

The Complete Guide to Adding Error Handling to Your ColdFusion Application

Posted At : December 5, 2007 2:36 PM | Posted By : Raymond Camden

Related Categories: ColdFusion

I've done more than a few posts recently about error handling and robust exception information, so I thought I'd write up one blog entry that can serve as a nice guide for folks to bookmark. By using "Complete" in the title, I've also virtually assured that I will forget something critical, so please, send suggestions for what I've missed. Let's get started.

What is the point of this guide? Unless you are a perfect code, there is a chance that your ColdFusion application will have errors in it. The question is - what are you doing with your errors? What was the last error that occurred on your site? If I asked you how many errors your site threw yesterday, could you answer with 100% complete accuracy? I'd be willing to bet most people would say no. Shoot, I know I couldn't answer that. So what do we do to help resolve this issue?

The first thing I want you to do is to create an error. Create a new CFM file named `errortest.cfm`. Insert one line into it:

```
<cfoutput>#xfhdshsdhj#</cfoutput>
```

The point of this template is to create an error. Unless you actually have a variable defined with that ugly name, the template will error out. Upload it to your server and run it with your browser.

One of three things will happen:

1) You will get an error with the full path shown, like below. This shows that you have not added error management to your application, and that you have Enabled Robust Exception information in your ColdFusion Administrator. Stop reading this blog entry, go to your Admin, and disable it. Now. (Please note that this guide is intended for a production machine environment. Obviously you can keep this setting on in development. I do.)

The web site you are accessing has experienced an unexpected error. Please contact the website administrator.

The following information is meant for the website developer for debugging purposes.

Error Occurred While Processing Request

Variable XFHDSDHJ is undefined.

The error occurred in `/Users/ray/Documents/Web Sites/webroot/test.cfm`: line 1
 1 : `<cfoutput>#xfhdshsdhj#</cfoutput>`

Resources:

- Check the [ColdFusion documentation](#) to verify that you are using the correct syntax.
- Search the [Knowledge Base](#) to find a solution to your problem.

Browser: Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10.5; en-US; rv:1.9b1)
 Gecko/2007110903 Firefox/3.0b1

Remote Address: 127.0.0.1

Referrer:

Date/Time: 05-Dec-07 09:17 AM

Stack Trace
 at cftest2ecfm324944294.runPage(/Users/ray/Documents/Web Sites/webroot/test.cfm:1)

coldfusion.runtime.UndefinedVariableException: Variable XFHDSDHJ is undefined.
 at coldfusion.runtime.CFjspPage._get(CFjspPage.java:315)

2) You will get an error with no path shown. This is slightly better. It still shows that you haven't added error management to your application though.

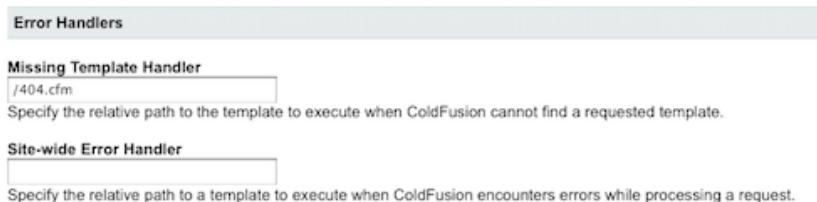
The web site you are accessing has experienced an unexpected error. Please contact the website administrator.



3) You get an "error page". By "error page" I mean a page saying that an error occurred, but displayed in your standard site layout, or close to it. That's good. You should still continue to read though as I'm going to discuss things that should normally be in an error handler.

Ok, before going on - delete the page you just uploaded. I'd be willing to bet a good 40-50% of us upload test.cfm type files to our servers and forget to delete them. I've never done that. Really.

So - lets talk error handling now. You have two high level options. The first is to set an error template in the ColdFusion administrator:



If you specify a template here, than ColdFusion will run the template when an error occurs. In general though I wouldn't recommend setting the template here.

Your second option - and what I recommend - is to handle the errors specifically in your application. How? If you are on an older ColdFusion server and only have access to Application.cfm, then you want to use the CFERROR tag. You *can* use these tags in Application.cfc as well, but I recommend onError for that. I'm going to cover both though.

A full description of cferror may be found in the [docs](#). I'm going to give you the Ray Camden quickie guide to it. The cferror basically lets you specify an template to run when a type of error occurs. There are 3 main types of errors it monitors: Exception, Request, and Validation. Forget about validation. Don't use it. Exception is the type of error we are most familiar with. It is what I call the 'basic' error. Request is the more serious error. It can occur when your error handler itself screws up. I call this the "Oh S****" error (seriously). It has special restrictions on it I'll discuss in a second. First though, let's look at the syntax you would use in your Application.cfm file:

```
<cfapplication name="rayrules">
```

```
<cferror type="exception" template="error.cfm">
<cferror type="request" template="error_request.cfm">
```

As you can see - it's pretty simple stuff. I basically said - for the Exception (again, think 'normal' error), run error.cfm. For the Oh Crap error, run error_request.cfm.

