Chapter

2

2. The CM3-IDE Environment

This chapter describes CM3-IDE's common screens, tools, and icons. You can use this chapter as a reference for finding information about elements of CM3-IDE's development environment.

If you haven't already, you may consider reviewing the previous chapter to learn the basics of the CM3-IDE environment.

Chapter Organization

There are four sections in this chapter:

Common Tools, Icons, and Visual Elements on page 29 describes the common tools and icons within the CM3-IDE environment.

CM3-IDE Start Screen on page 32 describes CM3-IDE's top-level screen in detail.

Summary Screens on page 36 describes screens that are most useful for your development tasks with CM3-IDE. Each screen accompanies a summary of the information and links available from that page.

CM3-IDE's Web Namespace on page 42 provides more in-depth information about CM3-IDE's web namespace.

2.1 Common Tools, Icons, and Visual Elements

CM3-IDE's design defines a consistent environment for navigating, building, and sharing your programs. Common icons, tool buttons, and visual elements reinforce the relationship between different elements, enhancing your ability to find, filter, or act upon information presented within the CM3-IDE development environment.

Here we describe three important elements common to most CM3-IDE pages: Quick Access Icons, Action buttons, and the Find Type-in.

2.1.1 Quick Access Icons

From the CM3-IDE start screen, click on any icon. Wherever you wind up, you will find a row of small icons that cross the top of the page, just under the title for the page. This row of icons is called the quick access icons. The left-most icon (CM3-IDE) always returns you to the start screen. They can bring you to the top of a package, the current subdirectory, or to program, module, or interface summaries of your package components. They can speed your navigation by allowing you to quickly move from one part of the CM3-IDE environment to another.

Together, these icons present an active, visual placeholder for your location in the CM3-IDE web namespace. By using them, you can eliminate most of the Back and Forward activities common to web navigation of complex namespaces.
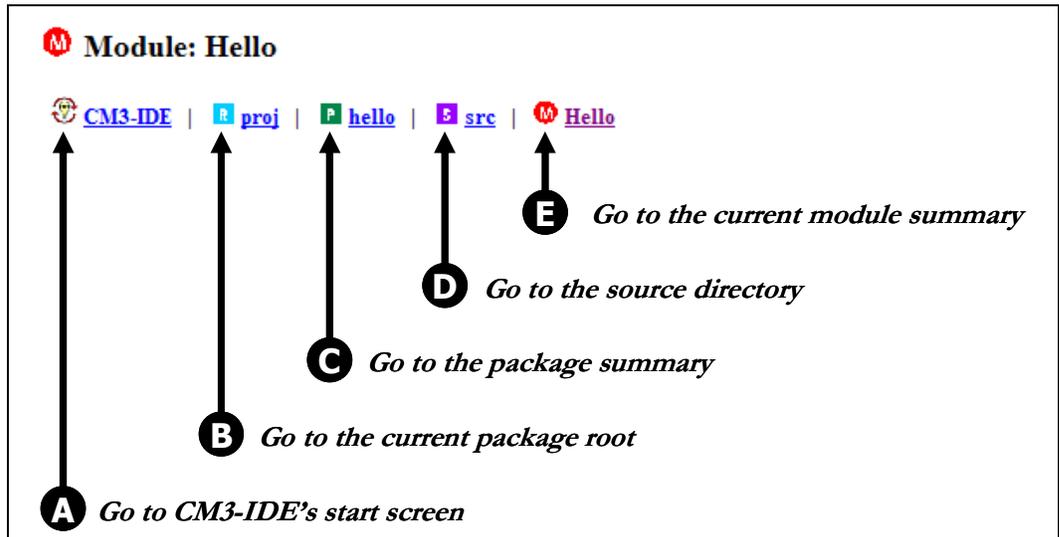

Figure 18. Quick Access Icons for `/proj/hello/src/Hello.m3`

Figure 18 displays a sample row of quick access icons. The icons are arranged according to their level in CM3-IDE's web namespace hierarchy:

A the left-most icon (the one with the CM3-IDE logo) will take you all the way back to the start screen

B takes you to the package root containing the current package

C takes you to the package summary for the current package

D takes you to the sources for your current package. CM3-IDE uses the "`src`" subdirectory of a package to keep the sources for your program.

