsers visiting an item in picture view:

- ☐ Whether node labels are shown initially
Default: yes.
- ☐ Whether arc labels are shown initially
Default: no.
- ☐ Rendering for different kinds of elements.
Default: See *Drawing Conventions in Graphs* on page 96.
- ☐ Whether automatic mapping is enabled for pasting graphs with replacement. If automatic mapping is disabled, the Replace Selection with Graph dialog box is opened for every subgraph replacement operation and the user must indicate match each boundary node of the original subgraph with a node of the pasted graph.
Default: yes (automatic mapping is enabled)

Text Format Options

The Text Format tab allows the user to select formatting options for text views in browsers and to control the font in any text region (including the version description area of the project window or the user notes area in a browser in information view. The following illustration shows the default options.



The Text Format tab contains the following controls:

Font pulldown

The user can choose the font to be used in text areas.

Size field

The user can enter the font size (in points) to be used in text areas.

Tab Size field

The user can enter the size of a tab character, that is, the number of space characters that move text over the same horizontal distance.

Syntax-Based Editing checkbox

Toggles whether to enable syntax-based editing when visiting a SPECWARE item in text view.

Autoindent checkbox

Toggles whether a return character should cause the following line to be autoindented when visiting a SPECWARE item in text view.

Replace Tabs with Spaces checkbox

Toggles whether tab characters should be converted to the equivalent number of space characters.

Apply button

Applies the indicated formatting options.

Cancel button

Removes all changes made in this tab since it was last opened or since the user clicked the Apply button, whichever happened more recently.

Color Use

This tab lets the user control whether items are rendered in different colors depending on their status.

- ☐ Color for committed items
Default: black
- ☐ Color for modified items
Default: black
- ☐ Color for read-only items
Default: black
- ☐ Color for units with a present value
Default: black
- ☐ Color for unparsable units
Default: black
- ☐ Color for invalid units
Default: black
- ☐ Color for units that are being evaluated
Default: black
- ☐ Color for exported items
Default: black
- ☐ Color for items that are not exported
Default: black

➤ Do we want to allow use of colors for other purposes than showing status?

Keyboard Shortcuts

➤ Add picture.

This tab lets the user set the keyboard shortcut for any supported command (even those that don't currently have keyboard shortcuts).

Interface Options

➤ Add picture.

This tab lets the user customize the behavior of various standard GUI controls and operations. For example the user can choose:

- ☐ Whether a pulldown menu:
 - ☐ Only stays open while the mouse button is depressed. (Default)
 - ☐ Opens into a list-like control (possibly including a scroll bar) when the user clicks the menu.
- ☐ Whether the drag operation:
 - ☐ Requires the user to depress the mouse during the drag and release the mouse button to drop. (Default)
 - ☐ Allows the user to signal the start of a drag operation by pressing a “drag prefix”, then drag without the mouse button depressed and click the mouse button to drop.
- ☐ Whether informational boxes pop up automatically when the mouse pointer lingers over various window elements.
Default: Yes
- ☐ Time in milliseconds that the mouse pointer needs to stay over a window element before an informational box pops up.
Default: ???
- ☐ What text editor to use when visiting an item in text view.
Default: The built-in text editor that is integrated with the browser in text view.
A foreign editor can be used to edit a unit only if it has not been modified since it was last committed. If the user goes to text view of a modified unit, a message window appears instructing the user to either commit the unit or roll back (or forward) to a committed state.
Saving the text from within the foreign editor serves as Commit of any changes made during the foreign edit session.
While a foreign edit is in progress, the unit involved is locked into view-only mode for any browser visiting the unit. (Visits will display the unit’s present committed state).
The assistance given by the default editor is obviously lost, as well as continual view updates of any other view on a unit involved in a foreign edit.

✂ Lambert’s *Design Decisions* paragraph 4.9.2 says that a foreign editor can’t be used to edit a project in text view. I don’t see why not. It would just require locking all items in the project.

SPECWARE Options

✂ Add picture.

This tab lets the user set SPECWARE behavior parameters that are available from the Customize menu in the current interface.

Prover Options

➤ Add picture.

This tab lets the user set preferences for the prover behavior that are available from the KITP Options menu in the current interface.

Printing Options

➤ Add picture.

This tab lets the user set preferences for printing.

- ☐ Font for text
Default: ???
- ☐ Font size for text
Default: ???
- ☐ Font for labels in graphs
Default: ???
- ☐ Font size labels in graphs
Default: ???

Miscellaneous

➤ Add picture; maybe these can be combined with Interface Options.

This tab lets the user set preferences that don't fit into other categories, for example:

- ☐ Select the time interval at which changes are written to persistent storage.
Default: ???
- ☐ Choose between lazy and eager evaluation of committed units.
Default: lazy
- ☐ Define the Commit command to show dependencies automatically by bringing up a dependency window (page 115) for the committed item and showing all items that depend on the committed item.
Default: No
- ☐ Define the Commit command to show the new present value immediately by switching to present-value view.
Default: No

Print

✂ Add picture.

The Print dialog box allows the user to select printing options such as the destination device (?), the number of copies to print, the range of pages, and so on.

Include a Page Setup button that brings up the Page Setup dialog box (page 144).

Opening the Print Dialog Box

The Print dialog box is opened when the user:

- ☐ Chooses Print Version Tree, Print Version Description, or Print Development History from the File menu in the project window.
- ☐ Chooses Print *VisitedItem* or Print Selection from the File menu in a browser.
- ☐ Chooses Print *SelectedItems* from the File menu in a browser visiting a project.

Rename

The Rename dialog box prompts the user for a new name for a particular project, unit, or version.