If you run any CFM again - you will immediately get an error stating that these files do not exist. You should create blank ones for the time being.

Now for the details. The error.cfm template is a normal CFM page. But it has access to a special variable: ERROR. This variable is a structure that contains a lot of information about the error. What you get there will also depend on the error. SQL errors for example will have different values in the structure than a simple runtime error. Let's just do a quick dump.

```
<cfdump var="#error#">
```

If you run your error test again (you may have to reupload it if you deleted it like I suggested, just don't forget to remove it later!), you will see something like this:

struct																																			
Browser	Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10.5; en-US; rv:1.9b1) Gecko/2007110903 Firefox/3.0b1																																		
DateTime	{ts '2007-12-05 09:41:49'}																																		
Diagnostics	Variable XFHDSDHJ is undefined. The error occurred on line 1.																																		
GeneratedContent																																			
HTTPReferer	{empty string}																																		
Mailto	{empty string}																																		
Message	Variable XFHDSDHJ is undefined.																																		
QueryString	{empty string}																																		
RemoteAddress	127.0.0.1																																		
RootCause	<table border="1"> <thead> <tr> <th colspan="2">struct</th> </tr> </thead> <tbody> <tr> <td>Detail</td> <td>{empty string}</td> </tr> <tr> <td>ErrNumber</td> <td>0</td> </tr> <tr> <td>Message</td> <td>Variable XFHDSDHJ is undefined.</td> </tr> <tr> <td>StackTrace</td> <td>coldfusion.runtime.UndefinedVariableException: Variable XFHDSDHJ is undefined. at coldfusion.runtime.CFjspPage._getColdfusion.runtime.CFjspPage.invoke(CFjspPage.java:192) at coldfusion.tagext.lang.IncludeTag.doStartTag(IncludeTag.java: coldfusion.filter.MonitoringFilter.invoke(MonitoringFilter.java:40) at coldfusion.filter.PathFilter.invoke(PathFilter.java:86) at coldfusion.filter.BrowserFilter.invoke(BrowserFilter.java:38) at coldfusion.filter.NoCacheFilter.invoke(NoCacheFilter.java:46) coldfusion.bootstrap.BootstrapServlet.service(BootstrapServlet.java:89) at jrun.servlet.FilterChain.doFilter(FilterChain.java:jrun.servlet.FilterChain.service(FilterChain.java:101) at jrun.servlet.ServletInvoker.invoke(ServletInvoker.java:106) at jrun.jrun.servlet.jrpp.RunProxyService.invokeRunnable(RunProxyService.java:203) at jrunx.scheduler.ThreadPool\$ThreadThro</td> </tr> <tr> <td>TagContext</td> <td> <table border="1"> <thead> <tr> <th colspan="2">array</th> </tr> </thead> <tbody> <tr> <td>1</td> <td> <table border="1"> <thead> <tr> <th colspan="2">struct</th> </tr> </thead> <tbody> <tr> <td>COLUMN</td> <td>0</td> </tr> <tr> <td>ID</td> <td>??</td> </tr> <tr> <td>LINE</td> <td>1</td> </tr> <tr> <td>RAW_TRACE</td> <td>at cftest2ecfm324944294.runPage(/Users/ray/Documents/Web Sites/webroot/test.cfm:1)</td> </tr> <tr> <td>TEMPLATE</td> <td>/Users/ray/Documents/Web Sites/webroot/test.cfm</td> </tr> <tr> <td>TYPE</td> <td>CFML</td> </tr> </tbody> </table> </td> </tr> </tbody> </table> </td> </tr> <tr> <td>Type</td> <td>Expression</td> </tr> <tr> <td>name</td> <td>XFHDSDHJ</td> </tr> </tbody> </table>	struct		Detail	{empty string}	ErrNumber	0	Message	Variable XFHDSDHJ is undefined.	StackTrace	coldfusion.runtime.UndefinedVariableException: Variable XFHDSDHJ is undefined. at coldfusion.runtime.CFjspPage._getColdfusion.runtime.CFjspPage.invoke(CFjspPage.java:192) at coldfusion.tagext.lang.IncludeTag.doStartTag(IncludeTag.java: coldfusion.filter.MonitoringFilter.invoke(MonitoringFilter.java:40) at coldfusion.filter.PathFilter.invoke(PathFilter.java:86) at coldfusion.filter.BrowserFilter.invoke(BrowserFilter.java:38) at coldfusion.filter.NoCacheFilter.invoke(NoCacheFilter.java:46) coldfusion.bootstrap.BootstrapServlet.service(BootstrapServlet.java:89) at jrun.servlet.FilterChain.doFilter(FilterChain.java:jrun.servlet.FilterChain.service(FilterChain.java:101) at jrun.servlet.ServletInvoker.invoke(ServletInvoker.java:106) at jrun.jrun.servlet.jrpp.RunProxyService.invokeRunnable(RunProxyService.java:203) at jrunx.scheduler.ThreadPool\$ThreadThro	TagContext	<table border="1"> <thead> <tr> <th colspan="2">array</th> </tr> </thead> <tbody> <tr> <td>1</td> <td> <table border="1"> <thead> <tr> <th colspan="2">struct</th> </tr> </thead> <tbody> <tr> <td>COLUMN</td> <td>0</td> </tr> <tr> <td>ID</td> <td>??