

NAME

mono-config – Mono runtime file format configuration

DESCRIPTION

The Mono runtime will load configuration data from the installation prefix `/etc/mono/config` file, the `~/mono/config` or from the file pointed by the `MONO_CONFIG` environment variable.

For each assembly loaded a config file with the name: `/path/to/the/assembly.exe.config` is loaded as well as the `~/mono/assemblies/ASSEMBLY/ASSEMBLY.EXT.config` file.

This file controls the behavior of the runtime.

The file contains an XML-like file with various sections, all of them contained inside a section (It actually uses GMarkup to parse the file).

This page describes the Unix-specific and Mono-specific extensions to the configuration file; For complete details, see the <http://www.mono-project.com/Config> web page.

<dllmap> directive

You use the `dllmap` directive to map shared libraries referenced by `P/Invoke` in your assemblies to a different shared library.

This is typically used to map Windows libraries to Unix library names. The **dllmap** element takes two attributes:

- dll* This should be the same string used in the `DllImport` attribute, optionally prefixed with "i:" to indicate that the string must be matched in a case-insensitive way
- target* This should be the name of the library where the function can be found: this name should be suitable for use with the platform native shared library loading routines (`dlopen` etc.), so you may want to check the manpages for that, too.

<dllentry> directive

This directive can be used to map a specific `dll/function` pair to a different library and also a different function name. It should appear inside a **dllmap** element with only the `dll` attribute specified.

The **dllentry** element takes 3 attributes:

- dll* This is the target library, where the function can be found.
- name* This is the name of the function as it appears in the metadata: it is the name of the `P/Invoke` method.
- target* This is the name of the function to lookup instead of the name specified in the `P/Invoke` method.

Mapping based on operating system and cpu

Both the **dllmap** and the **dllentry** elements allow the following two attributes which make it easy to use a single configuration file and support multiple operating systems and architectures with different mapping requirements:

- os* This is the name of the operating system for which the mapping should be applied. Allowed values are: `linux`, `osx`, `solaris`, `freebsd`, `openbsd`, `netbsd`, `windows`, `aix`, `hpux`.
- cpu* This is the name of the architecture for which the mapping should be applied. Allowed values are: `x86`, `x86-64`, `sparc`, `ppc`, `s390`, `s390x`, `arm`, `mips`, `alpha`, `hppa`, `ia64`.

wordsize

This is the size of registers on the target architecture, it can be either 32 or 64.

The attribute value for both attributes can be a comma-separated list of the allowed values. Additionally, the first character may be a `'!'` to reverse the meaning. An attribute value of `"!windows,osx"`, for example, would mean that the entry is considered on all operating systems, except on Windows and OS X. No spaces are allowed in any part of the value.

Note that later entries will override the entries defined earlier in the file.

EXAMPLES

The following example maps references to the 'cygwin1.dll' shared library to the 'libc.so.6' file.

```
<configuration>
  <dllmap dll="i:cygwin1.dll" target="libc.so.6"/>
</configuration>
```

The library name in the DllImport attribute is allowed to be in any case variant, like the following examples:

```
[DllImport ("cygwin1.dll")]
[DllImport ("Cygwin1.dll")]
[DllImport ("cygwiN1.Dll")]
```

This one maps the following C# method:

```
[DllImport ("libc")]
static extern void somefunction ();
```

to **differentfunction** in **libdifferent.so** , but to the same function in the library **libanother.so** when running under the Solaris and FreeBSD operating systems.

```
<configuration>
  <dllmap dll="libc">
    <dllentry dll="libdifferent.so" name="somefunction" target="differentfunction" />
    <dllentry os="solaris,freebsd" dll="libanother.so" name="somefunction" target="differentfunction" />
  </dllmap>
</configuration>
```

SEE ALSO

mono(1), **monodis(1)**, **mint(1)**