E the last icon in the path points to the current page, which happens to be a module summary in this example. You can click on the last icon to reload the current page.

The quick access icons work well in conjunction with the history mechanism of your web browser. If you navigate using the quick access icons, your web browser will not lose the starting point in your navigation. Quick access icons allow you to move to a page you've already visited without causing your browser to lose track of your current page in its history.

2.1.2   Action Buttons

| Ship | | Clean | | Edit m3makefile | | Build |
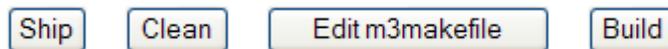
In many ways, working with CM3-IDE is just browsing a web server—much of CM3-IDE's functionality allows fluid navigation within your programming workspace, for example, to jump from a package summary page to the summary page for a library it builds. Each icon or link on a CM3-IDE screen is usually a reference to another element within CM3-IDE.

However, CM3-IDE is more than just a fancy web server. Many CM3-IDE screens allow the invocation of *actions*. For example, a module summary screen includes an action to invoke an editor on the code for the module. Such actions are represented by *action buttons*. The set of available actions varies from page to page, and so does the set of action buttons. Here we describe some of the common action buttons:

| Rescan | Forces CM3-IDE to scan the filesystem to update its state for the current page.

| Build | Builds the currently selected package and shows you the result of the build. Use the text field to the right of the Build button to enter options for the builder. For more information on building, see **Building and Sharing Packages** on page 47.

| Clean | Erases derived files from the current package. Clean will not erase source programs. For more information on building, see **Building and Sharing Packages** on page 47.

| Ship | Releases the current package as a public package, making its contents available for importing, browsing, or executing by other developers in your team. For more on shipping, see **Building and Sharing Packages** on page 47.

| Edit m3makefile | Instructs your text editor to open the makefile used to build the current program.

`Edit source` Instructs your text editor to open the source file corresponding to the current page.

`Run` Command: `C:\MySandbox\hello\NT386\hello.exe`
Directory: `C:\MySandbox\hello\NT386`

Instructs CM3-IDE to run the specified command in the specified directory. CM3-IDE normally fills in the command and directory fields, but you may change them.

2.1.3 The Find Type-in

Find

Wherever CM3-IDE displays a dynamic list of choices (for example, the list of all the packages in your system, or the list of all the modules in a particular package), it also presents the *Find type-in*. It allows you to filter the set of available entries in the current screen to a smaller set, based on the information specified in its type-in field.

To use the find type-in, simply type in the text you are searching for and press return.

The find type-in can accelerate your navigation within CM3-IDE in several ways:

- instead of visually searching a screen for a particular element, you may type the element name, for example "`Main.m3`", in the find type-in.

- instead of searching within a multi-page list of entries by navigating, you can type the name of the item of interest. For example, in the find type-in for the list of all interfaces in your system, you may type "`Fingerprint`" to view the interface `Fingerprint`.

- instead of using CM3-IDE's navigation via links, you may directly jump via a (full or partial) URL to any element. For example, while visiting package `MyPackage`, you may type "`/interface/IO`" in the find type-in to visit the interface `IO`. (See **CM3-IDE's Web Namespace** on page 42 for more information on CM3-IDE URLs.)

2.2 CM3-IDE Start Screen

The CM3-IDE environment consists of many screens (called pages in web-speak.) The next two sections describe some of the more common CM3-IDE screens.