</td> </tr> <tr> <td>LINE</td> <td>1</td> </tr> <tr> <td>RAW_TRACE</td> <td>at cftest2ecfm324944294.runPage(/Users/ray/Documents/Web Sites/webroot/test.cfm:1)</td> </tr> <tr> <td>TEMPLATE</td> <td>/Users/ray/Documents/Web Sites/webroot/test.cfm</td> </tr> <tr> <td>TYPE</td> <td>CFML</td> </tr> </tbody> </table> </td> </tr> </tbody> </table>	array		1	<table border="1"> <thead> <tr> <th colspan="2">struct</th> </tr> </thead> <tbody> <tr> <td>COLUMN</td> <td>0</td> </tr> <tr> <td>ID</td> <td>??</td> </tr> <tr> <td>LINE</td> <td>1</td> </tr> <tr> <td>RAW_TRACE</td> <td>at cftest2ecfm324944294.runPage(/Users/ray/Documents/Web Sites/webroot/test.cfm:1)</td> </tr> <tr> <td>TEMPLATE</td> <td>/Users/ray/Documents/Web Sites/webroot/test.cfm</td> </tr> <tr> <td>TYPE</td> <td>CFML</td> </tr> </tbody> </table>	struct		COLUMN	0	ID	??	LINE	1	RAW_TRACE	at cftest2ecfm324944294.runPage(/Users/ray/Documents/Web Sites/webroot/test.cfm:1)	TEMPLATE	/Users/ray/Documents/Web Sites/webroot/test.cfm	TYPE	CFML	Type	Expression	name	XFHDSDHJ
struct																																			
Detail	{empty string}																																		
ErrNumber	0																																		
Message	Variable XFHDSDHJ is undefined.																																		
StackTrace	coldfusion.runtime.UndefinedVariableException: Variable XFHDSDHJ is undefined. at coldfusion.runtime.CFjspPage._getColdfusion.runtime.CFjspPage.invoke(CFjspPage.java:192) at coldfusion.tagext.lang.IncludeTag.doStartTag(IncludeTag.java: coldfusion.filter.MonitoringFilter.invoke(MonitoringFilter.java:40) at coldfusion.filter.PathFilter.invoke(PathFilter.java:86) at coldfusion.filter.BrowserFilter.invoke(BrowserFilter.java:38) at coldfusion.filter.NoCacheFilter.invoke(NoCacheFilter.java:46) coldfusion.bootstrap.BootstrapServlet.service(BootstrapServlet.java:89) at jrun.servlet.FilterChain.doFilter(FilterChain.java:jrun.servlet.FilterChain.service(FilterChain.java:101) at jrun.servlet.ServletInvoker.invoke(ServletInvoker.java:106) at jrun.jrun.servlet.jrpp.RunProxyService.invokeRunnable(RunProxyService.java:203) at jrunx.scheduler.ThreadPool\$ThreadThro																																		
TagContext	<table border="1"> <thead> <tr> <th colspan="2">array</th> </tr> </thead> <tbody> <tr> <td>1</td> <td> <table border="1"> <thead> <tr> <th colspan="2">struct</th> </tr> </thead> <tbody> <tr> <td>COLUMN</td> <td>0</td> </tr> <tr> <td>ID</td> <td>??</td> </tr> <tr> <td>LINE</td> <td>1</td> </tr> <tr> <td>RAW_TRACE</td> <td>at cftest2ecfm324944294.runPage(/Users/ray/Documents/Web Sites/webroot/test.cfm:1)</td> </tr> <tr> <td>TEMPLATE</td> <td>/Users/ray/Documents/Web Sites/webroot/test.cfm</td> </tr> <tr> <td>TYPE</td> <td>CFML</td> </tr> </tbody> </table> </td> </tr> </tbody> </table>	array		1	<table border="1"> <thead> <tr> <th colspan="2">struct</th> </tr> </thead> <tbody> <tr> <td>COLUMN</td> <td>0</td> </tr> <tr> <td>ID</td> <td>??</td> </tr> <tr> <td>LINE</td> <td>1</td> </tr> <tr> <td>RAW_TRACE</td> <td>at cftest2ecfm324944294.runPage(/Users/ray/Documents/Web Sites/webroot/test.cfm:1)</td> </tr> <tr> <td>TEMPLATE</td> <td>/Users/ray/Documents/Web Sites/webroot/test.cfm</td> </tr> <tr> <td>TYPE</td> <td>CFML</td> </tr> </tbody> </table>	struct		COLUMN	0	ID	??	LINE	1	RAW_TRACE	at cftest2ecfm324944294.runPage(/Users/ray/Documents/Web Sites/webroot/test.cfm:1)	TEMPLATE	/Users/ray/Documents/Web Sites/webroot/test.cfm	TYPE	CFML																
array																																			
1	<table border="1"> <thead> <tr> <th colspan="2">struct</th> </tr> </thead> <tbody> <tr> <td>COLUMN</td> <td>0</td> </tr> <tr> <td>ID</td> <td>??</td> </tr> <tr> <td>LINE</td> <td>1</td> </tr> <tr> <td>RAW_TRACE</td> <td>at cftest2ecfm324944294.runPage(/Users/ray/Documents/Web Sites/webroot/test.cfm:1)</td> </tr> <tr> <td>TEMPLATE</td> <td>/Users/ray/Documents/Web Sites/webroot/test.cfm</td> </tr> <tr> <td>TYPE</td> <td>CFML</td> </tr> </tbody> </table>	struct		COLUMN	0	ID	??	LINE	1	RAW_TRACE	at cftest2ecfm324944294.runPage(/Users/ray/Documents/Web Sites/webroot/test.cfm:1)	TEMPLATE	/Users/ray/Documents/Web Sites/webroot/test.cfm	TYPE	CFML																				
struct																																			
COLUMN	0																																		
ID	??																																		
LINE	1																																		
RAW_TRACE	at cftest2ecfm324944294.runPage(/Users/ray/Documents/Web Sites/webroot/test.cfm:1)																																		
TEMPLATE	/Users/ray/Documents/Web Sites/webroot/test.cfm																																		
TYPE	CFML																																		
Type	Expression																																		
name	XFHDSDHJ																																		

