mod-xslt FAQ

version: 0.1.12, relased on 2003/06/13 Copyright © 2002-2008 Carlo Contavalli
written by Carlo Contavalli, <ccontavalli at inscatolati.net>

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A complete copy of this license can be found at http://www.commedia.it/ccontavalli/license/FDL/en/index.html.

1 Build process 4

1.1 4
- Configure outputs the error

```
error: C preprocessor [...] fails sanity check
```

what can I do?

1.2 4
- I get the error:

```
Cannot load /usr/lib/apache/1.3/mod_xslt.so into server: /usr/lib/apache/1.3/mod_xslt.so:

undefined symbol: mxslt_debug_firstlevel
```

what can I do?

1.3 4
- After I run ``make'', I get an error similar to the following:

```
../../lib/modxslt-opr-cmp.c:179: `PCRE_NO_AUTO_CAPTURE' undeclared (first use in this

function) ../../lib/modxslt-opr-cmp.c:290: `PCRE_ERROR_MATCHLIMIT' undeclared (first use in

this function)
```

how do I compile mod-xslt? What am I doing wrong?

1.4 4
- I can't compile mod-xslt with the xxx compiler. The compiler complains about: .... What can I do?

1.5 4
- Configure complains about libxml2 not providing ``setGlobalState'', by printing a message like ``Your libxml2 does not provide a setGlobalState'', what's wrong?

1.6 4
- I am configuring mod-xslt using ``--with-sapi=apacheX''. However, configure detects and uses the apxs for some other version of apache, causing the build process to fail...

1.7 5
- You lyar! You told me mod-xslt was able to parse dynamically generated .xml documents, using dinamically generated .xsl files, but it doesn't work...

1.8 5
- Hem.. I don't like to have a comment in my HTML output stating that the page was parsed with mod-xslt. How can I disable it?

2 General usage issues 5

2.1 5
- I get almost a completely blank page with mod-xslt, but I get the correct output with xsltproc. What am I doing wrong?

2.2 5
- I often see in my logs errors like ``missing DTD''. However, I get this error only for certain documents. What's going on?

2.3 5
- The media-type attribute of xsl:output seems to be ignored by mod-xslt .. what can I do?

2.4 5
- Some errors in my .xslt documents cause a ``500'' status page while others cause a partial output (like half a html page) to be generated. How come? Is this a bug? Is this correct?

2.5 5
- My .xml document is not parsed, I get a 500 Server Error. I can see some wierd errors in the log about my DTDs, but the document is ``valid''. What can I do?

2.6 6
- I'm using XSLTSetStylesheet and XSLTDefaultStylesheet from my web server configuration files, but I keep getting errors like:

```
fatal - no <?xml-stylesheet or <?modxslt-stylesheet found in ...
```

what am I doing wrong?

2.7 6
- What's the difference between XSLTSetStylesheet and XSLTDefaultStylesheet?

2.8 6
- Uhm... in the log, I see tons of messages like:

```
mod_xslt: xsltApplyOneTemplate: fallback was not compiled
```

what does this mean?

2.9 7
- I'm having a hard time matching regular expressions which use the ``&'' character as separator (instead of the standard ``/''), what can I do?

2.10 7
- How can I change the mime type (Content-Type) of the documents returned back by mod-xslt?

2.11 7
- Uhm... file://, http://localhost/... why bothering with local://?

2.12 7
- I often get Segmentation Faults while using mod-xslt... what can I do?

2.13 8
- What's the difference between XSLTAddFilter and XSLTAddForce?

3 Sapi specific issues 8

3.1 Apache 1.3.x 8

3.1.1 8
- After installing mod-xslt, apache does not start outputting the error:

```
Syntax error on line xxxx of /etc/httpd/httpd.conf: Cannot load /etc/httpd/modules/
mod_xslt.so into server: libmodxslt0.so.0: cannot open shared object file: No such file or
directory
```

what can I do?

3.1.2 8
- Apache 1.3.x is not threaded... why is configure complaining about libxml2 not being threaded with a message like ``Your libxml2 is not compiled with multithreading support...''?

