

# Re: ActiveX Script errors suppressed

---

*Source:*

<http://www.tech-archive.net/Archive/SQL-Server/microsoft.public.sqlserver.dts/2005-04/msg00527.html>

---

- *From:* "Seb" <Seb.Thirlway@xxxxxxxxxx>
  - *Date:* 22 Apr 2005 01:47:08 -0700
- 

Hi fsanchez

I've never actually tried to call GetExecutionErrorInfo in an ActiveX script – I'm using VB. It would be great to be able to do this, but as I found when researching my problem, it looks as though this isn't possible. The quote below is from Darren Green – I wish he was wrong about this but unfortunately for your problem he does know what he's talking about!

See further below for possible workaround

>GetExecutionErrorInfo uses ByRef parameters to pass back the information  
>such as error description etc. These are type variables, long and string  
>etc. ActiveX Script only has the variant type. For ByRef to work the >types must match exactly. Using CStr and similar only suppresses the >error by ensuring the data passed in is type correctly, but does not >change the variant to a true string. You will need to use a strongly >type language. You could write a DLL in VB that could be called from >ActiveX Script or even write a custom task to do the entire error capture >job. Using a custom task gives reuse too, without copy and paste of code >between packages.  
>  
>--  
>Darren Green (SQL Server MVP)  
>DTS – <http://www.sqldts.com>

There may be a workaround to your problem, IF you can replace your ActiveX script with a Run Package task. I've never tried to do this, but you may well be able to set the child package's parameters (which you do through the Package/Package2 object in your script) with a Dynamic Properties task instead in the main package. If you can use a Run Package task, then you could wrap the execution of the main package in a VB wrapper (using pretty much the same code as you use above), and add some extra-clever functionality to the

## Re: ActiveX Script errors suppressed

error-detection part of your code:

```
[can't remember the exact object model, so this is very pseudo]
For each Step in MainPackage
If Step.ExecutionStatus = [completed] and
Step.ExecutionResult=DTSSStepExecResult_Failure Then
'[here call GetExecutionErrorInfo, as in your code]
[this line VERY pseudo – but I think you'll get what I mean]
If Step.Task.Customtask.TypeOfTask = [Execute Package task]
Then
Set ChildPackage = Step.Task.CustomTask.Package
For Each ChildStep in ChildPackage
[call GetExecutionErrorInfo]
Next
End if
End If
Next
```

you'd have to handle how the routine accumulates errors: I just set up a strErrMsgs string, then each time I find an error, rather than exiting, append the error info to the string, preceded by a CRLF if there's anything already in the string. In this way the final string contains every error detected (though of course responding to/trapping an individual error becomes difficult or impossible – but then we are dealing with DTS, the error-handler's black nightmare...).

To be really clever this procedure could be re-designed into a recursive procedure, in case someone ever sets up a package which runs a child package which runs a grandchild package which runs a great-grandchild package....

I'm going to try out this approach, I'll post results as I go. If anyone reads this and knows it can't possible work for some reason, please do post knocking my idea down!

regards

Seb

---

### • *References:*

◆ [\*ActiveX Script errors suppressed\*](#)

◇ *From:* Seb

- Prev by Date: [\*Re: DTS package disappearing\*](#)
- Next by Date: [\*Export to DBASE III – Could not update; currently locked\*](#)
- Previous by thread: [\*ActiveX Script errors suppressed\*](#)

Re: ActiveX Script errors suppressed

- Next by thread: ***dtsrun Parameter Problems!***
- Index(es):
  - ◆ ***Date***
  - ◆ ***Thread***