I'm not going to list every field - again, check the [cferror syntax doc](#) for that. The item you will most care about normal is cferror.message. This is the simplest representation of the error and will be most useful for logging. The diagnostics value gives more detail including a line number which is handy during development.

So what now? Well first off - you probably don't want your public site showing a dump. Let's begin by outputting a nice message to the user. Here is the new version of error.cfm:

We are so sorry. Something went wrong. We are working on it now.

Obviously you can mark that up with nice HTML, use your custom tag layout wrapper, etc. This handles letting the user know something bad happened. At bare minimum, this is better than showing naked errors to the user, but we should do something with the error. I recommend two things.

Log the error. For some odd reason, ColdFusion will nicely log an unhandled error, but will not log a handled error. I believe Blue Dragon logs the error anyway. Since ColdFusion won't log it, we should:

```
<cflog file="myapperrorlog" text="#error.message# -
#error.diagnostics#">
```

Note that I've specified the message and diagnostics variable. This is a bit of a duplication since diagnostics information will have the same information as message, but I like the shortness of the message value. You could log more than this obviously, but since this is a log file, we don't want to overdue it here.

The next thing we should do is email the error. This is something I've covered before on the blog, so some of you may know my feelings on this already, but what I typically do is email me the entire error structure. I also include other structures so I can see what else was going on:

```
<cfsavecontent variable="errortext">
<cfoutput>
An error occurred:
http://#cgi.server_name##cgi.script_name##cgi.query_string#<br />
Time: #dateFormat(now(), "short")# #timeFormat(now(), "short")#<br />

<cfdump var="#error#" label="Error">
<cfdump var="#form#" label="Form">
<cfdump var="#url#" label="URL">

</cfoutput>
</cfsavecontent>

<cfmail to="bugs@myproject.com" from="root@myproject.com"
subject="Error: #error.message#" type="html">
    #errortext#
</cfmail>
```

I create my message within a cfsavecontent (more on why in a second), and then mail it. Don't forget the type=html. Now as you can guess, this creates a pretty big email. If you get a 1000 of these, you will be suffering, but consider it incentive to fix the darn bug ASAP. You could also add a dump of the session scope if you wanted, or CGI. Basically, it is better to send more information than you need than to be wanting for more. Having all this detail gives you a better idea of what is going on when the error occurred.

