Hello and welcome to “How to Create a Centralized Database environment”

Thank you for joining me today. My name is Michellea David. Please hold all questions to the end of the presentation. The slides have been uploaded to the New Stars of Data Team and will be made available by that team. If you would like a copy, please email me and I will forward the slides and my notes.

A little about me. I have 20 year’s experience in databases (Oracle and SQL Server); I am dating myself. I started in Oracle 9i and SQL Server 2000. I have programmed in Java at Sun Microsystems, which is now Oracle. I was a System Admin in a prior life and I have a Solaris Admin Certification.

I have worked for companies like Sun Microsystems and Los Alamos National Laboratory. Currently I am Covid unemployed and looking for a job. If you know of any Sr./Lead/or Management DBA positions, please contact me at my email address.

[Slide Change]

Today we will be discussing how to create a centralized database environment. Our itinerary is “What is a Centralized SSIS Environment”, “Why would you want to create a Centralized SSIS environment”, “Pros and Cons of a Centralized SSIS Environment”, the steps on “How to design a Centralized SSIS environment”, “Permission issues” you may run into between the SSIS catalog server and the database server, “Which server (Database, Catalog, or SSIS executable) is utilizing the CPU and memory or where to add memory if needed”, “Where will the job execution information be stored”, and then I will take questions.

Let’s get started!

[Slide Change]

What is a Centralized SSIS Environment? A Centralized Environment is all your jobs, code, and statistics in one centralized location. The jobs will reach out to another server, select data, then come back through the network to the SSIS server to finish the process. Procedures can be created to store job step timing, information, or errors into one centralized table on the same SSIS catalog server. Everything will be in one place for the DBA to review in the morning, instead of getting up and trying to located on each server the jobs that failed.

[Slide Change]

Reasons to use a Centralized SSIS Environment: Only one server to review daily, all jobs run from one server. There is one centralized code repository to maintain. All the code, statistics, errors, history for every server is stored on one SQL Server.

Are you tired of having the same code on many different servers? In a Centralized SSIS Environment you will be able to run Dev, QA, and Prod environments from the same code with different environment variables.

[Slide change]

There are many Advantages to using a Centralized Environment, I am only listing a few here. Only one central management location; The packages are easy to scale by adding memory to the central server. The ETL workload will be on the central server, so we keep the workload away from the OLTP, OLAP databases. and Lastly Standardize and control multiple SQL instances (Dev, QA, and Prod) from one SSIS package.

Some Disadvantages are: The Central server must be licensed (so one more license cost) and Remote data is copied over the network into the Central Server dataflow buffer which (depending on your network) can slow down the SSIS process.

[Slide change]

How do we create a Centralized SSIS?

1. Pick a central server to create the SSIS Catalog and set up the SSIS catalog
2. Create the SSIS package using Project variables
3. Deploy the SSIS Package to the Central SSIS Catalog
4. Set up the Environment variables for each Environment you want to run the code on
5. Link the Environment to the package
6. Create a job to run the SSIS package
7. In the job, include the environment you would like the code to run on

[Slide Change]

Step one: Create the SSIS Catalog

In SQL Server, right click on the Integration Services Catalogs. Select Create Catalog

Make sure you check box “Enable CLR Integration” and “Enable automatic execution of Integration Services stored procedure at SQL Server startup.” The create catalog screen will also ask for a password for encrypted content.

[Slide change]

Step 2: Create an SSIS Package with project variables

In this discussion, I assume the audience know how to code SSIS in Visual Studio. If you are new to SSIS coding or need a refresher, there are lots of courses online.

Create an SSIS package in Visual Studio. Make sure you add a project parameter for the Server and database (if using a single database). Link the connection variable in the Control Flow task to the project parameter.

[Slide change]

This slide is a picture of the connection using Expressions.

In our demo, I will show you where to locate the project parameters and connection link.

[Slide Change]

Demo time! I am using Visual Studio 2019, so you may notice differences from earlier versions.

