International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 03 Issue: 11 | Nov -2016 www.irjet.net p-ISSN: 2395-0072

ETL Tool for taking backups from a Database table: A Solution

Priyanshu Gupta, ETL Software Developer, UnitedHealth Group
Kislay K Lal, Project Manager, UnitedHealth Group

Abstract - *In this paper, the author has focused on explaining the usage of an ETL tool i.e. Datastage in order to perform a general project activity. This paper attempts to describe an approach designed to take automatic backups from a table and deleting old backups on daily basis. So, this study will be substantially fruitful for any project having the similar kind of requirement.*

Key Words: Data Backup, ETL, Datastage

---

1. INTRODUCTION

An ETL tool [1] is a software application which is generally used to extract data from desperate sources, transform it and load the data from various sources to the designed target. The data sources might include sequential files, indexed files, relational databases, external data sources, archives, enterprise applications, etc. Tool also facilitates business analysis by providing quality data to help in gaining business intelligence.

Likewise, Datastage is an ETL tool [2] that is part of the IBM Platform Solutions Suite and IBM Infosphere. It uses a graphical notation to develop data integration solutions. Additionally, it is available in various versions supporting additional functionality one over the other.

2. BACKGROUND OF THE PROJECT

There was a requirement in a project to take the backup of a critical table on a daily basis along with the dates included in the file naming convention and to delete old backups in order to save space. Though this can be done through UNIX shell scripting but instead of that an easier, graphical approach was agreed wherein the backup will be taken through the ETL Tool- Datastage.

3. DEVELOPMENT METHODOLOGY

The paper explains the process to take table backups in a dataset with current date in naming file convention, and to delete old backups as required in an automatic fashion.

The requirement to delete the old backups was just due to the space issue.

The solution proposed and implemented was divided into 2 phases:

A. Taking backup in a dataset with a pre-decided naming convention
B. Deleting old datasets by running a script saved at UNIX end and executing it through Datastage

The requirement lifecycle had 3 distinct stages i.e. 1> Design 2> Development 3> Testing.

3.1 PHASE 1:

Taking backup in a dataset with its name being <PROJ>_backup_<current_date>

We are using Datastage's 'After job subroutine' to add date as part of the filename of dataset.

We run the following UNIX command that would rename the file:

orchadmin copy [path]/backup.ds [path]/backup _$(date +%Y-%m-%d).ds ;

orchadmin delete [path]/backup.ds

We are copying the file created in job by adding the date in the name of the new copied file through function: $(date +%Y-%m-%d)

Also, we are deleting the old/ original file as we don't want to unnecessarily waste the disk space.

3.2 PHASE 2:

Deleting the old datasets by running a UNIX script

We can delete a dataset in 2 ways:

A. Through Dataset management utility tool
B. Through ORCHADMIN command in UNIX

Since, dataset management utility tool can be invoked automatically through DS Director tool only, here, we are using ORCHADMIN command to delete the old datasets.

And to delete the datasets through ORCHADMIN command, few environment variables [3] need to be set.

Here, we are actually sorting the list of datasets on the basis of date created and then using the 'tail' command in UNIX to keep the required number of files and delete the rest.

The script can be stored in any suitable UNIX directory and we are using 'ExecuteCommand' stage in Datastage to execute the above discussed script.

The structure of script is:

{Set Envionment Variables}

backup_file=`ls -t [path]/backup_file*.ds | tail -n +x`

/appl/infosphere/InformationServer/Server/PXEngine/bin/ orchadmin delete echo $backup_file

4. SOLUTION IMPLEMENTATION

4.1 Taking backup in a dataset:

In our project to get current backup of the table we have created a simple datastage job that writes the data in a dataset:

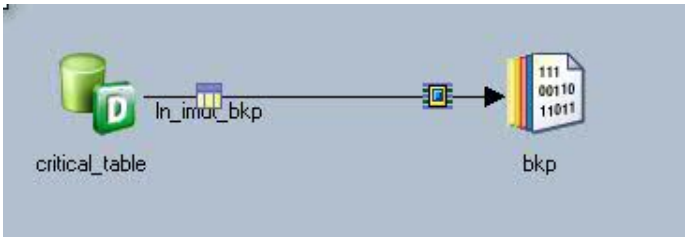

Fig -1: Datastage Job

The properties of the Dataset Stage are as follows:

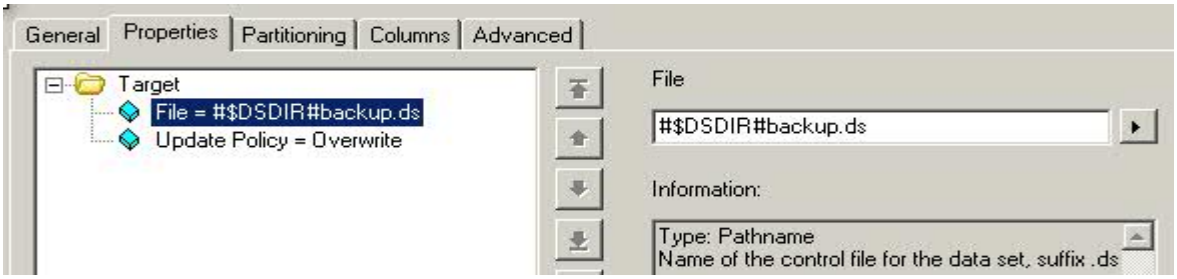

Fig -2: Dataset Stage

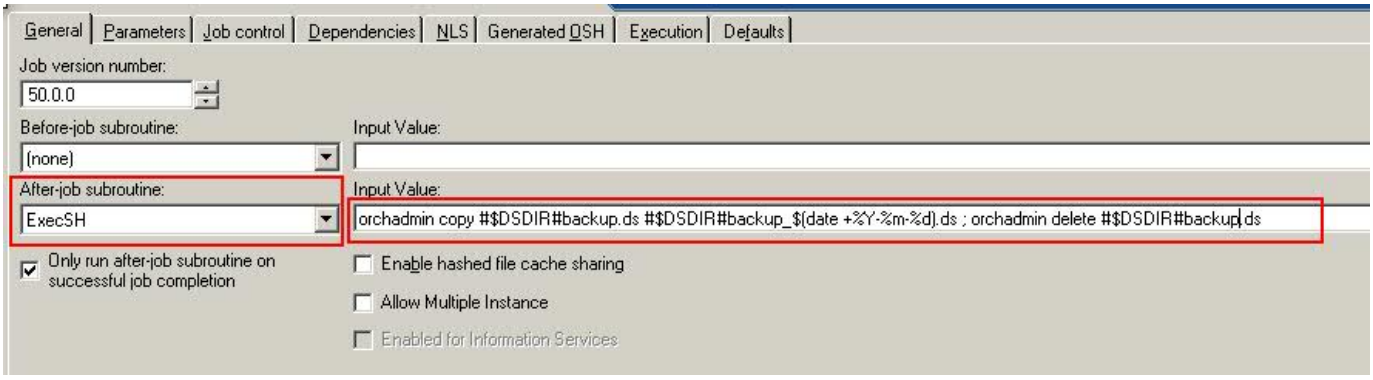

Fig -3: Job Properties

This would copy the backup file with name as (for example): backup_08-05-2015.ds and delete the backup.ds file. Thus we would be left with single Datastage file with current date included in the name as required.

4.2 Deleting old datasets by running a script saved at UNIX end executing it through DataStage:

We have created and saved a script in UNIX. The script used is:

cd /appl/infosphere/InformationServer/Server/DSEngine

. ./dsenv

LD_LIBRARY_PATH=$APT_ORCHHOME/lib:$LD_LIBRARY_PATH; export LD_LIBRARY_PATH

APT_CONFIG_FILE=/appl/infosphere/InformationServer/Server/Configurations/default.apt;

export APT_CONFIG_FILE

APT_ORCHHOME=/appl/infosphere/InformationServer/Server/PXEngine; export APT_ORCHHOME

PATH=$APT_ORCHHOME/bin:$PATH; export PATH

backup=`ls -t /data/infosphere/Datasets/change_imdt*.ds | tail -n +4`

/appl/infosphere/InformationServer/Server/PXEngine/bin/orchadmin delete echo $backup

Here in the command:

backup=`ls -t /data/infosphere/Datasets/change_imdt*.ds | tail -n +4`

tail –n +4 means that we are keeping last 3 day files. If required we can change the number here as per our requirement.