3.1.3 8
- How do I force mod-xslt to be compiled for apache 1.3.x?

3.1.4 8
- Can I use other modules linked with libxml/libxslt in apache 1.3?

3.2 Apache 2.x 9

3.2.1 9

- During start, apache outputs the error Can't locate API module structure: `mxslt_module' in ... undefined symbol ... what can I do?

3.2.2 9
- How do I force mod-xslt to be compiled for apache 2.x?

3.2.3 9
- Can I use apache 2.x multithreaded mpms? Is mod-xslt thread safe?

3.2.4 9
- Can I use other modules linked with libxml/libxslt in apache 2.x?

3.2.5 9
- It seems like my .xml documents get parsed twice, since I get errors about wrong .xml in the html output. What am I doing wrong?

4 General web server issues 10

4.1 10
- My xml files are correctly parsed by mod-xslt when the stylesheet is fetched from the local filesystem, while an error is printed if they are fetched using an ``http://'' or ``local://'' url. What can I do?
- Uhm... I get errors about a stylesheet not being loadable, but both the document and the stylesheet are valid. What can I do?
- I get a 500 error page, even if both the document and the stylesheet are valid. What's wrong?
- Apache is trying to parse my .xslt stylesheet as being a normal xml file, returning an error. What can I do?
- When parsing .xml files, mod-xslt complains with (apparentely) no reason for a ``subrequest failed'', and returns a 500 error page. What can I do?

4.2 10
- A document using local:// url doesn't seem to work, even though the .xml file and .xsl file look ok. I see strange error messages in the logs. What can I do?

4.3 10
- I have troubles using mod_xslt. I can see error reports in the error log, but I really can't understand what ``errno: x'' means...

4.4 10
- My browser complains about receiving an invalid .xml document. Shouldn't mod-xslt take care of parsing the .xml in html? Why does it still get .xml?
- My xml files are parsed correctly, but the generated output contains the xml header

```
<?xml version="1.0" ... ?>
```

How do I get rid of that header?
- When I try to access a document parsed by mod-xslt using my browser, I get an error like ``XML Parsing Error''. What can I do to solve the problem?

4.5 11
- I have installed mod_xslt correctly, but every time I try to access a particular page I get a ``Server Error'' (500 server status). How come?

4.6 11
- I don't usually have any problem using mod-xslt. However, I sometimes (apparently randomly) get a ``Server Error'' when trying to access a particular php/perl/... document. What can I do?

4.7 11
- I have some xml documents using a given xslt. The xslt exists and the paths are all correct, however, I keep getting a ``Server Error'', (500 server status), even if the documents are all valid. What can I do?

5 API 11

5.1 11
- My own application cannot make use of global MODXSLT variables... they either are not defined or the low level table handling functions return an error. How come?

5.2 11
- mxslt_error uses C99 constructs. What can I do to lessen their impact?

1 Build process

1.1 Configure outputs the error

```
error: C preprocessor [...] fails sanity check
```

what can I do?

Install a c++ compiler and the c++ library (g++, libstdc++, libstdc++-dev). This is due to the ``configure'' script being created with libtool <= 1.5.x, which tests for the availability of a c++ compiler. This is a known libtool problem, and future versions of modxslt configure scripts will be created using libtool >= 1.6. Either install a c++ compiler or recreate the ``configure'' script by yourself (using autoconf).

1.2 I get the error:

```
Cannot load /usr/lib/apache/1.3/mod_xslt.so into server: /usr/lib/apache/1.3/mod_xslt.so:
undefined symbol: mxslt_debug_firstlevel
```

what can I do?

You are probably using an older version of libmodxslt0, or two or more versions of libmodxslt0 are available on your system and apache is using the older one.
To solve the problem, make sure to run ``make install'' and to specify the correct --with-prefix to the configure script (usually, --with-prefix=/usr), and to remove any older libmodxslt0.

1.3 After I run ``make'', I get an error similar to the following:

```
../../lib/modxslt-opr-cmp.c:179: `PCRE_NO_AUTO_CAPTURE' undeclared (first use in this
function) ../../lib/modxslt-opr-cmp.c:290: `PCRE_ERROR_MATCHLIMIT' undeclared (first use
in this function)
```

how do I compile mod-xslt? What am I doing wrong?