The *start screen* is the screen you see when you first start CM3-IDE. From any page in CM3-IDE, you can return to this screen by clicking on the CM3-IDE icon located at the top-left corner of your page.
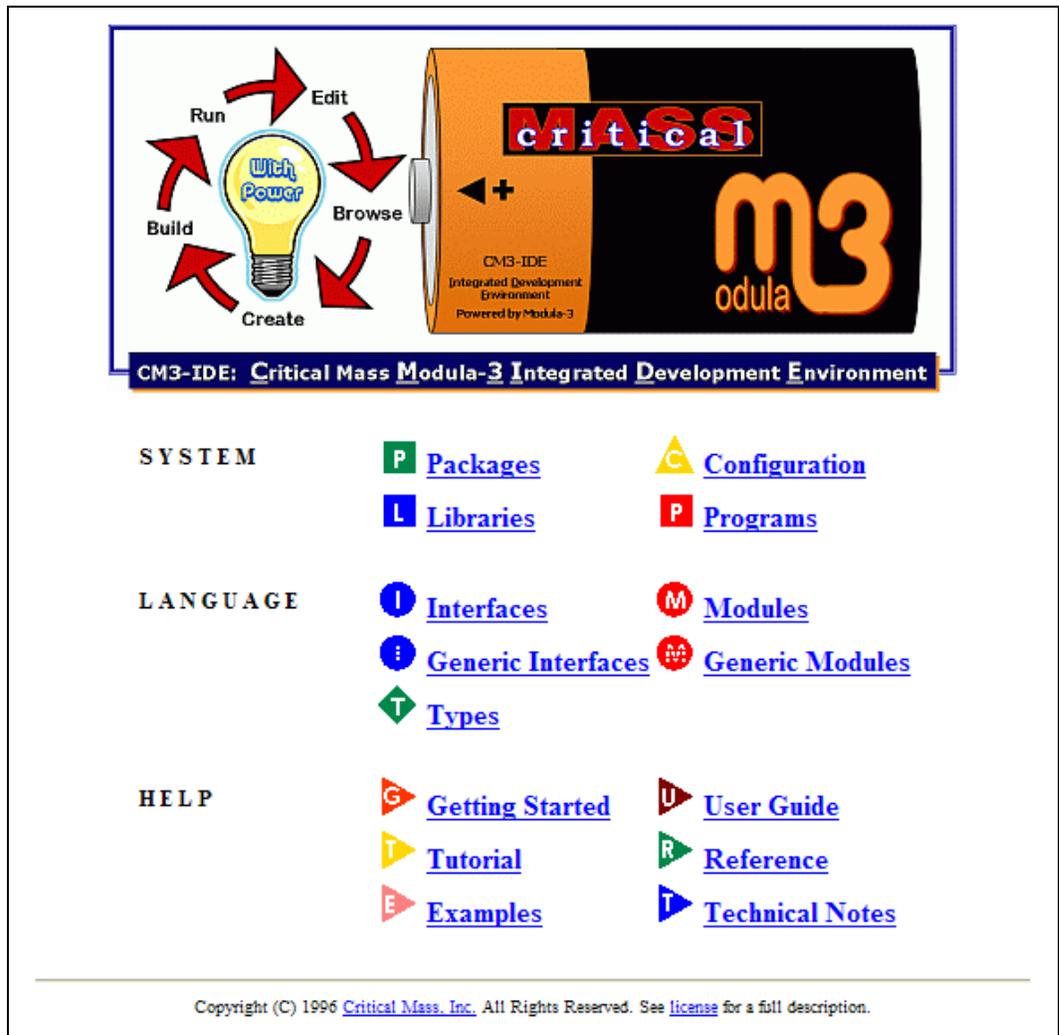
Figure 19. CM3-IDE's Start Screen

On the start screen you'll see three groups of icons. The groups are labeled System, Language, and Help. Each group includes a few links.

- System icons allow you to browse, modify, and customize the CM3-IDE system. You can create a package by following the P Packages link.

- Language icons provide access to program elements in your system. You can find out about all the available interfaces by following the I Interfaces link.

- Help icons refer you to on-line help and information sources. You can find many CM3-IDE examples by following the E Examples link.

Here we describe each of the icons on the start screen.

2.2.1 Start Screen: System

P Packages. Click on this icon to go to the Packages page, where you will see the list of packages in your system organized under package roots (such as public, the public package root, and proj, your private package root.) Click on a package name on the Packages page to visit a summary page for that package.

(For more information about packages, and their organization, see **Building and Sharing Packages** on page 47.)