✂ We might want to add a checkbox that is active when a unit is renamed. The checkbox would let the user control whether the name change is propagated through the project or not. If the user simply wants to change the name of unit A to B, all references to A should be changed to refer to B instead. On the other hand, I could imagine that the user might intend to rename unit A to B, then create a new unit A to replace the renamed one. In that case, all the old references to A should remain unchanged (though they would now refer to the new unit A).

The window title changes to indicate what is being renamed. The following table lists the ways this dialog box can be opened and shows the form of the window title in each case.

Opened When User Chooses	Window Title
Rename <i>SelectedItem</i> from the Project menu in a browser	Rename <i>ClassName</i> <i>ItemName</i>
Rename Project <i>CurrentProject</i> from the File menu of the project window	Rename Project <i>ProjectName</i>

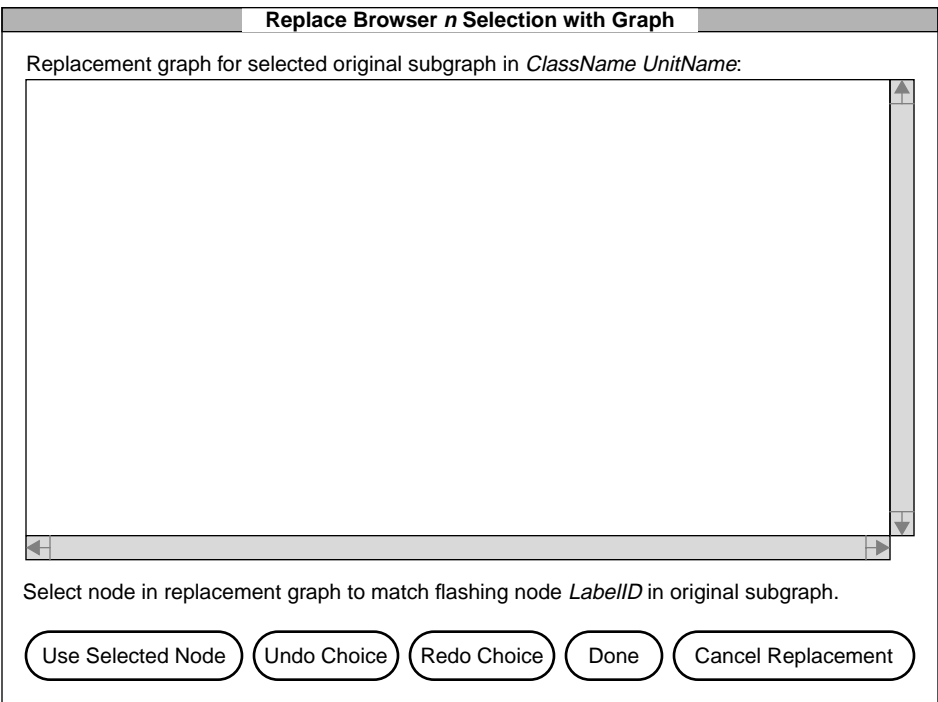
Opened When User Chooses	Window Title
Rename <i>SelectedVersion</i> from the Version menu of the project window	Rename Version <i>VersionName</i>

The New Name field is initialized with the name to be replaced.

The Rename button is greyed out if the name is not changed or if an illegal name is entered.

Replace Selection with Graph

The Replace Selection with Graph dialog box allows the user to indicate the mapping from nodes of a pasted graph to boundary nodes of an original subgraph during a graph replacement operation.



This dialog box is modal; operations are not allowed in other windows while this dialog box is open.

➤ Making this dialog box modal may be too restrictive. If so, see Lambert’s *Design Decisions* paragraph 4.10.23 for the limitations that must be enforced while this dialog box is open.

Opening the Replace Selection with Graph Dialog Box

The Replace Selection with Graph dialog box is opened when nodes or arcs are selected in picture view and the user performs a paste or drop operation as described in *Replacing Subgraphs* on page 99. The title of the dialog box identifies the browser in which the replacement operation is taking place.

A user preference controls whether the dialog box opens for every subgraph replacement operation or only for those operations in which SPECWARE is unable to match all boundary nodes of the original subgraph with nodes of the pasted graph.

Operation

This dialog box lets the user indicate how a replacement graph on the clipboard or being dragged should replace an original subgraph in the browser's visit area. During the replacement process, the boundary nodes of the original subgraph are matched with the nodes of the replacement subgraph. If automatic matching is enabled, any boundary node for which there is a unique match is automatically replaced by the corresponding node in the replacement graph.

While this dialog box is open, SPECWARE iterates through unmatched boundary nodes of the original subgraph, prompting the user for a matching node in the replacement graph. The following actions occur for each unmatched boundary node:

- ☐ In the browser visit area, the unmatched node to be matched is scrolled into view (if necessary) and starts to flash.
- ☐ In the dialog box, the compatible nodes in the replacement graph are rendered to emphasize them relative to the other nodes. The user can choose a compatible node to use as a match either by double clicking it or by clicking the node to select it and then clicking the Use Selected Node button.
- ☐ The chosen node replaces the flashing node in the browser window and SPECWARE proceeds to the next unmatched boundary node. If automatic matching is enabled, choosing one match may have the side effect of resolving other formerly unmatched boundary nodes.

When all boundary nodes have been matched, the Done button is ungreyed. When the user clicks Done, the dialog box closes and the replacement graph replaces the original subgraph in the browser. If the user clicks Cancel Replacement, the replacement process terminates. The original subgraph is restored in the browser and the dialog box closes.