Upgrade the libpcre installed on your system at least to version 4.5.

1.4 I can't compile mod-xslt with the xxx compiler. The compiler complains about: .... What can I do?

mod-xslt uses some ISO C99 constructs that are supported by very few compilers. You should:

- use gcc, which has supported them for quite a while now
- from your source tree, run ``./helpers/c99.sh'', a small script able to strip away all c99 constructs used by mod-xslt. After running ``c99.sh'', it should compile with any ANSI compiler.

If your compiler is not ANSI, go and get one. This is your problem.

1.5 Configure complains about libxml2 not providing ``setGlobalState'', by printing a message like ``Your libxml2 does not provide a setGlobalState'', what's wrong?

Althought libxml2 is thread safe and libxslt is not very thread sensitive, they both suffer some problems if used ``from modules'' by web servers like apache or apache2 (take a look to the thread on the libxml and libxslt mailing lists).
mod-xslt tries to lessen the impact of libxml/libxslt lack of support for modularized environments by trying to use some libxml2 internal functions.
The message just warns you that some compatibility layers have been enabled since some of the needed functions were missing. If you want to get rid of the warning, please patch the library (read README-Patches.sgml). Most of the times, you can just ignore this message since the ``compatibility layer'' will take care of everything.

1.6 I am configuring mod-xslt using ``--with-sapi=apacheX''. However, configure detects and uses the apxs for some other version of apache, causing the build process to fail...

You should also specify ``--with-apxs=...'' to force configure to use the right apxs. Without this parameter, configure will use the first apxs detected on your system, even if it is of the wrong apache version.

1.7 You lyar! You told me mod-xslt was able to parse dynamically generated .xml documents, using dinamically generated .xsl files, but it doesn't work...

By default, mod-xslt will try to fetch any external entity using directly the file system (which is kind of faster). However, if the file system is used, php, perl, ssi, are not parsed and the document is used raw as it is.

If you want to use dynamically generated stylesheets, use a http:// or local:// url, so mod-xslt will load the document by asking the web server for it.

If you want to use dynamically generated xml files, tell mod-xslt about that. To know how, take a deeper look to the manual. It seems like you need it.

1.8 Hem.. I don't like to have a comment in my HTML output stating that the page was parsed with mod-xslt. How can I disable it?

Latest version of mod-xslt don't add any comment at the end of your page. Just upgrade :). If you still see comments, they must be added by something else.

2 General usage issues

2.1 I get almost a completely blank page with mod-xslt, but I get the correct output with xsltproc. What am I doing wrong?

Try to update your libxslt. Version up to 1.1.11 have a problem in handling doctype-public and doctype-system attributes in the <xsl:output element. If you can't install/don't have any newer version, you can either:

- remove any of doctype-public or doctype-system attributes
- change the xsl:output method to html, instead of xml

2.2 I often see in my logs errors like ``missing DTD''. However, I get this error only for certain documents. What's going on?

mod-xslt doesn't care about the validity of documents. It just check for them to be ``well-formed''. However, DTDs may specify default values for some attributes or external entities, so they need to be loaded. However, loading one more additional file may slow things down, especially if DTDs are stored remotely. mod-xslt will thus try to load DTDs only if you specify the ``standalone="no"'' attribute in the xml declaration of your document.

Just set ``standalone'' to ``yes'' if you don't want mod-xslt to even try to load the DTD.

2.3 The media-type attribute of xsl:output seems to be ignored by mod-xslt .. what can I do?

There's currently a bug being investigated either in libxslt or mod-xslt2, such as xsl:output elements in imported documents are ignored...

A workaround is just to put xsl:output in the main .xsl stylesheet, instead of putting it in imported documents...

2.4 Some errors in my .xslt documents cause a ``500'' status page while others cause a partial output (like half a html page) to be generated. How come? Is this a bug? Is this correct?