⚠ Configuration. Click here to go to CM3-IDE's Configuration page.

(For information about changing your configuration, see **Customizing CM3-IDE** on page 65.)

L Libraries. Click here to see a list of all libraries in your system. Clicking on the name within this list will take you to that library's summary.

(For more about libraries, see **Building and Sharing Packages** on page 47.)

P Programs. Click on the Programs to see a list of all programs in your system. Clicking on the name within this list will take you to that program's summary, where you can execute the program.

(For more information about programs, see **Learning the Basics** on page 5.)

2.2.2 Start Screen: Language

I Interfaces, **i Generic Interfaces**. Click on the Interfaces icon to view the list of all available interfaces where you may click on the name of any interface to view its summary page. Generics Interfaces works similarly.

M Modules, **m Generic Modules**. Click on the Modules icon to view the list of available modules where you may click on the name of any module to view its summary page. Generics Modules works similarly.

T Types. Click the Types icon to view the list of types available in all the programs on your system. Selecting a type from the list of all types causes CM3-IDE to display information about that type.

2.2.3 Start Screen: Help

▷ Getting Started. If you haven't read the *CM3-IDE User Guide*, click here for a quick starting point. The page includes an annotated start page, along with tips about many of the common CM3-IDE elements.

Novice users may use this page as their start page. See **Customizing CM3-IDE** on page 65 to learn how to change your start page.

Impatient users can follow this link to find basic information so that they can start using CM3-IDE immediately.

Advanced users may use its HTML source to learn more about customizing CM3-IDE. (See also **CM3-IDE's Web Namespace** on page 42.)

User Guide. Click here to get to the on-line user guide. The *CM3-IDE User Guide* is available in PDF format. Your CM3-IDE distribution contains a copy of Adobe Acrobat PDF Reader for supported platforms.

Tutorial. Following the tutorial link, you will see the list of available tutorials. Two tutorials are available currently: one on Modula-3 and another on Trestle, the portable user interface toolkit shipped with your CM3-IDE distribution.

Reference. Click here to view an on-line, hypertext version of the Modula-3 language definition. This is the on-line version of the printed language definition you received as part of the CM3-IDE package.

Examples. Click here to see a list of example packages for CM3-IDE. When you click on the name of any example package in the list, CM3-IDE creates a private package for you, allowing you to build, modify, and experiment with the example program without corrupting its sources.

You can use the example packages to study many of the programming concepts of CM3-IDE. Sources within examples contain on-line documentation that will guide you through learning various CM3-IDE concepts.

Technical Notes. Click here to see a list of technical reports and information sources about the Modula-3 programming language. View them on-line or print them out as PostScript or Acrobat PDF documents.

2.2.4   Customizing the CM3-IDE Start Screen

You can change your CM3-IDE start screen at any time. Novices may like to use annotated start screens that include helpful comments; advanced users may want to tailor the start screen for more specific needs.

See **Customizing CM3-IDE** on page 65 for more information about changing your start screen.

2.3 Summary Screens

For each programming element in CM3-IDE, there is a corresponding summary screen. The rest of this section describes the common summary screens. Most of these screens are described as part of the hands-on examples in the previous section. Consider working through the examples in **Learning the Basics** on page 5 to learn more about CM3-IDE.

2.3.1   Package Summary

A package summary describes a package and its contents. From a package summary, you can access both sources and derived files in a package.

Figure 20. A Package Summary

Click on any package component (such as an interface or a module) to view it. When browsing package components, you can always return to this page by clicking on the package icon (marked by package-name) as part of the quick access icons.

From top to bottom, a package summary contains:

- quick access icons

- filesystem directory for the current package

- textual description of the current package. This information is extracted from an `index.html` or `README` source file in the current package. An `index.html` file for a package is interpreted within the web namespace for the package. Hence, it can include hypertext links to other package elements, or other names within CM3-IDE. See **CM3-IDE's Web Namespace** on page 42 for more information on CM3-IDE URLs.

- action buttons for this package, to build, ship, or clean the package

- a summary of the sources and derived files that comprise the current package. Clicking on the name of an element will point your browser to the files for that element.

