NAME

mdoc update – **mdoc**(5) documentation format support

SYNOPSIS

mdoc update [OPTIONS]* ASSEMBLIES

DESCRIPTION

mdoc update is responsible for the following:

* Creating documentation stubs based on *ASSEMBLIES*. The stub-creation process will create new **mdoc**(5) XML files for each type within *ASSEMBLIES*, and provide documentation stubs for each member of those types.

* Update documentation stubs based on *ASSEMBLIES*. Existing **mdoc**(5) documentation can be updated to reflect changes within *ASSEMBLIES*, such as added types and members, while preserving existing documentation.

  In some limited circumstances, renames will be tracked, minimizing the documentation burden when e.g. a parameter is renamed.

mdoc update does not rely on documentation found within source code, though it can import XML Documentation Comments via the **–i** option.

See **mdoc**(1) and **mdoc**(5) for more information.

OPTIONS

––delete

Allow **mdoc update** to delete members from documentation files. The only members deleted are members which are no longer present within *ASSEMBLIES* and are not present in any other assembly versions.

If a type is no longer present, the documentation file is *not* deleted, but is instead *renamed* to have a **.remove** extension.

Version detection is done with the *//AssemblyVersion* elements; if there are no *//AssemblyVersion* elements for a given *<Type>* or *<Member/>*, then the *<Type>* will be renamed and/or the *<Member/>* will be removed.

––exceptions[=*SOURCES*]

EXPERIMENTAL. This is not 100% reliable, but is intended to serve as an aid for documentation writers.

Inspect member bodies to determine what exceptions can be generated from the member.

SOURCES is an optional comma-separated list of the following sources that should be searched for exceptions:

added Only generate <exception/> elements for members added during the current program execution. This keeps mdoc-update from re-generating <exception/> elements for all members (and thus prevents re-insertion for members that had the <exception/> elements removed).

all Find exceptions created in the member itself, references to members in the same assembly, and references to members in dependent assemblies.

asm Find exceptions created in the member itself and references to members within the same assembly as the member.

depasm Find exceptions created in the member itself and references to members within dependent

assemblies.

If *SOURCES* isn't provided (the default), then only exceptions created within the member itself will be documented.

LIMITATIONS: Exception searching is currently implemented by looking for the exception types that are explicitly created based on the known compile-time types. This has the following limitations:

* This will not find exceptions which are implicit to the IL, such as NullReferenceException and IndexOutOfRangeException.

* This will find exceptions which are *not* thrown, e.g.

```
public void CreateAnException ()
{
  Exception e = new Exception ();
}
```

* This will not "follow" delegate and interface calls:

```
public void UsesDelegates ()
{
  Func<int, int> a = x => {throw new Exception ();};
  a (4);
}
```

The function *UsesDelegates()* won't have any exceptions documented.

* This will find exceptions which "cannot happen", such as ArgumentNullExceptions for arguments which are "known" to be non-null:

```
public void A ()
{
  B ("this parameter isn't null");
}

public void B (string s)
{
  if (s == null)
    throw new ArgumentNullException ("s");
}
```

For the above, if **--exceptions=asm** is provided then *A()* will be documented as throwing an ArgumentNullException, which cannot happen.

–f=*FLAG*

Specify a flag to alter behavior. Valid flags include:

no-assembly-versions

See the **-fno-assembly-versions** documentation, below.

–fno-assembly-versions

Do not generate */Type/AssemblyInfo/AssemblyVersion* and */Type/Members/Member/AssemblyInfo* elements.

This is useful to prevent "churn" during updates. Normally, if a type or member hasn't changed but the assembly version has changed, then all types and members will be updated to include a new *//AssemblyVersion* element, thus increasing the amount of changes that need review before

committing (assuming all changes are actually reviewed before commit).

WARNING: This *will* interact badly with the **--delete** option, as **--delete** uses the *//AssemblyVersion* elements to track version changes. Thus, if you have a member which is present in an early assembly version and is removed in a subsequent assembly version, such as *System.Text.UTF8Encoding.GetBytes(string)* (which is present in .NET 1.0 but not in .NET 2.0), then the member will be removed when the **--delete -fno-assembly-versions** options are specified, the member was present in an earlier version of the assembly, and the current version of the assembly does not contain the member.

Consequently, this option should *only* be specified if types and members will *never* be removed from an assembly.

–i, **––import**=*FILE*
Import documentation found within *FILE*.

FILE may contain either *csc /doc* XML or *ECMA-335* XML.

–L, **––lib**=*DIRECTORY*
Add *DIRECTORY* to the assembly search path, so that dependencies of *ASSEMBLIES* can be found without documenting those assemblies.

–o, **––out**=*DIRECTORY*
Place the generated stubs into *DIRECTORY*.

When updating documentation, *DIRECTORY* is also the source directory.

–r=*ASSEMBLY*
ASSEMBLY is a dependency for one of *ASSEMBLIES* which should *not* be documented but is required to process one of *ASSEMBLIES*. Add the directory containing *ASSEMBLY* to the assembly search path.

This option is equivalent to specifying **–L** ‘*dirname* ASSEMBLY‘.

––since=*VERSION*
When *updating* documentation for an assembly, if a type or member is encountered which didn’t exist in the previous version of the assembly a **<since version="***VERSION***"/>** element will be inserted.

––type=*TYPE*
Only update documentation for the type *TYPE*.

–h, –?, ––help
Display a help message and exit.

SEE ALSO

mdoc(1), mdoc(5), mdoc-assemble(1), mdoc-export-html(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 for details