Until the user closes the dialog box by clicking either Done or Cancel Replacement, any match can be undone. The user can click Undo Choice to remove the last match and then Redo Choice to redo it. The user can click Undo Choice repeatedly to undo any number of matches. The user can click Redo Choice repeatedly to redo a series of matches that were undone. After each undo or redo, the browser is updated to show the state that results from undoing or redoing the match. The next unmatched boundary node starts to flash; if none remain (because the last match was redone), the Done button is ungreyed.

Controls

Graph Display

A scrollable graph display area displays the replacement graph. The label on the graph display area identifies the unit being visited in the browser (that is, the unit in which replacement will occur).

As SPECWARE iterates through unmatched boundary nodes of the original subgraph, the compatible nodes in the replacement graph are rendered to emphasize them relative to the other nodes. A line of text beneath the graph display area instructs the user to select a node in replacement graph to match the node that is currently flashing in the browser's visit area; this text identifies the flashing node by its label ID.

The user can click a compatible node to select it or double-click the node to use it as the match for the flashing node. Noncompatible nodes do not respond to mouse clicks.

Command Buttons

Use Selected Node

Causes the selected node in the graph display area to be used as the match for the currently flashing node in the original subgraph. This button is greyed out unless a node is selected in the graph display area.

Undo Choice

Undoes the last match chosen by the user. This button is greyed out if the user has not chosen a matching node.

Redo Choice

Redoes the most recently undone match. This button is greyed out if the user has not undone any match.

Done

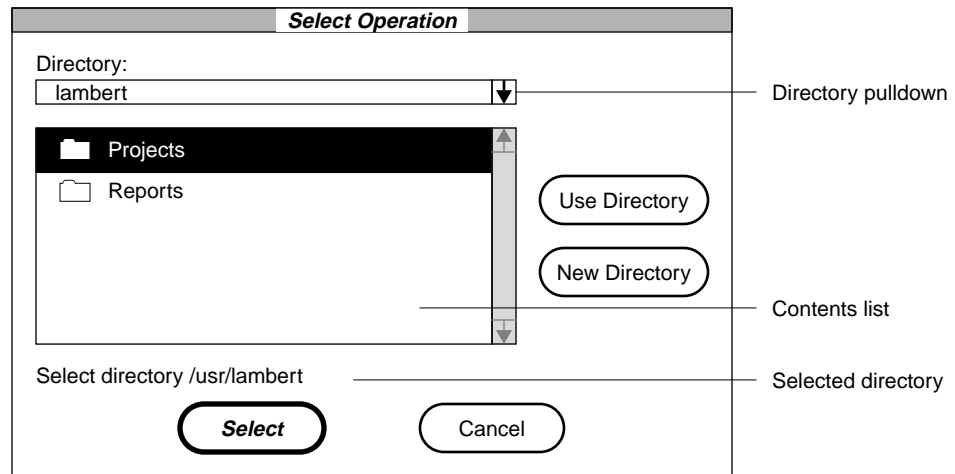
Accepts the replacement operation and closes the dialog box. This button is greyed out if any boundary nodes of the original subgraph are unmatched.

Cancel Replacement

Closes this dialog box without modifying the graph in the browser.

Select Directory

The Select Directory dialog box allows the user to select a directory for some purpose.



Opening the Select Directory Dialog Box

The Select Directory dialog box is opened when the user chooses File-System Index from the View menu of a browser. It's title and the label on the Select button indicate the reason why the window was opened.

Reason	Dialog Box Title	Select Button Label
File-System Index	Select Directory for File-System Index	Create Index

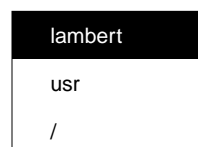
✂ Well, someday we might have other reasons :-)

Controls

The Select Directory dialog box has controls similar to a standard file-selection dialog box; however, it displays only directories, not files.

Directory Pulldown

The directory pulldown shows the name of the currently selected directory. Entries in the menu show the names of each directory in the path from the selected directory to the root of the file-system hierarchy. For example:



The full pathname of the selected directory appears below the subdirectory list.

When the dialog box opens for the first time in a given session, the directory pulldown shows the connected directory from which the user started SPECWARE. Thereafter, the pulldown shows the directory selected most recently in this dialog box, the Open File dialog box, or the Write to File dialog box. The user can move up in the file-system hierarchy by selecting an entry from the directory pulldown.

Subdirectory List

The subdirectory list shows the subdirectories of the directory that appears in the directory pulldown. Names of read-protected subdirectories are greyed out in the list. The user can:

- ☐ Click an entry to select it.
- ☐ Double click an entry to make that subdirectory appear in the directory pulldown, moving down in the file-system hierarchy.

Selected Directory

A label under the subdirectory list shows the full path name of the selected directory. If the full path name is too long to fit in the dialog box, it is truncated on the left and begins with ellipses (...) to indicate that the beginning of the path is not shown. (The user can see all components of the path by opening the directory pulldown.)

Command Buttons

The Select Directory dialog box contains the following command buttons:

Use Directory

If an entry is selected in the subdirectory list, makes that subdirectory appear in the directory pulldown, moving down in the file-system hierarchy. Otherwise, this button is greyed out.

New Directory

Creates a new subdirectory in the current selected directory and makes that subdirectory appear in the directory pulldown, moving down in the file-system hierarchy. This button opens the New Directory dialog box (page 141), which prompts the user for the name of the new directory.