- find type-in. Type in a regular expression to search within the names available in this package. For example, typing in "`m*`" will show you all the elements that start with "`m`".

As is the case with any web page, the information available as part of a package summary may become stale. To bring a package summary up-to-date with its directory contents, reload the package summary page, or use the Rescan button if it is visible. (The Rescan button is only enabled if you have specifically configured CM3-IDE to display it. See **Customizing CM3-IDE** on page 65 for more information.)

2.3.2 Library Summary

A *library summary* describes a library—a group of sources and compiled files in a package that can be used in construction of other packages. The summary page also shows the sources and derived files that comprise the library, where the library resides, and when it was last modified.

Figure 21. A Library Summary

Public libraries (filed under /public in CM3-IDE's namespace) may be imported by other packages in CM3-IDE. For more information about using and sharing libraries, see **Building and Sharing Packages** on page 47.

From top to bottom, a library summary consists of the following elements:

- quick access icons

- path to the library; last modified date; and last build date. If the last build was during the current session, the last built date is a link to the result of that build.

- action buttons for the library

- summary of the sources and derived files that comprise the current library. Clicking on the name of a library element listed here will point your browser to that element's summary page.

- the find type-in.

2.3.3 Program Summary

A *program summary* describes an executable program built using the CM3-IDE system. The summary page for a program shows the sources and derived files that comprise the program, where the program resides, and when it was last modified.
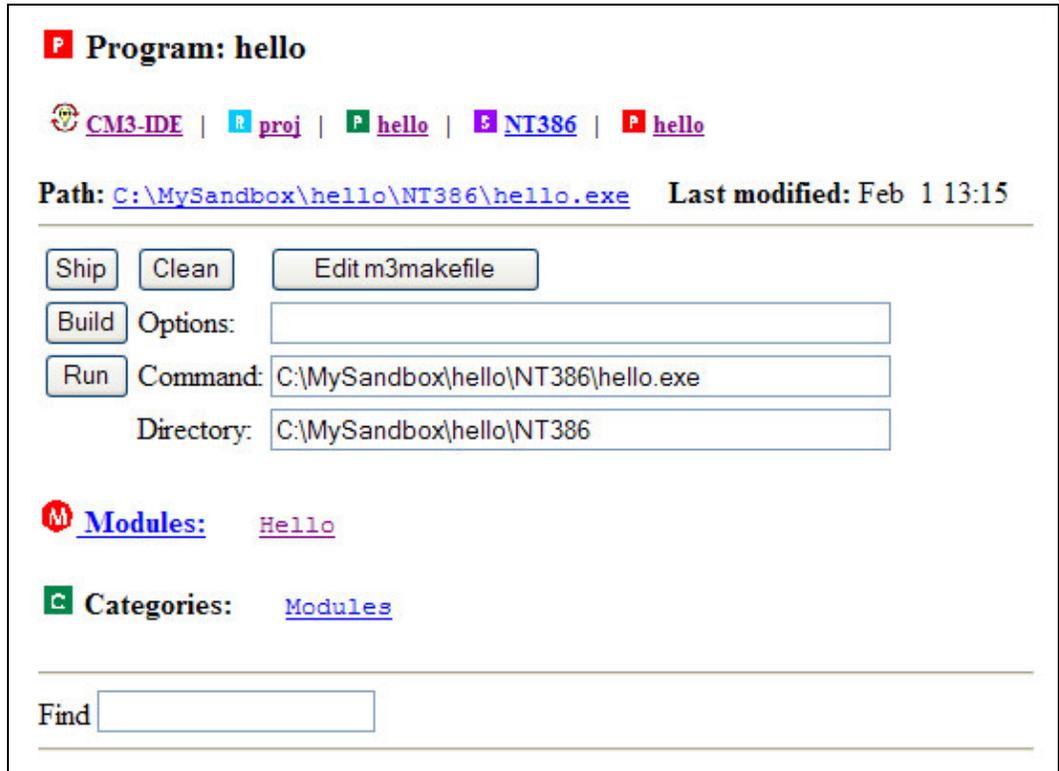

Figure 22. Top Portion of a Program Summary

Public programs (filed under **/public** in CM3-IDE's namespace) are visible to other developers in your group. For more information about using and sharing programs, see see **Building and Sharing Packages** on page 47.

