

NAME

mdoc – Mono documentation management tool

SYNOPSIS

mdoc *command* [*options*] [*args*]

OVERVIEW

mdoc is an assembly-based documentation management system.

mdoc permits creating and updating documentation *stubs* based on the contents of an assembly. It does not rely on documentation found within the source code.

The advantages are:

- * **Code readability.** Good documentation is frequently (a) verbose, and (b) filled with examples. (For comparison, compare Microsoft .NET Framework documentation, which is often a page or more of docs for each member, to JavaDoc documentation, which can often be a sentence for each member.)

Inserting good documentation into the source code can frequently bloat the source file, as the documentation can be longer than the actual method that is being documented.
- * **Localization.** In-source documentation formats (such as *csc /doc*) have no support for multiple human languages. If you need to support more than one human language for documentation purposes, *mdoc* is useful as it permits each language's output to reside in its own directory, and *mdoc* can add types/members for each separate documentation directory.
- * **Administration.** It's not unusual to have separate documentation and development teams. It's also possible that the documentation team will have minimal experience with the programming language being used. In such circumstances, inline documentation is not desirable as the documentation team could inadvertently insert an error into the source code while updating the documentation. Alternatively, you may not want the documentation team to have access to the source code for security reasons. *mdoc* allows the documentation to be kept *completely* separate and distinct from the source code used to create the assembly.

Documentation can be generated using the **mdoc update** command:

```
mdoc update -o docs/en ProjectName.dll
```

Once the documentation stubs have been generated (and hopefully later filled in with actual documentation), there are three ways to view the documentation:

- * To generate a simple directory of HTML pages (one HTML file per type), use **mdoc export-html**:

```
mdoc export-html -o /srv/www/htdocs/ProjectName docs/en
```

- * To use an ASP.NET webapp to display the sources, see: <http://anonsvn.monoproject.com/source/trunk/monodoc/engine/web/>.

From a *monodoc* source checkout, you can do this:

```
cd engine
make web
```

This will use **xsp(1)** to serve the ASP.NET webapp; Visit <http://localhost:8080/> to view the documentation.

- * To use the **monodoc(1)** documentation browser, you must first *assemble* the documentation:

```
mdoc assemble -o ProjectName docs/en
```

The above command creates the files *ProjectName.tree* and *ProjectName.zip*. An additional *ProjectName.sources* file must be provided which describes where in the help system the documentation should be hooked up; it is a very simple XML file, like this:

```
<?xml version="1.0"?>
<monodoc>
  <source provider="ecma" basefile="ProjectName"
    path="various" />
</monodoc>
```

The above configuration file describes that the documentation is in ECMA format, that the base file name is *ProjectName* and that it should be hooked up in the "various" part of the documentation tree. If you want to look at the various nodes defined in the documentation, you can look at the *monodoc.xml* file which is typically installed in */usr/lib/monodoc/monodoc.xml*.

Once you have all of the required files (.zip, .tree and .sources) you can install them into the system with the following command:

```
cp ProjectName.tree ProjectName.zip ProjectName.source \
  'pkg-config monodoc --variable sourcedir'
```

The above will copy the files into the directory that Monodoc has registered; you might need root permissions to do this. The actual directory is returned by the *pkg-config* invocation.

MDOC COMMANDS

mdoc assemble

Compiles documentation for use within the **monodoc(1)** browser.

See the **mdoc-assemble(1)** man page for details.

mdoc export-html

Exports documentation into a directory structure of HTML files.

See the **mdoc-export-html(1)** man page for details.

mdoc export-msxdoc

Exports documentation into the single-file *Microsoft XML Documentation* format.

See the **mdoc-export-msxdoc(1)** man page for details.

mdoc help

View internal help for a given command.

```
mdoc help assemble
```

is equivalent to:

```
mdoc assemble --help
```

Multiple sub-commands may be listed at once:

```
mdoc help assemble export-html update validate
```

mdoc update

Updates documentation, adding and removing members based upon a reference assembly.

See the **mdoc-update(1)** man page for details.

mdoc validate

Validates the documentation against the Mono documentation schema.

See the **mdoc-validate(1)** man page for details.

SEE ALSO

mdoc(5), mdoc-assemble(1), mdoc-export-html(1), mdoc-update(1), mdoc-validate(1)

MAILING LISTS

Visit <http://lists.ximian.com/mailman/listinfo/mono-docs-list> for details.

WEB SITE

Visit <http://www.mono-project.com/docs/tools+libraries/tools/mdoc/> for details