Select

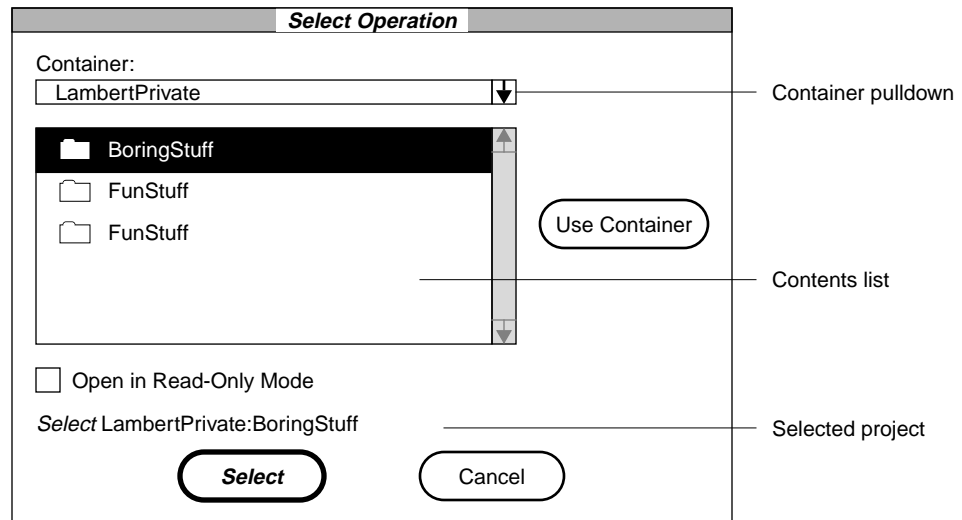
Selects the directory in the directory pulldown and exits the dialog box. This button is the default button.

Cancel

Exits the dialog box, cancelling the operation that brought up this dialog box.

Select Project

The Select Project dialog box allows the user to select a project for some purpose.



Opening the Select Project Dialog Box

The Select Project dialog box is opened when the user chooses Open Project or Duplicate Project from the File menu of the project window. It's title, the label of the selected project, and the label on the Select button indicate the reason why the window was opened.

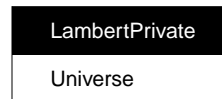
Reason	Dialog Box Title	Select Button Label
Open Project	Open Project	Open Project
Duplicate Project or Library	Duplicate Project or Library	If a project is selected: Duplicate Project If a library is selected: Duplicate Library If nothing is selected (the button is greyed out): Duplicate

Controls

The Select Project dialog box has controls that allow the user to navigate through the universe hierarchy.

Container Pulldown

The container pulldown shows the name of the currently selected container; that is, the universe, a registry, a project, or a library. Entries in the menu show the names of each container in the path from the selected container to the universe (which is the root of the hierarchy). For example:



When the dialog box opens for the first time in an session, the container pulldown shows the user's private registry. After that, it shows the container selected most recently in this dialog box. The user can move up in the universe hierarchy by selecting an entry from the container pulldown.

The full pathname of the selected directory appears below the subdirectory list.

Contents List

The contents list shows the subcontainers of the container that appears in the container pulldown. If the dialog box is being used to select a project to open, libraries do not appear in the list. Names of read-protected containers are greyed out in the list.

When the dialog box opens, the first entry in the list is selected. The user can:

- ☐ Click an entry to select it.
- ☐ Double click an entry to make that container appear in the container pulldown, moving down in the universe hierarchy.

Selected Container

A label under the container list shows the full path name of the selected project or library. When the selected container is the universe or a registry, this label has the form:

Select from *ContainerPathName*

Otherwise, this label has the form:

Select *ProjectOrLibraryPathName*

If the full path name is too long to fit in the dialog box, it is truncated on the left and begins with ellipses (...) to indicate that the beginning of the path is not shown. (The user can see all components of the path by opening the directory pulldown.)

Open in Read-Only Mode

The user can enable the Open in Read-Only Mode checkbox to open the selected project in read-only mode. This checkbox is greyed out when the dialog box is being used to select a project or library to duplicate.

Command Buttons

The Select Project dialog box contains the following command buttons:

Use Container

If an entry is selected in the container list, makes that container appear in the container pulldown, moving down in the file-system hierarchy. Otherwise, this button is greyed out.

Select

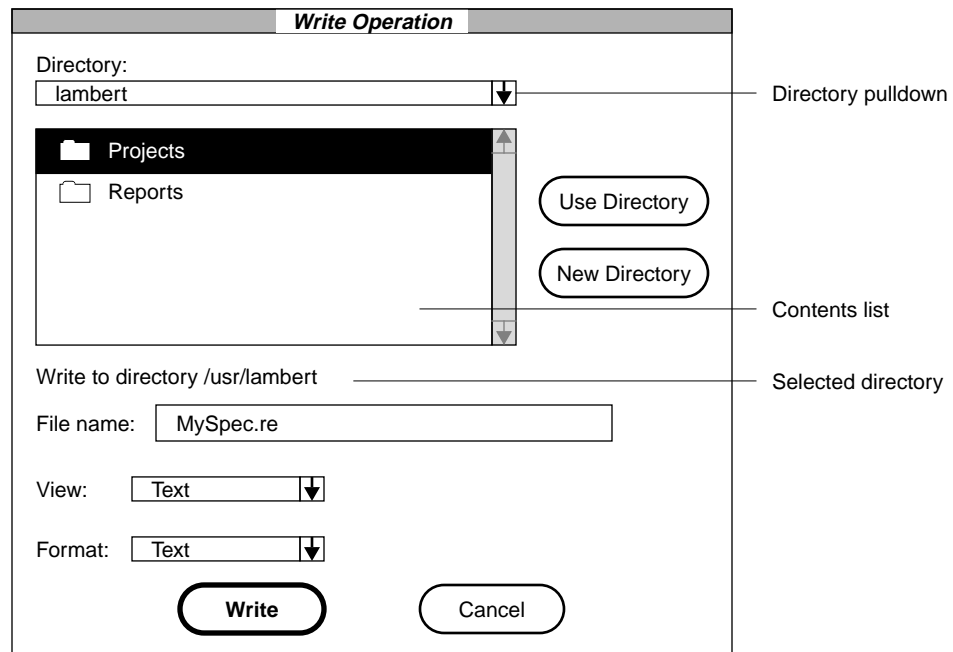
Selects the project or library that is selected in the contents list and exists the dialog box. This button is greyed out when the selected container is the universe or a registry. When it is not greyed out, it is the default button.