No, it is not. You get a 500 error page when the xslt is not valid xml (it is not well-formed), contains invalid xsl instructions or when it cannot get parsed **at all** (when no output could be generated). You get as much of the output as possible when there is an error in your XPath expressions or a non fatal error in your templates. Take a look to your web server logs to know what exactly is going on.

2.5 My .xml document is not parsed, I get a 500 Server Error. I can see some wierd errors in the log about my DTDs, but the document is ``valid''. What can I do?

mod-xslt **does not** check for document validitity (it doesn't care about it). However, it does load external DTDs when ``<xml version... standalone="no"''.

When a locally stored DTD (== on a locally mounted disk) cannot be read for some reason an error is printed in the logs but parsing goes on. When the DTD is fetched from a remote URL, the error is fatal and parsing is stopped with a 500 error status.

If this is the case, you'll see in your logs something like:

```
mod_xslt: http://127.0.0.1/yaslt/fuffa/dtd/faq.dtd:1: mod_xslt: Content error in the
external subset mod_xslt: <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN"> mod_xslt:
^ mod_xslt: http://127.0.0.1/yaslt/fuffa/dtd/faq.dtd:1: mod_xslt: Extra content at
the end of the document mod_xslt: <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
mod_xslt: ^
```

Does this look familiar to you? This is not my fault. libxml2 http client does not check for the status of remote servers, and tries to parse any returned document, even when the returned document is an error page (like those returned by a 400 or 500 status).
However, when a DTD is not available nothing bad happens, while if it is available but it is not a valid DTD (contains junk, trash, fuffa, html or something that does not make it a valid DTD), a fatal error arise.
At time of writing, and according to the libxml (2.5.9) mailing list, the lack of check of the status of remote servers and the lack of an API to allow users to decide how to handle error pages is a ``feature''.
Furthermore, I couldn't find any good way to disable this feature, that didn't involve patching the library or rewriting the whole http client (which is something I'm not going to do).
If this problem really annoies you, patch your library and let your voice be heard by the upstream team.

2.6 I'm using XSLTSetStylesheet and XSLTDefaultStylesheet from my web server configuration files, but I keep getting errors like:

```
fatal - no <?xml-stylesheet or <?modxslt-stylesheet found in ...
```

what am I doing wrong?

The most common source of errors is the MimeType specified with XSLTSetStylesheet or XSLTDefaultStylesheet. Infact, both parameters must be followed by the mime type **of the document the stylesheet is to be applied to**.
Additionally, mime types are configurable for most web servers, so it is quite easy to specify the wrong mime type for the given document. As an example, if we want to apply stylesheet.xsl to all .xml documents, we need to use something like: [...] XSLTSetStylesheet application/xml stylesheet.xsl [...] or XSLTSetStylesheet text/xml stylesheet.xsl to tell your web server you want all documents of type text/xml and application/xml to be transformed by applying stylesheet.xsl... Web servers like apache use parameters like AddType to decide which mime type to use for every document, or files like /etc/mime.types.
Remember: XSLTSetStylesheet forces the usage of a given stylesheet for a given kind of documents, where the document kind is indicated thanks to its mime type. XSLTDefaultStylesheet provides a stylesheet to be used in case there is no XSLTSetStylesheet and no stylesheet is specified in the document itself.

2.7 What's the difference between XSLTSetStylesheet and XSLTDefaultStylesheet?

Remember: XSLTSetStylesheet forces the usage of a given stylesheet for a given kind of documents, where the document kind is indicated thanks to its mime type. XSLTDefaultStylesheet provides a stylesheet to be used in case there is no XSLTSetStylesheet and no stylesheet is specified in the document.

2.8 Uhm... in the log, I see tons of messages like:

```
mod_xslt: xsltApplyOneTemplate: fallback was not compiled
```

what does this mean?

It just means you're using a <xsl:fallback... tag under a <mod-xslt:header tag, or in a ``strange'' place (like as a top element).
I believe it to be a libxml2 bug, which I already took care of reporting.
However, at time of writing it has not been corrected (libxml 2.5.9).
To avoid the problem, that under some conditions may cause a Segmentation Fault, you can rely on some dirty hacks provided by mod-xslt.

