**Title :** Tips on Automation for Data Migration using BODS

**Author :** Swetha Narayana

**Document Date :** 05Mar2019

**Applicable for :** BODS data migration

**Overview :** This document covers the various options and details for automating data migration loads using SAP BODS and covers examples from an Insurance project perspective.

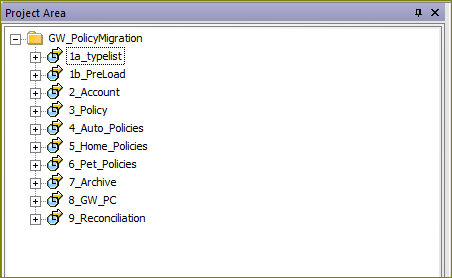
**Pre-requisite :** basic concept of various objects in BODS



**Business Case :** Joe is a data migration developer working on an Insurance project. To be specific - the insurance product being Guidewire – Policy Centre [GW-PC] and legacy data needed to be migrated to Guidewire. Say, the project is in the final phase of implementation and about to enter the UAT phase and Joe wants to automate the various data load tasks in this project, without the use of any 3rd party monitoring tool, due to budget constraints in the project. Joe needs to implement a customized solution based on his requirements, due to the non-availability of a build-in scheduling/monitoring tool.



Here is the list of sample jobs that needs to be automated and be sequenced in below order:

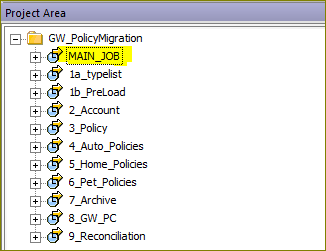


Some basic details of ETL jobs, with the considerations for a data migration from a Guidewire Insurance project perspective.

|  |  |
| --- | --- |
| **Job** | **overview** |
| 1a\_typelist | To load the base data needed for type list conversion that is needed in all the further jobs |
| 1b\_PreLoad | To load some pre load data needed for all the further jobs |
| 2\_Account | Loads stage data for account entities in PC – account, address, contact, account contact, account contact role etc |
| 3\_Policy | Loads stage data for policy entities in PC – policy, policy period, Policy Line, Policy Location, Policy Contact Role etc |
| 4\_Auto\_Policies | Loads stage data for line of business - auto policy in PC –  Vehicle, driver, auto coverages etc |
| 5\_Home\_Policies | Loads stage data for line of business - home policy in PC – Dwelling, ho-location, home coverages etc |
| 6\_Pet\_Policies | Loads stage data for line of business - pet policy in PC |
| 7\_Archive | To archive only some important stage tables |
| 8\_GW\_PC | To kick off Guidewire-Policy centre load to read data from stage tables |
| 9\_Reconciliation | Post Policy Centre load, to do a reconciliation :  legacy <-> stage <-> Policy Centre |

**Automation requirement:**

Requirement is to automate the jobs to get triggered in sequence and in the event of any job having a failure should stop the further jobs from getting triggered. So can create one main job can be scheduled at a specific time daily which would take care of the scheduling in the desired sequence.