From top to bottom, a program summary consists of the following elements:

- quick access icons

- path to the program; last modified date; and last build date. If the last build was during the current session, the last built date is a link to the result of that build.

- action buttons for this page, including the Run button

- summary of the sources and derived files that comprise the current program. Clicking on the name of a program element listed here will point your browser to that element's summary page.

2.3.4 Interface Summary

An *interface summary* describes an interface. See **Learning the Basics** on page 5 if you would like to learn about interfaces.

Figure 23. An Interface Summary

From top to bottom, an interface summary consists of the following elements:

- quick access icons

- file information for this interface (physical path, last modified)

- export and import dependencies (if the interface is part of a compiled package). Click on "imported by" to see the sources that import this interface. Click on "exported by" to see the sources that export this interface.

- action buttons for this interface, such as Build, Ship, Clean, or Edit Source. These actions are only available when you are allowed to build the package that contains the current interface.

- the code for the interface.

You may click on any of the links within the body of an interface to reach the corresponding program element. For example, you click on a procedure name to view the procedure body within the corresponding module. Some cross-reference links may not be available until the enclosing package is built.

2.3.5 Module Summary

A *module summary* describes an module. See **Learning the Basics** on page 5 if you would like to learn about modules.

Figure 24. Portions of a Module Summary

From top to bottom, a module summary consists of the following elements:

- quick access icons

- file information for this module (physical path, last modified)

- action buttons for this module, such as Build, Ship, Clean, or Edit Source. These actions are only available when you are allowed to build the package that contains the current interface.

- code for the module.

You may click on any of the links within the module body to reach the corresponding program element. For example, you click on a procedure name to view its declaration in the corresponding interface. Some cross-reference links may not be available until a package is built.

Following the links in the **MODULE** or **EXPORTS** clauses of the code, you will visit the interfaces exported by this module. Following the links in the **IMPORT** clause for a module will take you to the interfaces imported by this module.

Most modules include many links to other elements (inside or outside the module) referred to by the code in the module. CM3-IDE marks up your code dynamically to help you navigate your programs.

2.4 CM3-IDE's Web Namespace

This section describes CM3-IDE's web namespace. It is intended as a reference for advanced users.

The heart of CM3-IDE's user interface is its custom web server; CM3-IDE is driven by HTTP requests from your browser. To do this, CM3-IDE associates every one of its elements with a URL or a path.

To see the URL to the current CM3-IDE element, turn on the "Location" or the "URL" display on your browser. CM3-IDE URLs are not only useful for internal use by CM3-IDE itself, they can be used by you in the same way you use ordinary web URLs. You can send URLs referring to a CM3-IDE package, or interface, or even a procedure to a co-worker. Or you may save bookmarks to useful URLs within CM3-IDE. Moreover, you may refer to various elements within CM3-IDE's namespace from your own sources and documentation.

For example, you may include in the **index.html** file for a package, a link to a procedure in your package, with an implementation note for the procedure that points to a particular section of the language reference.

The possibilities for extending CM3-IDE based on this concept are endless. The rest of this section aims at explaining the basic syntax and semantics for CM3-IDE URLs.

2.4.1   CM3-IDE URLs

A CM3-IDE URL follows the format:

> `http://`*`host`*`:`*`port`*`/`*`path`*

where *host* is the host name where you are running CM3-IDE (default: `localhost` which is the host running your browser), *port* is the port that the CM3-IDE server uses (default: `3800`), and *path* is the location of the element within the CM3-IDE namespace. A typical CM3-IDE path may be:

> `http://localhost:3800/interface/IO`

By convention, we omit the protocol, host, and port information for a CM3-IDE URL, so the above example will be described as:

> `/interface/IO`

If an expression matches multiple elements within CM3-IDE's namespace, CM3-IDE will display a list of matches and will allow you to choose one. If there is only one match for your query, CM3-IDE will automatically display its contents. (Note that even a fully-qualified name, such as `/module/Main`, may result in multiple matches.)