The Segmentation Fault is caused by yet another bug that sometimes combines badly with this bug: the ``xsltApplyOneTemplate: fallback ...'' error message is printed by libxml by calling the application error handler by using the arguments of the debug handler.
This latest problem is detected by mod-xslt code most of the times. However, to get rid of both problems forever, you should 1 - patch the library to avoid error messages to be printed when the fallback tag is correctly used 2 - patch the library to avoid the error handler to be called with the debug arguments.
If you don't want to patch the library, you can enable two dirty hacks in mod-xslt: the first one will try to remove the fallback nodes where libxml would illegally comply, while the second one will enable the debug handler with the same arguments as the error handler, in order to always avoid the SEGFAULT.
Read README-Patches.sgml for more details.

2.9 I'm having a hard time matching regular expressions which use the ``&'' character as separator (instead of the standard ``/''), what can I do?

The ``&'' character has a special meaning in XML. Since it is used inside the attribute of a tag, it is interpreted by the xml parsing library as a special character. If you want to use ``&'' as a separator, you need to use a regular expression like ``&[a-z]+[0-9]{2}&i'' which is exactly equivalent to ``/[a-z]+[0-9]{2}/i''.

2.10 How can I change the mime type (Content-Type) of the documents returned back by mod-xslt?

You must use the attribute ``media-type'' of the ``<xsl:output...'' tag, much like in the example below: <xsl:output method="html" media-type="text/fuffa" indent="no" encoding="UTF8"> where the output mime-type is set to ``text/fuffa''.

2.11 Uhm... file://, http://localhost/... why bothering with local://?

- **file://** - asks for a stylesheet or for an entity to be loaded directly by mod-xslt by reading the local file system. Path will be calculated starting from the root of the file system if a third ``/'' is specified, otherwise it will be taken as a relative path from the current directory (the real directory of the xml file being parsed). **By default**, if no scheme is specified, the path is assumed to be relative to the file system.
- **http://localhost/** - asks for a stylesheet or for an entity to be taken from the specifyed host (in this case localhost) using the http protocol.
- **local://** - is similar to both ``http://'' and ``file://'', in the sense that the specified file will be accessed by using the http protocol, but... the name of the server will be that of the virtual host that received the request, and if the path doesn't start with a ``/'' (leading to something like local:///file.html), the url will be considered as relative to the current page. This is useful when you have, for example, php scripts generating .xsl files or .DTD.

Reading the **README** for more details is strongly encouraged. As an example, imagine a browser requesting ``www.site.com/your/path/to/file.xml''. If, in file.xml, you have ``local://file.xsl'', file.xsl will be accessed using http as if it was ``http://www.site.com/your/path/to/file.xsl''. If instead we had ``local:///file.xsl'', the address would be: ``http://www.site.com/file.xsl''. Note that, for various concerns, the name of the virtual host (www.site.com) is taken from the ServerName parameter in apache.

2.12 I often get Segmentation Faults while using mod-xslt... what can I do?

There's **only one known** condition in which case mod-xslt will crash with segmentation fault, and it's not mod-xslt fault.
The problem is caused by libxslt calling the ``error'' handler with the arguments of the ``debug'' handler under certain conditions. mod-xslt cannot really detect when this error verifies, but some ``parachutes'' have been written to avoid this problem most of the times.
To get rid of the problem, you should patch libxslt.
However, keep in mind that the highlited segfault will always be triggered by an error in your xslt file, error that is ``printed uncorrectly'' by the libxslt library.
If you don't want to patch libxslt, you can compile mod-xslt with ``--enable-libxslt-hack''. By enabling this option, mod-xslt will enable libxslt debugging by specifying an handler which accepts the same arguments as the error one.

However, with debugging enabled, the debug function will always be called (even if the output is discarded), and may possibly slow things down.
If this error really bothers you, you'd be better patch the library.
For more details, please read ``README-Patches''.