Cancel

Exits the dialog box, cancelling the operation that brought up this dialog box. This is the default button when the Select button is greyed out.

Write to File

The Write to File dialog box prompts the user for the name of a file to be written.



Opening the Write to File Dialog Box

The Write to File dialog box is opened when the user:

- ☐ Chooses Write *ItemName* to File from the File menu of a browser.
- ☐ Chooses Write *SelectedItems* to File from the File menu of a browser visiting a project or library. When several items are being written, the only supported view is text view; the default file format is text format.
- ☐ Chooses Save As from the File menu of a browser visiting a file-system file.
- ☐ Clicks the Save Script button in the Command window.

The operation that opens the dialog box determines the title of the dialog box and the default file name. The extension *ext* in default file names means the appropriate extension for the selected view and file format.

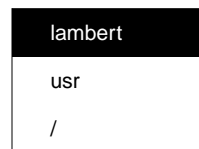
Operation	Title of Dialog Box	Default File Name
Write <i>VisitedItem</i> to File	Write <i>VisitedItem</i>	<i>VisitedItem.ext</i>
Write <i>SelectedItem</i> to File	Write <i>SelectedItem</i>	<i>SelectedItem.ext</i>
Write Selected Items to File	Write Selected Items	SelectedItems.ext
Save As	Save As	Same name as visited file
Save Script	Save Script	Scriptn.txt

Controls

The Write to File dialog box has controls that allow the user to navigate through the file-system hierarchy.

Directory Pulldown

The directory pulldown shows the name of the currently selected directory where the file will be written. Entries in the menu show the names of each directory in the path from the selected directory to the root of the file-system hierarchy. For example:



The full pathname of the selected directory appears below the contents list.

When the dialog box opens for the first time in a given session, the directory pulldown shows the connected directory from which the user started SPECWARE. Thereafter, the pulldown shows the directory selected most recently in this dialog box, the Open File dialog box, or the Select Directory dialog box. The user can move up in the file-system hierarchy by selecting an entry from the directory pulldown.

Contents List

The contents list shows the subdirectories and files in the directory that appears in the directory pulldown. Names of write-protected files and subdirectories are greyed out in the list. The user can:

- ☐ Click a subdirectory entry to select it.
- ☐ Double-click a subdirectory entry to make that subdirectory appear in the directory pulldown, moving down in the file-system hierarchy.
- ☐ Click a file entry to make its name appear in the File Name field.
- ☐ Double-click a file entry to write to a file of that name in the selected directory and exit this dialog box.

Selected Directory

A label between the contents list and the File Name field shows the full path name of the selected directory. If the full path name is too long to fit in the dialog box, it is truncated on the left and begins with ellipses (...) to indicate that the beginning of the path is not shown. (The user can see all components of the path by opening the directory pulldown.)

File Name Field

The File Name field contains the name of the file that will be written in the selected directory. When the dialog box opens, this field contains the default file name as described on page 160.

The user can replace the default name by typing, pasting, or selecting from the contents list.

View Pulldown

The View pulldown menu contains a list of the possible views for the item to be written. When the dialog box opens, this pulldown menu shows the same view being displayed in the browser from which the dialog box was opened. The user can select a different view if desired.

Format Pulldown

The Format pulldown menu contains a list of possible file formats for the chosen view of the item being written to file. For example, the formats for the text view of a spec might be Text and Postscript; the formats for the text view of a project might be Text, Project Interchange, and Postscript; the formats for the picture view of a diagram might be GIF and Postscript.

When the dialog box opens for the first time, this pulldown menu shows:

- ☐ If the same view of the same item has been written before, the same format that was chosen last time.
- ☐ Otherwise, if the same view of an item of the same class has been written before, the same format that was chosen for that item.
- ☐ Otherwise, the default format for the selected view of the item's class.

✂ We need a table of all classes and views giving the available formats and the default format for each view of each class and the appropriate extension for each file format.

Command Buttons

The Write to File dialog box contains the following command buttons:

Use Directory

If a subdirectory entry is selected in the contents list, makes that subdirectory appear in the directory pulldown, moving down in the file-system hierarchy. Otherwise, this button is greyed out.

New Directory

Creates a new subdirectory in the current selected directory and makes that subdirectory appear in the directory pulldown, moving down in the file-system hierarchy. This button opens the New Directory dialog box (page 141), which prompts the user for the name of the new directory.

Write

Writes a file of the indicated name to the selected directory and exits this dialog box. If a file of that name already exists in the directory, a message window opens allowing the user to confirm overwriting the existing file or else cancel the operation. If the user cancels the operation, the Write to File dialog box reopens with the file name selected so the user can type over it.

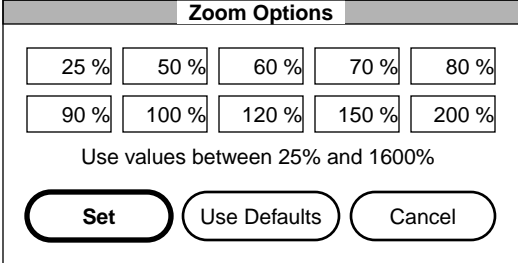
This is the default button. It is greyed out if the File Name field is blank (because the user cleared it).

