

## Re: DTS Environment Control

**Source:** <http://www.tech-archive.net/Archive/SQL-Server/microsoft.public.sqlserver.dts/2004-07/0784.html>

---

**From:** Allan Mitchell (*allan\_at\_no-spam.sqldts.com*)

**Date:** 07/22/04

Date: Thu, 22 Jul 2004 16:40:40 +0100

You are right it can be hairy.

I have the same environments as you so I do this.

I know I have machines which will execute packages in a given environment. I am about the only person that will execute packages in all 4. I like to make aliases to my SQL servers using CliConfig. This way I do not need to ever change them again after I have built my packages. If i need other information like The XL file for production location I can store that in a table on the aliased server and retrieve it through a DP task.

This means I need to make sure that the boxes that will call my packages in each environment have aliases set up. The users will not have permissions onto environments which they shouldn't.

In your 200 table wizard generated package you could simply use the object model to change the Data Source properties to an Aliased Server then save the package back out.

HTH

--  
--

Allan Mitchell MCSE,MCDBA, (Microsoft SQL Server MVP)  
www.SQLDTS.com - The site for all your DTS needs.  
www.konesans.com - Consultancy from the people who know  
"Joe Horton" <horj235 at lni dot wa dot gov> wrote in message  
news:uqvJkDAcEHA.4092@TK2MSFTNGP11.phx.gbl...  
DTS Environment Control  
What solutions have other worked up to handle DTS packages and promotions through their different environments?  
In our scenario we have a 4 tier environment (Dev/Test/PreProd/Prod). The challenge is when you promote your packages up; the connections have to be changed to reflect the new environment. For simple packages it's not too big of deal - but for more complete ones - you can spend an hour carefully changing all your connections. Try promoting a package that imports/exports 200 tables that was created with the wizard - there is a separate connection for every table.  
I had one solution where we would add an ActiveX script task as the very first step, that would determine what SERVER the package was executing from - and based on that would dynamically change all the connections by looking up the package name and server from a DB. Of course that was hairy

## microsoft.public.sqlserver.dts: Re: DTS Environment Control

to use because when you develop your DTS package and run it, it is running in the context of YOUR machine, despite being saved on a different server. The environment control solution I developed was really only useful when the package was ran as a scheduled job - as then it was truly running as a job on one of the 4 tier environments. The DBA's shot my solution down as it required that I keep the batch login user names and passwords in my control table that was used to change the connections based on the environment the package was running from.

The other problem is source control - we use VSS. If I have a package called XYZ that grabs files from an FTP location every morning - my XYZ package for DEV will look different than the XYZ package for TEST. I could pre-pend the environment in front of the package name and have DEV-XYZ, TEST-XYZ, PREPROD-XYZ, PROD-XYZ - but then imagine that we decided to add one more step to message the data further or pick up more files within the same package - I would have to make the changes to 4 different version of the package called XYZ - or make it to one and re-save it across the environments and change the connections for each environment. Anyone have better solutions?