So why the cfsavecontent? One trick I'll often do is to skip the email if I'm currently logged in as an admin on the site. I'll do a quick check, and if I'm an admin, I'll display the error on screen. This lets me see the error more quickly than waiting for an email.

So all together now, here is the error.cfm file:

We are so sorry. Something went wrong. We are working on it now.

```
<cflog file="myapperrorlog" text="#error.message# -
#error.diagnostics#">

<cfsavecontent variable="errortext">
<cfoutput>
An error occurred:
http://#cgi.server_name##cgi.script_name##cgi.query_string#<br />
Time: #dateFormat(now(), "short")# #timeFormat(now(), "short")#<br />

<cfdump var="#error#" label="Error">
<cfdump var="#form#" label="Form">
<cfdump var="#url#" label="URL">

</cfoutput>
</cfsavecontent>

<cfmail to="bugs@myproject.com" from="root@myproject.com"
subject="Error: #error.message#" type="html">
    #errortext#
</cfmail>
```

Now for fun, try modifying your error.cfm. Change the first cfdump tag to a cfpoop tag. If you rerun your template, you will see a blank page. Remember that we created a blank error_request.cfm file earlier? This is what is running now. Basically, ColdFusion has noticed that we had an error, and then our error management had an error, and it's thrown its hands up in the air and given up. We are now in the request template. The request template has special rules - the most important being - no CFML. That's right - you can't cflog. You can't email the error. You can - however - output error variables. You don't use cfoutput, you just include them. Consider this sample:

This went wrong: #error.diagnostics#

This will display:

This went wrong: Unknown tag: cfpoop. ColdFusion cannot determine how to process the tag cfpoop because the tag is unknown and not in

any imported tag libraries. The tag name might be misspelled. The error occurred on line -1.

That's all you can do really. But guess what - I wouldn't do that. Remember - we don't want to reveal any sensitive information to our users, including what caused an error. So what do I recommend?

Go to your site and view source. This will give you the HTML result of one of your pages. Find the content and replace it with a "We're sorry" type message like we used in error.cfm. Then save that HTML. Basically you are creating a static page. This means that if you change your layout, you have to regenerate your error_request.cfm page. Of course, you could just not use any layout at all, but most people want their pages to have a standard look and feel.

Unfortunately, nothing is logged when this error happens. So what can you do? One thing to consider is checking the web server log files to see when the file is run. If you see it running often, then double check your error.cfm file for possible errors. If worse comes to worse, temporarily wrap your error.cfm itself in a try/catch and see what shows up when you dump cfcatch. I'm open to suggestions here - but there is a reason they (ok, I) call this the "Oh Crap" error.

Ok, so I've covered quite a bit of information here, but it applies to Application.cfm and CFERROR. What if you are using Application.cfc? Well one thing to remember is that you can just as easily put CFERROR tags inside your Application.cfc file. That is allowed, and I've done that before. But what if you want to use the onError method?

In general - a lot of what I said about the Exception type for CFERROR applies here. You want to present a nice message to your user. You want to log the error. You want to email the error to yourself. But there are some subtle differences.

Consider this very simple onError:

```
<cffunction name="onError" returnType="void" output="true">
  <cfargument name="exception" required="true">
  <cfargument name="eventname" type="string" required="true">
  <cfdump var="#arguments#"><cfabort>
</cffunction>
```

All I've done here is dump all the arguments sent in. Now go back to your error file (the one you made to throw errors) and change it to this:

```
<h1>Hello World</h1>
<cfoutput>#xfhdshsdhj#</cfoutput>
```

Run it in your browser, and you will see this:


```
<cflocation url="error.cfm">
```

```
</cffunction>
```

All I did was take the code from my original error.cfm file and place it in here. The Exception argument here looks a bit different. No diagnostics key. So I just logged the message for now. My error.cfm file now only contains the message:

```
We are so sorry. Something went wrong. We are working on it now.
```

Lets recap:

- Do a quick test to figure out how your application responds to errors.
- If robust exception information is displayed, turn it off.
- Use CFERROR or onError to handle errors.
- Do your own logging, and email yourself a detailed report.

I hope you find this guide useful, and please let me know how I can improve it.