[DEMO]

1. Switch to Visual Studio
2. Show the package.dtsx “Notice this SSIS package is just a rebuild index of all user databases on a SQL Server.”
3. Show the Project parameters “This is where you set up the parameters that will be used in your connection” “I only have the server name, but we could also include the database name if we need it. These parameters will be changed when we set up the environment variables in the SSIS Catalog” Notice the variable value is “LAPTOP-2IV6S22G\TWO”
4. Go back to package and show the connection expression variable. “When we go to the connection string the project variable will be pulled into the connection. “
5. Go into Expressions …
6. Show the evaluate expression
7. Show the servername and connection string

[Back to Power Point Slides]

[Slide Change]

Step 3: Deploy the SSIS package in to the SSIS Catalog that was created in step 1. In this screen shot, I show that you right click on the package name and select deploy. Then follow the directions.

[Slide Change]

Step 4: Set up the environment variables in the SSIS Catalog

In the Catalog under the package, there are two folders: Projects and Environment. Right click on environment. Select the create environment and name the new environment. I named my environment ONE, because I will be running the job on server one with this environment.

I created a second environment variable called TWO, which will run on the second server.

[Slide Change]

After the environment is named, we need to find out what variable to put in to the environment variables. I open the environment and go to the variables page. Keep this window open.

[Slide Change]

Then I go back to the main screen and right click on the project and select configure. Select the Parameters page. Look at the Parameter name in the table on the Parameters tab. The name we need to include on the environment variable page is Server. Keep this window open.

[Change Slide back to last slide]

Go back to the environment variables page and type the variable name “Server” then type in the value for the Server. Do this for each of your environments and click OK.

[Slide Change]

Go back to the Project variable and link the parameter to the environment variable, by clicking on the three dots next to the value.

[Slide Change]

In this window we will link the variable to the environment variable. In the Set Parameter Value window, select the Use Environment variable radio button and select the Server variable from the dropdown. Then click OK.

You will see the Server value change from “LAPTOP-2IV6S22G\TWO” to SERVER with an underline. The underline tells you that this value is linked.

[Slide Change]

In the Configure Project, select the References page. Add the environments that we have created. In this demonstration I added environments One and Two.

If the environments are not added, you will not be able to use the environment in the job.

Now click OK to get out of the configuration.

[Slide Change]

Create your job. In the job create a step to run the SSIS package. After adding the package link and still in the job step properties window, click on the Configuration tab in the middle of the page. Select the checkbox called Environment at the bottom and select the environment in which you want to run this job. Click OK.

[Slide Change]

[DEMO]

1. Go to SSMS
2. Right click on the Integrated Services Catalog – show where create catalog is greyed out
3. Right click on the project – then configuration – show the parameters and references
4. Show each environment
5. Show the job and where the environment is set in the job
6. Run the job for environment Two
7. View the history: goto project right click, select reports, all executions
8. Goto view overview and show the environment selected and the variable for TWO
9. Reset the environment to ONE
10. Run the job for environment one
11. Goto view overview and show the environment selected and the variable for One

[END DEMO]

{back to powerpoint}

[Slide Change]

Permissions: Make sure the Service Account that is running SQL Agent on the central server has privileges to run the process on the target server. What I like to do is run all the SQL Server Agents under the same Service Account, if security allows.

[Slide Change]

Which server resources are used? It depends - SSIS is an executable that will run on the server it originated from. It is not run in SQL Server; it uses its own memory and CPU.

Select statement - So if Job A is created on Server A but goes to Server B to select data. Then the start of the job is on Server A, the SSIS executable is on Server A, the process is on Server A, however a link will be set up to Server B to run the select statement and bring the data back to Server A for processing.

Maintenance -In our case, we did a rebuild of indexes. The SSIS job is on server one but we want to rebuild on Server two. The SSIS job starts on Server one, SSIS executable is on Server one, the rebuild is on Server two, and the finish job is on server one.

[Slide Change]

As we saw in the demo, the job execution information is stored on the central server.

[Slide Change]

Next steps to take: If you want to take your centralized server to the next level, you can create procedures that will record all the statistics into one table. Use the package OnError, OnInfomation, OnPostExecute, OnPreExecute, and OnWarning to call the procedure and record the data in a table.

[Slide Change]

This slide contains references that will guide you further on how to create and use an SSIS Catalog and Environment variables. The slides are being uploaded into github.

[Slide Change]

Thank you all for your time. If you don’t have any questions, feel free to take a break and wait for the next presentation. Or if you want stay on and listen to question and answers.

Any Questions?

Again, Thank you all for your time. If you have any questions that were not answered, please contact me at SQLMichelleaDavid@gmail.com.