Hide Your Errors (Application Events: onError)

Posted At : March 22, 2005 2:27 PM | Posted By : Steve
Related Categories: ColdFusion, Usability

I have noticed that a great many ColdFusion sites show the default ColdFusion error when something goes wrong. This is a bad idea for many reasons.

In the "**Research-Based Web Design & Usability Guidelines**" (pdf) put out by **Usability.gov**, "Detect Error Automatically" was given an importance of 5 out of 5. In his popular "**Top 10 Web Security Tips**" article, Michael Smith listed "Have an error-handler" as his number-one security tip.

In his article "Toward Better Error Handling" (**part 1**, **part 2**, **part 3**), Charlie Arehart covers some techniques for error-handling in ColdFusion. As of the release of ColdFusion MX 7, a new method exists for handling errors in ColdFusion; the onError event of Application.cfc.

The onError event is only available if you are using Application.cfc. To get an introduction to Application.cfc, see my "**Application Events: onRequest**" blog entry or see the **LiveDocs for Application.cfc**.

To add the onError method to your Application.cfc, simply add code like the following to Application.cfc (replacing the comments with whatever code you want to run when an error occurs).

```
<cffunction name="onError">
    <cfargument name="exception" required="true">
    <cfargument name="EventName" type="String" required="true">
    <!--- Error handling code here --->
</cffunction>
```

Of course, as **Raymond Camden points out**, ColdFusion will raise an exception (and run the code in onError) when it runs in to a <cfabort>.

In order to prevent that from adversly effecting your code, he suggests adding the following lines to the top of your onError method:

```
    <cfif arguments.exception.rootCause eq "coldfusion.runtime.AbortException">
        <cfreturn/>
    </cfif>
```

So now our onError method looks like this:

```
<cffunction name="onError">
    <cfargument name="exception" required="true">
    <cfargument name="EventName" type="String" required="true">

    <cfif arguments.exception.rootCause eq "coldfusion.runtime.AbortException">
        <cfreturn/>
    </cfif>

    <!--- Error handling code here --->

</cffunction>
```

So, now you can just take this code and add your own error-handling code (for example, display a user-friendly message and send yourself an email), right? Well, almost.

While ColdFusion MX 7 is a lovely product, it has a few bugs that still need to be resolved. One bug occurs if you are using jsessions and a session expires. It results in a "Session is invalid" error being displayed to the user. **Paul Kenney provides a solution for this problem**.

My format is slightly different than his, but the result should be the same. Here is a complete example of an Application.cfc from a site that used to use Application.cfm and OnRequestEnd.cfm:

```
<cfcomponent>

<cffunction name="onRequestStart"><cfinclude template="Application.cfm"></cffunction>

<cffunction name="onRequest">
    <cfargument name="targetPage" type="string" required="true">
    <cfinclude template="#arguments.targetPage#">
</cffunction>

<cffunction name="onRequestEnd"><cfinclude template="OnRequestEnd.cfm"></cffunction>

<cffunction name="onError" returntype="void" access="public">
    <cfargument name="Apxception" type="any" required="true">
    <cfargument name="eventName" type="string" required="true">
```

```
        <cfset var Except = arguments.exception>
        <cfset var SessionExpiration = false>
        <cfset var redirectUrl= "/index.cfm">

        <cfif arguments.exception.rootCause eq "coldfusion.runtime.AbortException">
            <cfreturn/>
        </cfif>

        <cfif StructKeyExists(Except,"RootCause") AND StructKeyExists(Except["RootCause"],"Detail")>
            <cfif Except["RootCause"]["Detail"] CONTAINS "Session is invalid">
                <cfset SessionExpiration = true>
            </cfif>
        </cfif>

        <cfif SessionExpiration>
            <cfcookie name="JSESSIONID" value="" expires="NOW">
            <cfheader statuscode="302" statustext="Moved Temporarily">
            <cfheader name="location" value="#redirectUrl#">
        <cfelse>
            <!--- Code to output text to user and send email to site administrator goes here. --->
        </cfif>

</cffunction>

</cfcomponent>
```

This is the lazy route of leaving old code in Application.cfm and OnRequestEnd.cfm. I would suggest eventually moving the code around to take advantage of the structure of Application.cfc (hopefully more on that later).

Good luck!