2.13 What's the difference between XSLTAddFilter and XSLTAddForce?

XSLTAddFilter tells mod-xslt to watch a given mime type to see if, after being executed by apache, it outputs xml code to be parsed.
XSLTAddForce tells mod-xslt to parse anything sent as output by apache while parsing the given mime type.
The only difference between the two is that the first one looks at the output mime type, and ignores the document if it is not text/xml, while the second one parses anything the script may send.

3 Sapi specific issues

3.1 Apache 1.3.x

3.1.1 After installing mod-xslt, apache does not start outputting the error:

```
Syntax error on line xxxx of /etc/httpd/httpd.conf: Cannot load /etc/httpd/modules/
mod_xslt.so into server: libmodxslt0.so.0: cannot open shared object file: No such file or
directory
```

what can I do?

When you configured and installed mod-xslt, you didn't specify the correct ``--prefix'' or ``--libdir''.
This means that everything got compiled and installed correctly, but that your system is not able to find the installed libraries since they have not been placed in the directories it is looking into.
In order to solve the problem, either:

- specify something like --prefix=/usr --exec-prefix=/usr, to get the libraries installed in /usr/lib, one of the directories most system look into to search for libraries (thanks to Boyan Dzambazov).
- specify something like --libdir=/usr/lib, to manually force the place where you want mod-xslt libraries to be installed.
- on linux, add the line ``/usr/local/lib'' into the configuration file /etc/ld.so.conf and run the command ``ldconfig'', to tell your linux system to also look in /usr/local/lib.

3.1.2 Apache 1.3.x is not threaded... why is configure complaining about libxml2 not being threaded with a message like ``Your libxml2 is not compiled with multithreading support...''?

If you are not using any apache module that links to libxml2 (you can check by using ``ldd'' on each apache module you are using) beshide mod-xslt itself, you should ignore this warning.
In any other case, mod-xslt will try to lessen the impact of libxml2 lack of support for modularized applications by using some libxml2 internal functions available only when libxml2 is compiled with threading support.
If you want to use other apache modules that are linked to libxml2, you should really make sure your libxml2 has been compiled with threading support.
Even with threading support, however, some problems may arise, depending on how the other modules make use of libxml or libxslt.

3.1.3 How do I force mod-xslt to be compiled for apache 1.3.x?

Use ``configure --with-sapi='apache1' --with-apxs='/path/to/apxs'''. You can find where ``apxs'' is installed on your system by running something like ``whereis apxs'', ``which apxs'', ``locate apxs'' or ``find / -name apxs -perm +0100''.
Without the ``--with-apxs'' parameter, configure will try to use the first apxs found in some common places.

3.1.4 Can I use other modules linked with libxml/libxslt in apache 1.3?

Great care has been taken to allow mod-xslt to run even when some other apache 1.3 module is linked with mod-xslt. Unluckily, libxml and libxslt were not made with modularized softwares in mind and keep some variables shared and ``secret'', so I can't do much about them.
However, if you enable threading support in libxml2, mod-xslt will be able to ``copy'' and ``restore'' most of the important variables and to work with ``most'' of other apache modules making use of libxml (as long as they don't change libxml memory handlers).
libxslt condition is worse: althought it uses much fewer global variables, it doesn't have any additional support for multithreading. However, the lower number of global variables should make it usable in most condition, even if errors won't be handled correctly (will probably go to the wrong place).

3.2 Apache 2.x

3.2.1 During start, apache outputs the error Can't locate API module structure: `mxslt_module' in ... undefined symbol ... what can I do?

Verify the correctness of your LoadModule directive for mod-xslt in your apache 2.x configuration file. Probably, you specified the wrong path: try to run something like ``apxs -q LIBEXECDIR'' or ``apxs2 -q LIBEXECDIR'' to know the correct one to use on your system.

3.2.2 How do I force mod-xslt to be compiled for apache 2.x?

Use ``configure --with-sapi='apache2' --with-apxs='/path/to/apxs2'''. You can find where ``apxs2'' is installed on your system by running something like ``whereis apxs'', ``which apxs'', ``locate apxs'' or ``find / -name apxs -perm +0100''.
Without the ``--with-apxs'' parameter, configure will try to use the first apxs found in some standard places.
Note that depending on how apache2 was compiled on your system, ``apxs'' may as well be named ``apxs2''.