Cancel

Exits the dialog box, cancelling the operation that brought up this dialog box.

Zoom Options

Zoom Options dialog box lets the user set the ten magnifications displayed in a browser's Magnification pulldown.



Zoom Options

25 %	50 %	60 %	70 %	80 %
90 %	100 %	120 %	150 %	200 %

Use values between 25% and 1600%

Set Use Defaults Cancel

The Zoom Options dialog box is opened when the user chooses **Set** from the Magnification pulldown in the tool bar of the project window or a browser.

A Keyboard Shortcuts

The following table lists the keyboard shortcuts described in this document.

Prefix	Key	Meaning
(none)	End	Scroll to end of active window area
(none)	Home	Scroll to beginning of active window area
(none)	Page down	Scroll active window area forward
(none)	Page up	Scroll active window area backward
(none)	Tab	In table, move selection one cell to the right
(none)	←	In text view, move cursor back one character In version tree, select parent In table move selection one cell to the left
(none)	→	In text view, move cursor forward one character In version tree, select first child In table move selection one cell to the right
(none)	↑	In text view, move cursor up one line In version tree, select previous sibling In table, move selection up one cell
(none)	↓	In text view, move cursor down one line In version tree select next sibling In table, move selection down one cell
Alt	+	In outline, expand heading to section In picture view, expand defining diagram
Alt	-	In outline, contract section to heading In picture view, contract defining diagram
Alt	A	Select all
Alt	B	New Browser
Alt	C	Copy
Alt	F	Find

Prefix	Key	Meaning
Alt	G	Find next
Alt	I	Show/hide index
Alt	J	Visit selected item
Alt	L	Subscribe to selected library
Alt	M	In picture view, search for morphisms
Alt	N	New project
Alt	O	Open project
Alt	P	Print visited item
Alt	Q	Quit Specware
Alt	S	Commit
Alt	V	Paste
Alt	W	Close window
Alt	X	Cut
Alt	Y	Redo
Alt	Z	Undo
Alt	Page down	Forward to next item visited
Alt	Page up	Back to previous item visited
Alt	←	In text view or outline, move cursor back one word
Alt	→	In text view or outline, move cursor forward one word
Alt	↑	In text view, move to previous expression In outline, move to previous paragraph
Alt	↓	In text view, move to next expression In outline, move to next paragraph
Ctrl	G	Go to line
Shift-Alt	+	In picture view, show details of selected node or arc
Shift-Alt	-	In picture view, hide details of selected node or arc
Shift-Alt	A	In picture view, new arc
Shift-Alt	B	Revert to version
Shift-Alt	C	Duplicate selected items
Shift-Alt	F	Find in Project
Shift-Alt	N	In picture view, new node
Shift-Alt	P	Print Selection
Shift-Alt	S	Search for Items
Shift-Alt	T	In outline, change heading to text
Shift-Alt	V	New Version

Prefix	Key	Meaning
Shift-Alt	←	In outline, higher level heading In version tree, move version to previous level in hierarchy
Shift-Alt	→	In outline, lower level heading In version tree, move version to next level in hierarchy
Shift-Alt	↑	In outline, move paragraph up In version tree, move version to before its previous sibling
Shift-Alt	↓	In outline, move paragraph down In version tree, move version to after its next sibling

B Displaying and Editing Labels in Picture View

This appendix lists the possible forms of labels on the nodes and arcs of a graph and explains how a label of each form is displayed in the graph and in the browser's Label field. It also lists the possible modifications that a user can make to the label as displayed in the graph and as displayed in the Label field.

Displaying Labels

The following symbols describe component parts of a label:

ID A label ID

UnitName The name of the unit attached to the labeled node or arc

Term The defining term for the anonymous unit attached to the labeled node or arc

MorphismOp

The morphism operation used to construct the morphism attached to the labeled arc. One of: identity-morphism, translation-morphism, import-morphism, cocone-morphism, inferred-morphism

The following table list the permissible forms for a label and shows how a label of each form appears in the graph (in the node's box or next to the arc's arrow) and in the Label field.

Label	In Graph	In Label Field
<i>ID:UnitName</i>	<i>ID:UnitName</i>	<i>ID:UnitName</i>
<i>UnitName:UnitName</i>	<i>UnitName</i>	<i>UnitName</i>
<i>UnitName</i>	<i>UnitName</i>	<i>UnitName</i>
<i>ID:Term</i>	<i>ID:</i>	<i>ID:Term</i>
<i>ID:</i>	<i>ID:</i>	<i>ID:</i>
<i>MorphismOp</i>	<i>MorphismOp</i>	<i>MorphismOp</i>
<i>ID:MorphismOp</i>	<i>ID:</i>	<i>ID:MorphismOp</i>

Editing Labels

Labels may be edited in the browser's Label field or by entering text-edit mode on the label displayed in the graph in the browser's visit area.

The following symbols are used as components of labels to illustrate how labels may be edited:

A, B	Two different label IDs
X, Y	Two different unit names (which are also different from A and B)
T, U	Two different defining terms
M, N	Two different morphism operations

The following table lists the ways in which an existing label (the old label) can be modified to a syntactically legal new label and describes the effect of each possible modification.