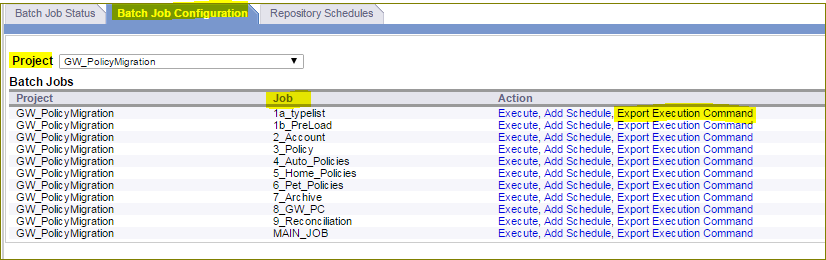
**Solution Implementation:**

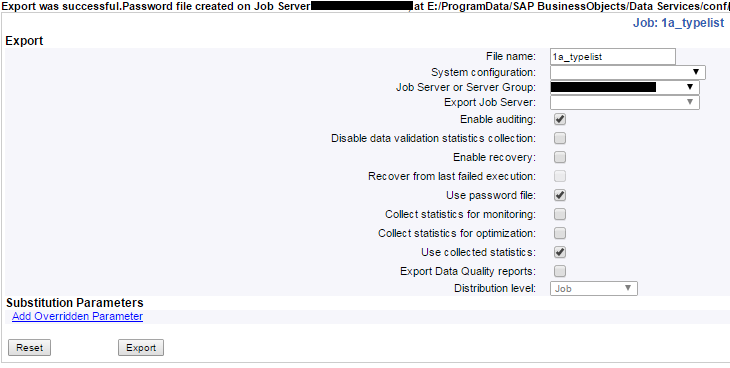
Best option to implement the requirement is by using a customized approach with a combination of batch jobs and scripts.



Step1: login to management console http://<host:portnumber>/DataServices/admin.jsp

Step2: Export the execution command for each of the batch jobs from BODS management console – “batch job configuration” tab and generate batch file as per below instructions





Step3: place the generated files on a location that is accessible by job server during execution.

Step4: configure the path & subfolder variables from step3, preferably by reading from a substitution parameter, naming convention can be as mentioned below

* $$OnRenewal\_Path
* $$OnRenewal\_MainJob\_SubFolder

Step5: login to BODS designer and create the main job in the respective project

Step6: use one of the below attached script options to configure the job.

Basically, we use below script to call the generated batch files from step2.

Sample script**:**  exec(''||$GV\_Path||'\MAIN\_JOB.bat','',8));

*Scripts: option1:* Full job script in notepad format

*procedure to implement -* Just copy the attached notepad script in a new BODS job.



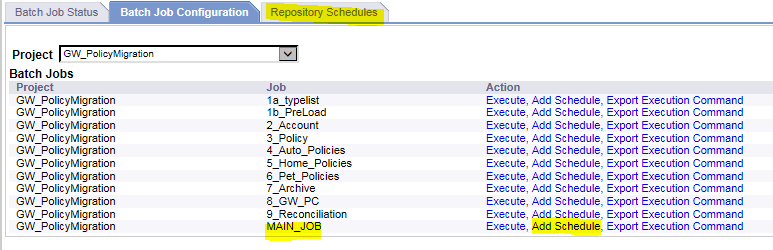
*Scripts: option2:* will be sharing a plug-in file in bods ATL format shortly

*procedure to implement -* can be directly imported in BODS designer

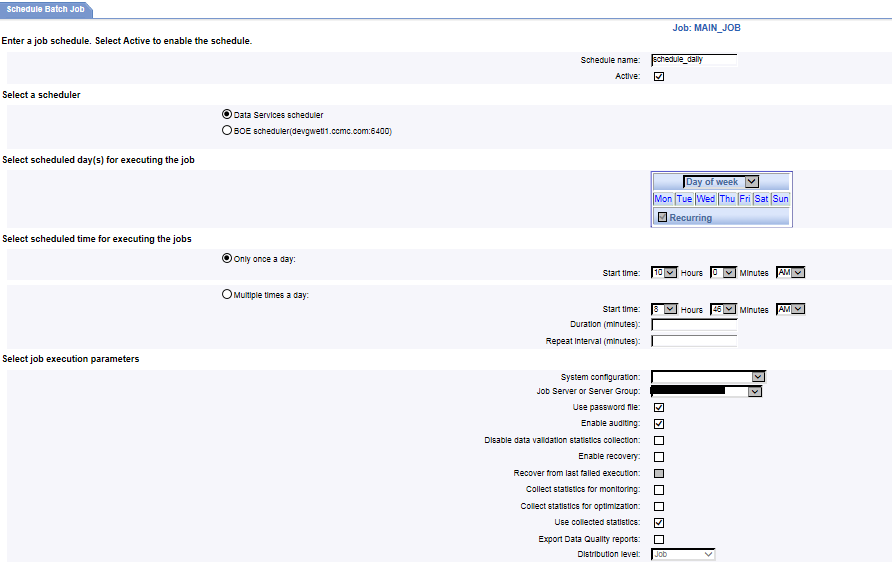
Step7: main job to be scheduled based on the requirement from management console. [say daily at 10 am PST]

