NAME

mdoc assemble − Compile documentation for use in **monodoc**(1)

SYNOPSIS

mdoc assemble [OPTIONS]+ PATHS+

DESCRIPTION

mdoc assemble creates *.tree* and *.zip* files from *PATHS* for use in the **monodoc**(1) documentation browser.

The input files must have a supported *format*, specified with the *--format* option.

The *.tree* and *.zip* files are copied into **monodoc**'s *sources* directory, alongside a *.source* file which is used by **monodoc**(1) to specify where the documentation should be displayed.

The *.source* file has the following format:

```
<?xml version="1.0"?>
<monodoc>
  <node label="LABEL" name="PATH" parent="PARENT">
    <node label="LABEL2" name="PATH2" />
    <!-- ... -->
  </node>
  <source provider="PROVIDER" basefile="BASEFILE" path="PATH" />
  <!-- other <source/> elements -->
</monodoc>
```

The */monodoc/node* node is an optional node that specifies where in the monodoc tree the documentation should be displayed, and *//node* elements may be nested to any depth to create trees. *//node/@label* is the label that will be displayed within the monodoc tree.

//node/@name is the name of the monodoc tree node, and may be used as the value of the */monodoc/source/@path* value.

//node/@parent is the node name to use as the parent node. *$MONO_INSTALL_PREFIX/lib/monodoc/monodoc.xml* contains a list of such names, and this can be any *//node/@name* value. If the *//node/@parent* value isn't found, then it's inserted under the "Various" tree node.

The */monodoc/source/@provider* attribute specifies which format provider should be used when reading the *.tree* and *.zip* files; this *must* correspond to one of the *--format* values.

The */monodoc/source/@basefile* attribute specifies the filename prefix for the documentation files. This must be the same prefix as used with the *−−out* parameter. There should be *no* filename extension on this value.

The */monodoc/source/@path* attribute specifies the parent node in **monodoc**(1)'s tree view where the documentation will be inserted. See the *$MONO_INSTALL_PREFIX/lib/monodoc/monodoc.xml* file for a list of *PATH* values (the *//node/@name* values), or it may be a *//node/@name* value in the same *.source* file.

Once the *BASEFILE.source* has been written, the documentation can be installed so that **monodoc**(1) will display the documentation with the command:

```
cp BASEFILE.source BASEFILE.tree BASEFILE.zip \
  `pkg-config monodoc --variable=sourcesdir`
```

OPTIONS

−f, **−−format=***FORMAT*

Specifies the documentation format used within *PATHS*. Valid *FORMAT* values include: *ecma*, *ecmaspec*, *error*, *hb*, *man*, *simple*, and *xhtml*.

See the *FORMATS* section below for more information about these formats.

The default format (if none is specifed) is *ecma*.

The *--format* option may be interleaved with *PATHS* to change the format used for the remaining parameters (until the next *--format* option is seen), e.g.:

```
mdoc assemble -o PREFIX A B --format=man C D --format=xhtml E
```

will assemble directories *A* and *B* with the *ecma* format, files *C* and *D* with the *man* formt, and directory *E* with the *xhtml* format.

-o, **--out**=*PREFIX*

Specify the output file prefix. **mdoc assemble** creates the files *PREFIX.zip* and *PREFIX.tree*.

-h, **-?**, **--help**

Display a help message and exit.

FORMATS

The following documentation formats are supported:

ecma

The *Mono ECMA Documentation Format*, an XML documentation format with one file per type.

See the **mdoc**(5) man page for more information.

ecmaspec

The *Mono ECMA Specification Documentation Format*. This is not the format you're looking for; it is the format used to represent the ECMA-334 (C#) standard within **monodoc**(1). It is not used to display class library documentation; for class library documentation, use the **ecma** format.

error

The *Error Documentation Format* is used to present detailed error messages, and is used in **monodoc**(1)'s "C# Compiler Error Reference" tree.

In this format, *PATHS* is a configuration file, containing the XML:

```
<ErrorProviderConfig>
  <FilesPath>../../mcs/errors</FilesPath>
  <Match>cs????*.cs</Match>
  <ErrorNumSubstringStart>2</ErrorNumSubstringStart>
  <ErrorNumSubstringLength>4</ErrorNumSubstringLength>
  <FriendlyFormatString>CS{0:0###}</FriendlyFormatString>
</ErrorProviderConfig>
```

The elements mean:

/ErrorProviderConfig/FilesPath

Specifies where to look for files.

/ErrorProviderConfig/Match

Specifies the filename pattern to look for within the */ErrorProviderConfig/FilesPath* directory.

/ErrorProviderConfig/ErrorNumSubstringStart

Specifies where within the filename the error number starts.

/ErrorProviderConfig/ErrorNumSubstringLength

Specifies how many characters after */ErrorProviderConfig/ErrorNumSubstringStart* to use for the error number.

/ErrorProviderConfig/FriendlyFormatString

Specifies the formatting/display of the node in the **monodoc**(1) tree.

For each file found, it is converted to HTML with C# syntax coloring applied.

simple

The *Simple Documentation Format* file format recursively adds all files and directories underneath *PATHS*. When displayed, HTML files are displayed as-is. Text files are converted into HTML by translating each newline into an HTML *<br>* element. No other file type is supported.

man

The *Man Page Documentation Format* displays groff man pages. (This is *not* a full groff parser, and only handles the man page constructs used within the mono man pages.)

PATHS is a set of XML files containing:

```
<?xml version="1.0"?>
<manpages>
  <manpage name="NAME" page="FILE" />
</manpages>
```

There may be multiple *//manpage* elements within the root */manpage* element.

The */manpages/manpage/@name* attribute contains the display name for the tree view node, which is also the URL of the man page when using **monodoc**(1)'s *Lookup URL* command (before prefixing with a *man:* prefix). Thus, if */manpages/manpage/@name* contains *mono(1)*, then *man:mono(1)* can be used in the *Lookup URL* command to view the *mono(1)* man page.

The */manpages/manpage/@page* attribute is the filename that contains the man page. If this file does not exist, then */manpages/manpage/@name* will not be displayed within the tree view.

xhtml

The XHTML provider interprets *PATHS* as a Windows Help file XHTML TOC file and looks for referenced documents to create the help source.

SEE ALSO

mdoc(1), **mdoc**(5), **monodoc**(1)

MAILING LISTS

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

WEB SITE

See also: http://www.mono-project.com/docs/tools+libraries/tools/mdoc/