Article
Stephen Walasavage · Nov 1, 2016 2m read

Mysterious <COMMAND> errors when running common commands!

This post is meant to provide a quick possible explanation for a very perplexing problem.

Scenario: You ’ ve just created your own administrative user in your 2014.1 (or later) instance of Caché. You gave it every possible security role (including %All), so it should in theory be able to do anything within the instance.

You ’ ve written a very advanced routine with a break command in it for debugging:

```
MyTestRoutine
        set ^MyInitGlobal = 1
        write "Hello, my name is..."
        break
        write "Steve"
        quit
```

You log into Terminal with your new user, run the command, and start debugging your routine… And then something strange happens:

You get <COMMAND> errors for many very common commands! Even your attempts to figure out just what is going on return <COMMAND> errors. Eventually, you decide to quit out and try again. This fails as well!

After many attempts to diagnose the problem, you eventually close the Terminal window and vow never to write code again…

There is likely a very simple explanation here: %SecureBreak.

http://docs.intersystems.com/latest/csp/docbook/DocBook.UI.Page.cls?KEY=GCAS_secmgmt#GCAS_secmgmt_secureshell

The Secure Debug Shell was introduced in Caché 2014.1.0, and helps to better control access to sensitive data that could be affected in a regular debug shell. Basic debugging is still possible, but more advanced or possibly malicious commands are not. Many common commands, such as DO, KILL, SET, and QUIT, are restricted.

To enter the Secure Debug Shell, a user must have the %SecureBreak:Use privilege, which by default is included with the predefined %SecureBreak role. Since the %SecureBreak role is intended to restrict rather than allow certain activities, care must be taken when assigning it to a user. Selecting all the roles on a system and assigning them to your user can inadvertently give it the %SecureBreak role, which as we see above can cause <COMMAND> errors with no further explanation. One interesting thing to note here is that even though the user also has %All, the %SecureBreak role is still in effect and the Secure Debug Shell restrictions still apply.

If you ’ re encountering <COMMAND> errors at every turn while debugging, you might try inspecting the user ’ s roles (via SMP or another Terminal session) to confirm that %SecureBreak is unexpectedly one of them and removing it if needed.

#Access control #Debugging #Security #Terminal #Tips & Tricks #Caché

Source URL:https://community.intersystems.com/post/mysterious-errors-when-running-common-commands