

Generating Slice Documentation

On this page:

- [Generating Slice Documentation](#)
- [Documentation Comments](#)
 - [Hyperlinks](#)
 - [Explicit Cross-References](#)
 - [Markup for Operations](#)
 - [General HTML Markup](#)
- [Using slice2html](#)

Generating Slice Documentation

If you look at the online [Slice API reference](#), you will find reference documentation for all the Slice definitions used by Ice and its services. In the binary distributions of Ice, you will also find HTML documentation that contains the same information. The HTML documentation is generated from special comments in the Slice definitions using `slice2html`, a tool that scans Slice definitions for special comments and generates HTML pages for those comments.

As an example of documentation comments, here is the definition of `Ice::Current`:

Slice

```

/**
 *
 * Information about the current method invocation for servers. Each
 * operation on the server has a <tt>Current</tt> as its implicit final
 * parameter. <tt>Current</tt> is mostly used for Ice services. Most
 * applications ignore this parameter.
 */
local struct Current {
    /**
     * The object adapter.
     */
    ObjectAdapter adapter;

    /**
     * Information about the connection over which the current
     * method invocation was received. If the invocation is direct
     * due to collocation optimization, this value is set to null.
     */
    Connection con;

    /**
     * The Ice object identity.
     */
    Identity id;

    /**
     * The facet.
     */
    string facet;

    /**
     * The operation name.
     */
    string operation;

    /**
     * The mode of the operation.
     */
    OperationMode mode;

    /**
     * The request context, as received from the client.
     */
    Context ctx;

    /**
     * The request id unless oneway (0) or collocated (-1).
     */
    int requestId;

    /**
     * The encoding version used to encode the input and output parameters.
     */
    Ice::EncodingVersion encoding;
};

```

If you look at the comments, you will see these reflected in the documentation for `Ice::Current` in the online [Slice API reference](#).

Documentation Comments

A documentation comment:

- starts with `/**`
- ends with `**/`

Such a comment can precede any Slice construct, such as a module, interface, structure, operation, and so on. Within a documentation comment, you can either start each line with a `*`, or you can leave the beginning of the line blank — `slice2html` can handle either convention:

Slice

```
/**
 *
 * This is a documentation comment for which every line
 * starts with a '*' character.
 */

/**

This is a documentation comment without a leading '*'
for each line. Either style of comment is fine.

**/
```

The first sentence of the documentation comment for a Slice construct should be a summary sentence. `slice2html` generates an index of all Slice constructs; the first sentence of the comments for each Slice construct is used as a summary in that index.

Hyperlinks

Any Slice identifier enclosed in `{@link ...}` is presented as a hyperlink in code font. For example:

Slice

```
/**
 * An empty {@link name} denotes a null object.
 */
```

This generates a hyperlink for the name markup that points at the definition of the corresponding Slice symbol. (The symbol can denote any Slice construct, such as a type, interface, parameter, or structure member.)

Explicit Cross-References

The directive `@see` is recognized by `slice2html`. Where it appears, the generated HTML contains a separate section titled "See Also", followed by a list of Slice identifiers. For example:

Slice

```
/**
 * The object adapter, which is responsible for receiving requests
 * from endpoints, and for mapping between servants, identities,
 * and proxies.
 *
 * @see Communicator
 * @see ServantLocator
 */
```

The Slice identifiers are listed in the corresponding "See Also" section as hyperlinks in code font.

Markup for Operations

There are three directives specifically to document Slice operations: @param, @return, and @throws. For example:

Slice

```
/**
 * Look for an item with the specified
 * primary and secondary key.
 *
 * @param p The primary search key.
 *
 * @param s The secondary search key.
 *
 * @return The item that matches the specified keys.
 *
 * @throws NotFound Raised if no item matches the specified keys.
 */

Item findItem(Key p, Key s) throws NotFound;
```

slice2html generates separate "Parameters", "Return Value", and "Exceptions" sections for these directives. Parameters are listed in the same order as they appear in the comments. (For clarity, that order should match the order of declaration of parameters for the corresponding operation.)