Now at DataStage end we have used a Execute Command Stage in Sequential job:

Fig -4: Execute Command Stage

With the parameters as:

Fig -5: Execute Command Script

The sh –x command as explained earlier gives the details in DataStage logs and let us know which files were deleted.

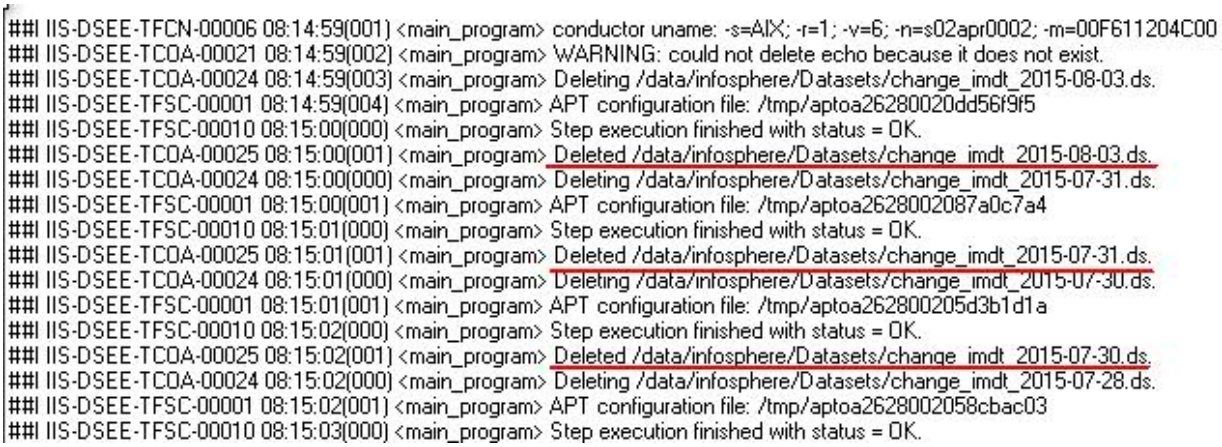

Fig -6: Datastage Logs

The simplified sequencer to run the whole process could be like:

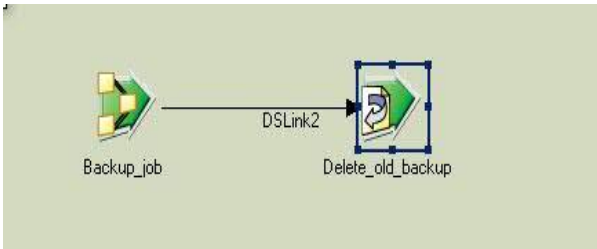

Fig -7: Datastage Sequence Job

5. RESULT AND SCOPE

It is often required in a project to take daily backup of few tables maybe for archival keeping, for debugging, for analysis etc.

The sequential file/ dataset could both be used for this purpose but as there is a limit to the size of sequential file so it is preferred to take back-up in Dataset for ever increasing data. The method provides a simplified job to save a dataset with the date so as to provide information regarding the date on which it was created as it adds current date and also overwriting of same dataset is avoided as, each day the name of dataset would be different.

While taking back-up we often overlook the importance of deleting old backups as at one point of time the data is of no use for us and is still lying and eating up space.

When we face disk full situation, at that time me have to sometimes manually delete backups taken long back, so it's better to automate this as well as we know which old data would be of no use to us.

The solution implemented addresses this method of automatically deleting the backup files.

REFERENCES:

[1] Inmon W. Building the Data Warehouse. – New York: John Willey & Sons, 1992.

[2] 1. E. Malinowski, E. Zimanyi, "A Conceptual Model For Temporal Data Warehouses And Its Transformation To The ER And The Object-Relational Models".

[3] Kar Sitikantha, "Large scale data migration using ETL tool", submitted for Software Engineering 2010 Conference, Innsbruck, Austria