3.2.3 Can I use apache 2.x multithreaded mpms? Is mod-xslt thread safe?

Great care has been taken to keep mod-xslt thread safe, althrought some of the libraries it uses are not. Most of the problems were due to libxslt not supporting multithreaded applications (at time of writing - since mod-xslt needs to change the error context handler for every apache request and requests are processed paralleley by multiple threads).
The problem has been worked around by using some tsd variables, and you should not have any problem. However, you may see your error messages going to the wrong place if you ever decide to use mod-xslt with any other apache module linked with libxslt.

3.2.4 Can I use other modules linked with libxml/libxslt in apache 2.x?

Great care has been taken to allow mod-xslt to run even when some other apache 2.x module is linked with mod-xslt. Unluckily, libxml and libxslt were not made with modularized softwares in mind and keep some variables shared (among modules) and ``secret'', so I can't do much about that.
However, if you enable support for threading in libxml2, mod-xslt will be able to ``copy'' and ``restore'' most of the important variables and to work with ``most'' of other apache modules making use of libxml (as long as they don't change libxml memory handler).
libxslt condition is worse: althought it uses much fewer global variables, it doesn't have any additional support for multithreading. However, the lower number of global variables should make it usable in most condition, even if errors won't be handled correctly (will probably go to the wrong place or cause a segfault, let me know if it happens).

3.2.5 It seems like my .xml documents get parsed twice, since I get errors about wrong .xml in the html output. What am I doing wrong?

Take a look to your apache.conf or httpd.conf, whatever it is called on your system. You probably specified more than one ``AddOutputFilter'', ``SetOutputFilter'' or ``AddOutputFilterByType''.
You need to specify only **one** of those instructions, otherwise your xml files will be parsed by mod-xslt multiple times, one for each of the directives you used. And while at the first pass the html

will be generated, during the second pass it would try to transform the html into something else, leading to the errors you are seeing.

4 General web server issues

4.1 My xml files are correctly parsed by mod-xslt when the stylesheet is fetched from the local filesystem, while an error is printed if they are fetched using an ``http://'' or ``local://'' url. What can I do? Uhm... I get errors about a stylesheet not being loadable, but both the document and the stylesheet are valid. What can I do? I get a 500 error page, even if both the document and the stylesheet are valid. What's wrong? Apache is trying to parse my .xslt stylesheet as being a normal xml file, returning an error. What can I do? When parsing .xml files, mod-xslt complains with (apparentely) no reason for a ``subrequest failed'', and returns a 500 error page. What can I do?

In the log, you'll probably see something like:

```
fatal - couldn't load stylesheet for: file.xsl http://127.0.0.1/yaslt/xslt/file.xsl:1:
Space required after the Public Identifier <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML
2.0//EN">
```

And, if you look carefully to the url of your stylesheet, you'll see that it is loaded from a ``http://'' or ``local://'' url.

This error usually shows up when the xsl file is stored in a mod-xslt handled directory, and is due to apache (correctly) believing that both .xml and .xsl files are of type ``text/xml''.

When the xml file is parsed, the xsl file is fetched, but, as being fetched using a remote url, it is parsed again by apache. apache verifyes the file is of type ``text/xml'', and calls mod-xslt again to parse the stylesheet. However, the stylesheet cannot be parsed since it does not contain xml-stylesheet or modxslt-stylesheet directive, causing the sub-request to fail and mod-xslt to return a 500 error page, which, in turn, cannot be used by mod-xslt as a .xslt stylesheet (as not being even a valid .xml page).

To solve the problem, either disable mod-xslt in the directories where you store .xsl files or tell apache ``.xsl'' files are of type ``text/xsl'', using something like: AddType text/xsl .xsl

4.2 A document using local:// url doesn't seem to work, even though the .xml file and .xsl file look ok. I see strange error messages in the logs. What can I do?

For apache, local:// gets replaced with the name of the server the request came from. Under certain circumstances, the server name used is the one specified by the ServerName parameter in httpd.conf file. If that parameter is wrongly configured, than the local:// request might fail. The error you see in the log will be the one corresponding to the http error encountered (most likely, "No such file or directory", meaning that the file is not there).