A URL also may contain a trailing action, specified via brackets "`[`" and "`]`". For example, `/interface/IO/[edit]` opens your text editor with the file containing the `IO` interface, or `/proj/hello/[build]` builds the package hello.

2.4.2   Example CM3-IDE URLs

Here we briefly include some examples of URLs for CM3-IDE.

`/proj`	your private packages
`/public`	all public packages
`/interface/IO`	interface IO
`/module/M*`	all modules that start with "M"
`/intf/*/Get*`	all interface contents starting with "Get"
`/log`	the last 500 lines of the CM3-IDE log
`/help`	on-line help
`/reference`	language reference
`/`	the start page
`/proj/hello[build]`	build my private package hello
`/module/Hello.m3/[edit]`	edit Hello.m3 module

2.4.3   Regular Expressions in CM3-IDE URLs

CM3-IDE's namespace lookup includes support for regular expressions. That is, in CM3-IDE, paths that express a regular expression (with the correct syntax) are valid URLs. This feature is useful in browsing the large amounts of information available in CM3-IDE.

In particular, you may use the regular expression language in the find type-in. For example, at any point of time, you can type in "`/interface/Rd*`" in a find type-in to search all interfaces whose name starts with "`Rd`", or "`/interface/(Rd|Wr)/Get*`" to find all declarations in `Rd` and `Wr` which start with `Get`.

Regular Expressions Syntax. Here is a list of meta-characters that you may use in CM3-IDE URLs.

`*`	any string of zero or more characters
`@`	any single character
`<expr1>|<expr2>`	strings matching <expr1> or <expr2>
`<expr1>&<expr2>`	strings matching <expr1> and <expr2>
`(expr)`	strings matching <expr>
	(and has higher precedence than or)
`\*`	the single character '*' (asterisk)
`\@`	the single character '@' (at sign)
`|`	the single character '|' (vertical bar)
`\&`	the single character '&' (ampersand)
`\(`	the single character '(' (left parenthesis)
`\)`	the single character ')' (right parenthesis)
`\\`	the single character '\' (back slash)

To learn more about CM3-IDE URLs, examine the sources of fixed CM3-IDE screens, such as `/help/getting-started.html`, or `/example/*/src/index.html` or even the dynamically generated CM3-IDE pages.

2.5 Summary

This chapter describes the common user interface elements in CM3-IDE.

Quick Access Icons. To quickly navigate within the CM3-IDE screens, use the quick access icons. They are displayed on top of every screen. (See page 30.)

Action Buttons. Each CM3-IDE screen supports a number of actions, for example, a module may support an edit action, and a package may support a build action. Available actions for each CM3-IDE element are presented as action buttons in the summary page for that element. (See page 31.)

The Find Type-in. You may search for a particular element within the currently displayed list, or specify a path to a new place in the CM3-IDE namespace in the find type-in box. (See page 32.)

CM3-IDE Start Screen. The start screen is displayed when CM3-IDE starts up. It provides links to directories of elements in the CM3-IDE environment, such as interfaces, modules, packages, libraries, programs, and types. You may also customize CM3-IDE to use a different start screen. (See page 32.)

Summary Screens. Each programming element, such as a package or an interface, has a corresponding summary screen within CM3-IDE. Each summary displays a particular element, links to relevant information, and action buttons for the relevant actions. (See page 36.)

CM3-IDE's Web Namespace. The heart of CM3-IDE's user interface is a custom HTTP sever. CM3-IDE screens are normal web pages; you can send references to them in e-mail to your co-workers, or save a bookmark for them in your browser. CM3-IDE's URLs may contain regular expressions. For example `/interface/F*` displays a list of all interfaces that start with the letter "`F`". Actions for an element are enclosed in square brackets ("`[`" and "`]`"). You can invoke actions by using URLs also, for example, `/module/Hello[edit]` edits `Hello.m3` the file for the `Hello` module. (See page 42.)

This page left blank
intentionally.