**IGC Metadata Backup and Migration Utility**

**ABSTRACT:**

This Backup and Migration Utility Tool is written in Customized Python script to help content administrators in Metadata Backup process at single instance from multiple environments without connecting them independently and helps to reduce 50% manual effort.

This is very useful utility for migrating the content across different environments (Dev, QE and Production) with quick backup storage for all Asset updates. This Utility helps to revert the code to original changes if required in all the environments. This showed significant change in day to day deployments and major releases as everything is automated.

**DESCRIPTION**

**IGC Metadata Backup and Migration tool is a** Customized utility that focuses on enhancing content Migration and Backup automation process for multiple environments in single instance.

**FUNCTIONALITY**

It provides Backup storage files for multiple Assets in different environments at same instance and parallelly migrate them to required/all environments (Dev,QE and Production) with single utility.

**INSTRUCTION**

Copy this backup folder with Run\_backup script to your local directory then executes at Anaconda prompt (Python) as below:

 Run\_backup.py <userID> <Password> <Asset File> <backup>

Then this will show us as imported or exported with backup or import process (["all","backup","qa","prd","prd\_import","qa\_import"]

**RETURNCODE**

1 - Required parameter not passed. Fix: Please pass the user/Password/file/cmd parameters while backup script execution

2 - No file found. Fix: Please check the excel file availability on your backup directory with assets to be migrated/imported.

This utility tool is developed out of the below needs:

1. To Backup the Metadata content across environments using single utility tool
2. To Migrate/Import the Metadata content in multiple environments using single utility.
3. Quick storage and Manual Versus Automated Approach over 60% reduction in effort
4. Easy to plug in anywhere.

**How to install:**

You'll need the following components installed:

Python(Anaconda ) – Install Python and Import all the pandas to run the script

IGC Application (Information Governance catalog- Business Glossary) - Access is required.

**How to use:**

1. Once Python is installed with required Pandas, Copy the python script including backup folder to current home directory.
2. Change the Server name (yellow)to required IGC server environment (QA/prod) in **Run\_backup.py** script.
3. Keep the assets to be imported in Excel file in backup folder with the given components in script as (host,db,schema and table)
4. Execute the script like "Python Run\_backup.py <userID> <Password> <Asset File> <backup>
5. This script will show output with number of Assets exported or migrated or both based on the selection given by user

==================================================================

**Python SCRIPT**  **:**

==================================================================

========================================================

**import os**

**import sys**

**import pandas as pd**

**usr = sys.argv[1]**

**passwd = sys.argv[2]**

**file = sys.argv[3]**

**cmd = "all"**

**if len(sys.argv) > 4:**

**cmd = sys.argv[4]**

**if not cmd in ["all","backup","qa","prd","prd\_import","qa\_import"]:**

**cmd = "all"**

**curr\_dir = os.path.dirname(os.path.realpath(\_\_file\_\_))**

**path = "c:\\ibm\\InformationServer11\_5\\Clients\\istools\\cli"**

**cmds = {**

**"prd\_export":"call istool export -dom NJROS1BA0237 -u %s -p %s -v -ar \"%s\\prd\\{host}\_{db}\_{schema}\_{table\_}.tbl.isx\" -cm '\"/{host}/{db}/{schema}/{table}.tbl\"'"%(usr,passwd,curr\_dir),**

**"qa\_export":"call istool export -dom NJROS1BA0668 -u %s -p %s -v -ar \"%s\\qa\\{host}\_{db}\_{schema}\_{table\_}.tbl.isx\" -cm '\"/{host}/{db}/{schema}/{table}.tbl\"'"%(usr,passwd,curr\_dir),**

**"prd\_import":"call istool import -dom NJROS1BA0237 -u %s -p %s -v -ar \"%s\\qa\\{host}\_{db}\_{schema}\_{table\_}.tbl.isx\" -cm '-allowDuplicates' -replace"%(usr,passwd,curr\_dir),**

**"qa\_import":"call istool import -dom NJROS1BA0668 -u %s -p %s -v -ar \"%s\\imp\\{host}\_{db}\_{schema}\_{table\_}.tbl.isx\" -cm '-allowDuplicates' -replace"%(usr,passwd,curr\_dir),**

**}**

**df = pd.read\_csv(file)**

**os.chdir(path)**

**# Create backup folders**

**qa\_path = os.path.join(curr\_dir,"qa")**

**prd\_path = os.path.join(curr\_dir,"prd")**

**for directory in [qa\_path,prd\_path]:**

**if not os.path.exists(directory):**

**os.makedirs(directory)**

**for index, row in df.iterrows():**

**print((index,row))**

**table\_= row["table"]**

**if row["table"]=="\*":**

**table\_ = "\_"**

**if cmd in ["all", "backup"]:**

**# QA & PROD export**

**pcmds = [**

**cmds["prd\_export"].replace("{host}",row["host"]).replace("{db}",row["db"]).replace("{schema}",row["schema"]).replace("{table}",row["table"]).replace("{table\_}",table\_),**

**cmds["qa\_export"].replace("{host}",row["host"]).replace("{db}",row["db"]).replace("{schema}",row["schema"]).replace("{table}",row["table"]).replace("{table\_}",table\_),**

**]**

**if cmd in ["qa"]:**

**# QA & PROD export**

**pcmds = [**

**cmds["qa\_export"].replace("{host}",row["host"]).replace("{db}",row["db"]).replace("{schema}",row["schema"]).replace("{table}",row["table"]).replace("{table\_}",table\_),**

**]**

**if cmd in ["prd"]:**

**# QA & PROD export**

**pcmds = [**

**cmds["prd\_export"].replace("{host}",row["host"]).replace("{db}",row["db"]).replace("{schema}",row["schema"]).replace("{table}",row["table"]).replace("{table\_}",table\_),**

**]**

**if cmd in ["qa","prd","all","backup"]:**

**for tcmd in pcmds:**

**print(tcmd)**

**os.system(tcmd)**

**# Import specific Tables into PRD**

**if cmd in ["all","prd\_import"]:**

**for index, row in df.iterrows():**

**table\_= row["table"]**

**if row["table"]=="\*":**

**table\_ = "\_"**

**# QA & PROD export**

**pcmds = [**

**cmds["prd\_import"].replace("{host}",row["host"]).replace("{db}",row["db"]).replace("{schema}",row["schema"]).replace("{table}",row["table"]).replace("{table\_}",table\_),**

**]**

**for tcmd in pcmds:**

**print(tcmd)**

**os.system(tcmd)**

**# Import specific Tables into QA**

**if cmd in ["qa\_import"]:**

**for index, row in df.iterrows():**

**table\_= row["table"]**

**if row["table"]=="\*":**

**table\_ = "\_"**

**# QA Import**

**pcmds = [**

**cmds["qa\_import"].replace("{host}",row["host"]).replace("{db}",row["db"]).replace("{schema}",row["schema"]).replace("{table}",row["table"]).replace("{table\_}",table\_),**

**]**

**for tcmd in pcmds:**

**print(tcmd)**

**os.system(tcmd)**

**os.chdir(curr\_dir)**

**====================================================**