4.3 I have troubles using mod_xslt. I can see error reports in the error log, but I really can't understand what ``errno: x'' means...

Take the number and run the command (from your console) ``modxslt-perror x''. It will tell you what the problem is. I couldn't use ``strerror'', since this function is not thread safe on many systems.

4.4 My browser complains about receiving an invalid .xml document. Shouldn't mod-xslt take care of parsing the .xml in html? Why does it still get .xml? My xml files are parsed correctly, but the generated output contains the xml header

```
<?xml version="1.0" ... ?>
```

How do I get rid of that header? When I try to access a document parsed by mod-xslt using my browser, I get an error like ``XML Parsing Error''. What can I do to solve the problem?

In your stylesheet, add the following xsl instruction: <xsl:output method="html" omit-xml-declaration="yes" encoding="UTF-8" indent="no" doctype-public="-//W3C//DTD XHTML 1.0 Strict//EN" doctype-system="http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd" /> Unless you use the above xsl instructions, the xsl processor assumes that the output will still be valid xml,

and prepends the correct xml headers. The browser, when receiving the document, thinks that the document is still .xml, and tries to parse it.

4.5 I have installed mod_xslt correctly, but every time I try to access a particular page I get a ``Server Error'' (500 server status). How come?

For security reasons mod-xslt won't disclose any information about the errors in your xml pages, and will return a 500 error page when:

- the file being parsed by mod-xslt is not a valid XML file
- a valid XSLT cannot be found
- the XSLT is having troubles translating the given XML file

When you ask mod_xslt to parse a file, that file will either be parsed or an error given back to the user. mod-xslt won't give away the source of your xml files or any information about the errors they contain (unless you ask for it). If you want to know what caused mod-xslt to return the error, read the error.log of your server.
If you want to return the xml source, just disable mod-xslt for the given file, using

- in apache 1.3.x: the apache <file> directive or set the mime type to text/plain.

You can also use the <modxslt-stylesheet pi without any href to tell mod-xslt you don't want any xslt to be used.

4.6 I don't usually have any problem using mod-xslt. However, I sometimes (apparently randomly) get a ``Server Error'' when trying to access a particular php/perl/... document. What can I do?

I got several questions like this, and most of the times they were related to the language parser generating errors in the ``html'' output. Consider php as an example: when an error verifies in your code, like a wrong mysql query or something similar, php inserts an error text in what it thinks to be an html document. However, the document is not html and the error text is not written using valid xml, causing mod-xslt to complain about the validity of the xml document.
However, the xml will be invalid only when your code encounters an error, leading to ``random'' errors from ``mod-xslt''. To trace the problem down, I suggest you take a good look to the error log and disable the unlinking of temporary files or enable the store of ``intermediate files'' (depends on the SAPI being used). In the temporary file, you will get the xml file generated by your script that mod-xslt had to parse.

4.7 I have some xml documents using a given xslt. The xslt exists and the paths are all correct, however, I keep getting a ``Server Error'', (500 server status), even if the documents are all valid. What can I do?

This problem often shows up when the privileges of the web server do not allow it to access the xslt document or any external entity required by the xml document. In the error log, you should see something like:

```
failed to load external entity "name.xsl" fatal - couldn't apply stylesheet to: /tmp/
name.tmp (file.xml)
```

As shown in the messages above, libxslt is not able to open ``name.xsl''. In this case ``name.xsl'' either does not exist or cannot be opened by the web server for privilege problems (verify the .xsl file can be read by the web server!).

5 API

5.1 My own application cannot make use of global MODXSLT variables... they either are not defined or the low level table handling functions return an error. How come?

Did you call mxslt_xml_preinit? Did you call it before your application started to spawn threads? Well, make sure you did...

5.2 mxslt_error uses C99 constructs. What can I do to lessen their impact?

Take care of writing the call to mxslt_error on one, single line. If the line is too long, just live with it. The c99-strip script cannot handle function calls that eat up multiple lines.