General HTML Markup

A documentation comment can contain any markup that is permitted by HTML in that place. For example, you can create separate paragraphs with <P> and </P> elements:

Slice

```
/**
 * This is a comment for some Slice construct.</p>
 *
 * <p>This comment appears in a separate paragraph.
 */
```

Note that you must neither begin a documentation comment with a <p> element nor end it with a </p> element because, in the generated HTML, documentation comments are already surrounded by <p> and </p> elements.

There are various other ways to create markup — for example, you can use <table> or elements. Please see the [HTML specification](#) for details.

Using slice2html

slice2html uses the following syntax:

```
slice2html [options] slice_file...
```

If you have cross-references that span Slice files, you must compile all of the Slice files with a single invocation of slice2html.

The command supports the following options:

- -h, --help
Displays a help message.
- -v, --version
Displays the compiler version.
- -DNAME
Defines the preprocessor symbol NAME.

- **-DNAME=DEF**
Defines the preprocessor symbol *NAME* with the value *DEF*.
- **-UNAME**
Undefines the preprocessor symbol *{NAME}*.
- **-IDIR**
Add the directory *DIR* to the search path for `#include` directives.
- **-E**
Print the preprocessor output on `stdout`.
- **--output-dir DIR**
Place the generated files into directory *DIR*.
- **-d, --debug**
Print debug information showing the operation of the Slice parser.
- **--ice**
Permit use of the normally reserved prefix *Ice* for identifiers. Use this option only when compiling the source code for the *Ice* run time.
- **--underscore**
Permit use of underscores in Slice identifiers.
- **--hdr FILE**
Prepend *FILE* to each generated HTML file (except for `_sindex.html`). This allows you to replace the HTML header and other preamble information with a custom version, so you can connect style sheets to the generated pages. The specified file must include the `<body>` tag (but need not end with a `<body>` tag). *FILE* is expected to contain the string `TITLE` on a line by itself, starting in column one. `slice2html` replaces the `TITLE` string with the fully-scoped name of the Slice symbol that is documented on the corresponding page.
- **--ftr FILE**
Append *FILE* to each generated HTML file (except for `_sindex.html`). This allows you to add, for example, a custom footer to each generated page. *FILE* must end with a `</body>` tag.
- **--indexhdr FILE**
`slice2html` generates a file `_sindex.html` that contains a table of contents of all Slice symbols that hyperlink to the corresponding page. This option allows you to replace the standard header with a custom header, for example, to attach a JavaScript. The specified file must include the `<body>` tag (but need not end with a `<body>` tag). The default value is the setting of `--hdr` (if any).
- **--indexftr FILE**
Append *FILE* to the generated `sindex.html` page. This allows you to add, for example, a custom footer to the table of contents, or to invoke a JavaScript. *FILE* must end with a `</body>` tag. The default value is the setting of `--ftr` (if any).
- **--image-dir DIR**
With this option, `slice2html` looks in the specified directory for images to use for the generated navigation hyperlinks. (Without this option, text links are used instead.) Please see the generated HTML for the names of the various image files. (They can easily be found by looking for `img` elements.)
- **--logo-url URL**
Use the specified URL as a hyperlink for the company logo that is added to each page (if `--image-dir` is specified). The company logo is expected to be in `<image_dir>/logo.gif`.
- **--search ACTION**
If this option is specified, the generated pages contain a search box that allows you to connect the generated pages to a search engine. On pressing the "Search" button, the specified *ACTION* is carried out.
- **--index NUM**
`slice2html` generates sub-indexes for various Slice symbols. This option controls how many entries must be present before a sub-index is generated. For example, if *NUM* is set to 3, a sub-index will be generated only if there are three or more symbols that appear in that index. The default settings is 1, meaning that a sub-index is always generated. To disable sub-indexes entirely, set *NUM* to 0.
- **--summary NUM**
If this option is set, summary sentences that exceed *NUM* characters generate a warning.

See Also

- [Slice API reference](#)
- [HTML specification](#)