

## Introducing third party applications into your frameworks, good practice?

Posted At : 1 October 2009 13:18 | Posted By : Shaun McCran

Related Categories: Software Architecture, Coldfusion, Recommended Apps, Best practices, File Interactions

A recent development involved some changes to a large script that was using a product called ImageMagick ( [link](#) ). It is an image manipulation tool that allows you to perform transformations and other actions on image files. The product seems stable enough, and can be executed from a web interface using cfexecute, in a kind of command line prompt method.

It got me thinking as to why the application wasn't using cfimage, and then lead me to examine the wider topic of whether to use ColdFusion's in built functionality, or opt for other third party products.

I can think of a few obvious deciding factors for both pro and con, the first that springs to mind is performance.

Is there a marked performance difference between a ColdFusion function, and the third party application? Is it better to pass the load of to the operating system, rather than have ColdFusion perform whatever processing function it is supposed to do? In this case it would be a race between cfexecute, and cfimage, so there may be very little difference in it.

The second major point that springs to mind is the knowledge base of the developer, and the structure of whatever framework you are using. What I really mean by this is 'if your chosen technology can perform a function, why not utilise it to its full potential?' . It seems a bizarre choice indeed to deliberately not harness a function that your platform can already provide, and instead introduce another code base or application into the framework.

It also introduces another hurdle for the development staff, they may well be familiar with how an existing Tag works, but be totally unaware of the third party application, as was my case here.

A long winded intro, but here is the code I used:

```
        <!--- setup vars --->
        <cfset variables.destination = "C:\dev\images\testimage.jpg">
        <cfset variables.source = "C:\dev\images\testimage.jpg">
        <cfset variables.width = "100">
        <cfset variables.height = "100">
        <cfset variables.exec = "C:\dev\apps\imageMagick\convert.exe">

        <!--- executing an external application version --->
#variables.width# #variables.source# -geometry #variables.height# -strip #variable

        <!--- CFimage version --->
:ce#" action="resize" width="#variables.width#" height="#variables.height#" destin
```