Old Label	New Label	Effect
A:X	A:Y	Keep label name, but detach unit X and attach unit Y
A:X	A:T	Keep label name, but detach unit X and attach term T
A:X	A:	Keep label name, but detach unit X
A:X	A:M	Keep label name, but detach unit X and attach morphism created by morphism operator M
A:X	B:X	Change label name, but keep attached unit
A:X	B:Y	Change label name, detach unit X and attach unit Y
A:X	B:T	Change label name, detach unit X and attach term T
A:X	B:	Change label name and detach unit X
A:X	B:M	Change label name, detach unit X and attach morphism created by morphism operator M
A:X	X:X	Change label name, but keep attached unit
A:X	Y:Y	Change label name, detach unit X and attach unit Y
A:X	X	Change label name, but keep attached unit
A:X	Y	Change label name, detach unit X and attach unit Y
A:X	M	Delete label name, detach unit X and attach morphism created by morphism operator M
A:T	A:X	Keep label name, but detach term T and attach unit X
A:T	A:U	Keep label name, but detach term T and attach term U
A:T	A:	Keep label name, but detach term T
A:T	A:M	Keep label name, but detach term T and attach morphism created by morphism operator M
A:T	B:X	Change label name, detach term T and attach unit X
A:T	B:T	Change label name, but keep attached term

Old Label	New Label	Effect
A:T	B:U	Change label name, detach term T and attach term U
A:T	B:	Change label name and detach term T
A:T	B:M	Change label name, detach term T and attach morphism created by morphisms operator M
A:T	X:X	Change label name, detach term T and attach unit X
A:T	X	Change label name, detach term T and attach unit X
A:T	M	Delete label name, detach term T and attach morphism created by morphisms operator M
A:	A:X	Keep label name, but detach term (if any) and attach unit X
A:	A:T	Keep label name, but detach term (if any) and attach term T
A:	A:M	Keep label name, but detach term (if any) and attach morphism created by morphism operator M
A:	B:X	Change label name, detach term (if any) and attach unit X
A:	B:T	Change label name, detach term (if any) and attach term T
A:	B:	Change label name, but keep attached term (if any)
A:	B:M	Change label name, detach term (if any) and attach morphism created by morphism operator M
A:	X:X	Change label name, detach term (if any) and attach unit X
A:	X	Change label name, detach term (if any) and attach unit X
A:	M	Delete label name, detach term (if any) and attach morphism created by morphism operator M
A:M	A:X	Keep label name, but detach morphism created by morphism operator M and attach unit X
A:M	A:T	Keep label name, but detach morphism created by morphism operator M and attach term T
A:M	A:	Keep label name, but detach morphism created by morphism operator M
A:M	A:N	Keep label name, but detach morphism created by morphism operator M and attach morphism created by morphism operator N
A:M	B:X	Change label name, detach morphism created by morphism operator M and attach unit X
A:M	B:T	Change label name, detach morphism created by morphism operator M and attach term T
A:M	B:	Change label name and detach morphism created by morphism operator M
A:M	B:M	Change label name, but keep attached morphism
A:M	B:N	Change label name, detach morphism created by morphism operator M and attach morphism created by morphism operator N

Old Label	New Label	Effect
A:M	X:X	Change label name, detach morphism created by morphism operator M and attach unit X
A:M	X	Change label name, detach morphism created by morphism operator M and attach unit X
A:M	M	Delete label name, but keep attached morphism
A:M	N	Delete label name, detach morphism created by morphism operator M and attach morphism created by morphism operator N
X:X	A:X	Change label name, but keep attached unit
X:X	A:Y	Change label name, detach unit X and attach unit Y
X:X	A:T	Change label name, detach unit X and attach term T
X:X	A:	Change label name and detach unit X
X:X	A:M	Change label name, detach unit X and attach morphism created by morphism operator M
X:X	B:X	Change label name, but keep attached unit
X:X	B:Y	Change label name, detach unit X and attach unit Y
X:X	B:T	Change label name, detach unit X and attach term T
X:X	B:	Change label name and detach unit X
X:X	B:M	Change label name, detach unit X and attach morphism created by morphism operator M
X:X	Y:Y	Change label name, detach unit X and attach unit Y
X:X	X	Keep label name and keep attached unit
X:X	Y	Change label name, detach unit X and attach unit Y
X:X	M	Delete label name, detach unit X and attach morphism created by morphism operator M
X	A:X	Change label name, but keep attached unit
X	A:Y	Change label name, detach unit X and attach unit Y
X	A:T	Change label name, detach unit X and attach term T
X	A:	Change label name and detach unit X
X	A:M	Change label name, detach unit X and attach morphism created by morphism operator M
X	B:X	Change label name, but keep attached unit
X	B:Y	Change label name, detach unit X and attach unit Y
X	B:T	Change label name, detach unit X and attach term T
X	B:	Change label name and detach unit X
X	B:M	Change label name, detach unit X and attach morphism created by morphism operator M
X	X:X	Keep label name and keep attached unit

Old Label	New Label	Effect
X	Y:Y	Change label name, detach unit X and attach unit Y
X	Y	Change label name, detach unit X and attach unit Y
X	M	Delete label name, detach unit X and attach morphism created by morphism operator M
M	A:X	Add label name, detach morphism created by morphism operator M and attach unit X
M	A:T	Add label name, detach morphism created by morphism operator M and attach term T
M	A:	Add label name and detach morphism created by morphism operator M
M	A:M	Add label name, but keep attached morphism
M	A:N	Add label name, detach morphism created by morphism operator M and attach morphism created by morphism operator N
M	B:X	Add label name, detach morphism created by morphism operator M and attach unit X
M	B:T	Add label name, detach morphism created by morphism operator M and attach term T
M	B:	Add label name and detach morphism created by morphism operator M
M	B:M	Add label name, but keep attached morphism
M	B:N	Add label name, detach morphism created by morphism operator M and attach morphism created by morphism operator N
M	X:X	Add label name, detach morphism created by morphism operator M and attach unit X
M	X	Add label name, detach morphism created by morphism operator M and attach unit X
M	N	Detach morphism created by morphism operator M and attach morphism created by morphism operator N

C Variance with *Design Decisions*

This appendix lists differences between this document and Lambert's *Design Decisions for the Specware UI*, Version 4. Throughout this document, notes to the reader (which are preceded by the \neq symbol) raise issues for discussion. Some of those notes point out differences between this document and *Design Decisions*. This appendix lists other discrepancies that do not appear explicitly in notes to the reader.

Design Decisions describes the conceptual foundation for the interface; most of those descriptions are not duplicated here. Unless otherwise noted, this document is meant to build on those descriptions. An omission in this document should not be interpreted as implying disagreement.

The remainder of this appendix is organized by the paragraph numbers in *Design Decisions*.

4.0.8. I believe all discrepancies are a matter of terminology and that this document described functionality that is consistent with *Design Decisions*. This document does not consider a label in a diagram to be an item. Instead, it considers a label to be a reference to the unit attached to a particular node or arc. The unit can be anonymous; when the user visits an anonymous unit referenced by a label, the unit is not of class *Label*, but of the appropriate class for the category of the diagram containing the label.

The last sentence of 4.0.8 says "They [labels] can be accessed from outside using qualified names." The interface does not allow anonymous unit to be accessed by qualified names, but does allow the user to visit them (using shift-double-click) from either picture or text view of the diagram. See *Visiting Anonymous Units* on page 32.

4.0.9. Terminology difference only: this document uses the single term *browser* instead of the two synonymous terms *browser* and *visit window*.

4.0.11. This document does not mention window chaperones. I am not convinced that we need to invent a new kind of GUI element rather than use standard auxiliary windows and dialog boxes. I believe the intention is to help the user keep track of which browser (host window) a dialog box (window chaperone) goes with. Instead of limiting the movement of dialog boxes, this document proposes titles and descriptive text in dialog boxes that should make in clear what browser and/or unit the dialog box refers to.

I think users would find it unexpected and probably annoying for dialog boxes to move with the window that opened them. This will particularly be the case when screen real estate is limited. Similarly, I can imagine situations where it would be annoying not to be able to move the host window at least partly in front of the dialog box.

I suggest that we implement chaperones as standard dialog boxes and only introduce limitations on their movement if we find that users want (or need) us to do so.

4.0.12 and 4.6.9. *Design Decisions* describes window settings as if each window is visiting a single item. However, a browser visits a sequence of items in succession. I would find it very confusing if the size of the window changed each time I visited a different item. For this reason, I have separated window settings into those related to the browser window itself and those related to a particular visited item. I believe I have captured the intention of these paragraphs in *Window Settings* on page 55 and *Visit History* on page 56 (but check to be sure).

4.8.2. See note to the reader on page 151. I think the user should be able to indicate that they want to rename a unit **without** changing all references.

Neither document mentions what happens when a referencing unit is changed as a result of a Rename operation on a referenced unit. I assume that the referencing unit's status changes to modified (in need of being committed) if it is changed automatically to use the new name.

4.10.11. Terminology difference only: I use the term *label ID* or simply *ID* instead of *label name*. As a consequence, when I talk about a *name* in a label, I mean a *unit name*; when I talk about a label, I always mean the entire label (never just the label ID/name). I found descriptions either confusing or verbose when I used Lambert's terminology. I think we could still do better though because there is a potential of confusing *ID* with *identifier* (name plus term).

4.10.27. This document requires at least one node to be selected for the Colimit operation to be performed. I don't understand the need to allow the Colimit operation with no selected node. A user who wants to add a node with the category's initial object attached can use the New Node operation and enter the node's label.

4.11.8. I think the **only** way to "stabilize" the defining term of an inferred morphism should be by copying and pasting its defining term, For example:

- ☐ Visit the anonymous morphism
- ☐ Copy its defining term
- ☐ Change the label on the arc to the name of a new (undefined) morphism of the appropriate class
- ☐ Visit that morphism in text view and paste the copied defining term

Renaming the label will not work according to our list of ways to edit a label, replacing one syntactically legal label with another. See *Editing Labels* on page 168.

4.11.9. There is no special UI operation Show Defining Diagram; instead, the user simply selects picture view to see the defining diagram of a (morphism or spec).

4.13.4. This paragraph describe some capabilities that are not currently supported by the Search for Items window (page 129), which performs the filtering. In particular:

- ❑ There is no way to find units in libraries only. You can find read-only items (which are probably read-only because they are in libraries). You could also explicitly limit the search to one or more libraries.
- ❑ There is no way to find items modified in the last three days. We could add controls that limit the modification date and/or the commit date. (I'm not sure whether we will keep track of both dates for all items.)

4.13.5. This document doesn't comment on what happens when you add items to a project whose display is currently filtered. I don't object to the proposal in this paragraph, but I'm not sure it's what users would expect.

4.13.16. The Find/Replace in Project window (page 120) doesn't actually have all the same search criteria as the Search for Items window (page 129) because the former is customized for finding text in Klang definitions and the latter is customized for finding items with certain attributes. I think both windows are complex enough as is, so I would prefer not trying to combine the functionary into one window.

These two windows do allow the user to perform the search that this paragraph suggests, but it requires two steps. The user first sets a browser's index to a filtered index of those items that satisfy certain search criteria (entered by the user in the Search for Items window). Then the user performs the text search of the Klang definitions of the items in that browser's index (using the Find/Replace in Project window).

4.18.1. See Note on page 90. The Generate Code command shouldn't be hidden from a new user. Even if the user can generate code by attaching inferred-morphism to an arc, visiting that morphism, and writing it to a file, there should also be a Generate Code menu entry that lets a new user discover that this command is possible (and that guides the user through the process of writing a code file as necessary).