Procedure to add a schedule:

Management console -> Repository Schedule -> Add Schedule option -> set settings -> apply



Set up the desired time/frequency settings

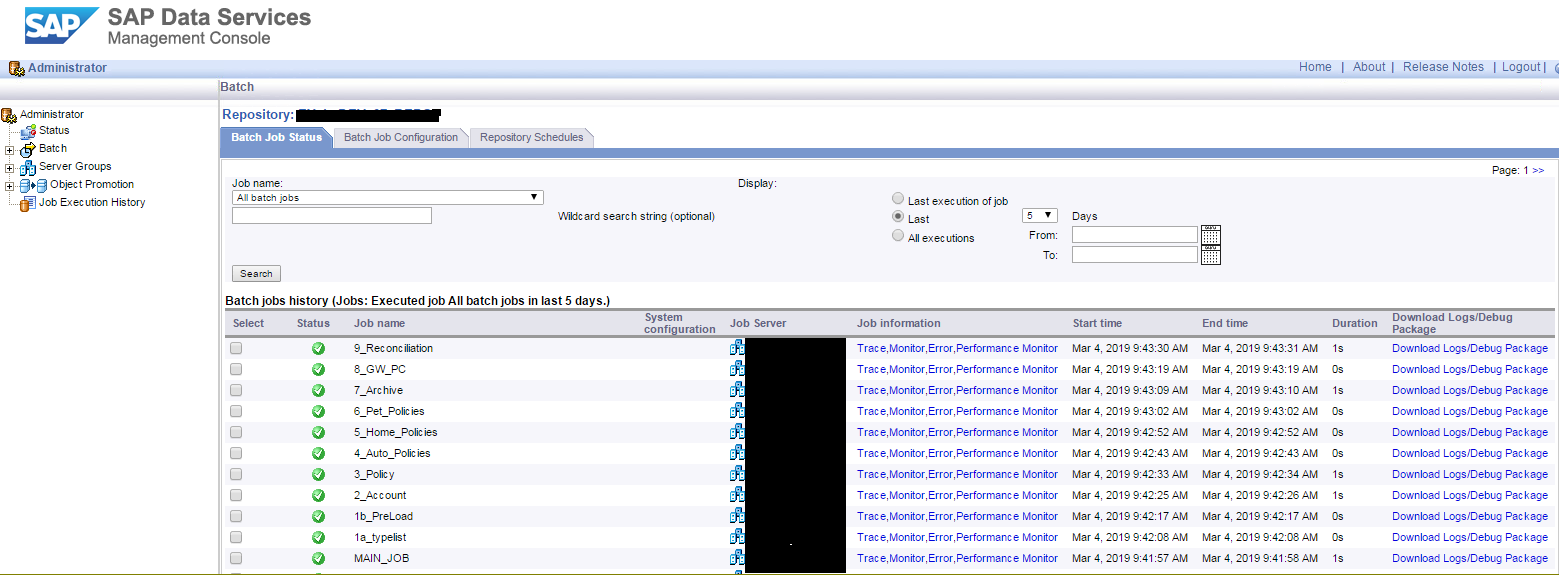


Tips: We can use custom table and use custom function to update values during job execution as per the customization requirement.

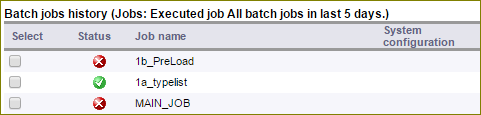
**Execution Behaviour:** Below logs [from BODS management console] depicts the expected behaviour of how the main job triggers all the dependent jobs in sequence, but only upon successful completion of each job. Upon failure of any job with an unhandled exception, any further load will not get triggered.

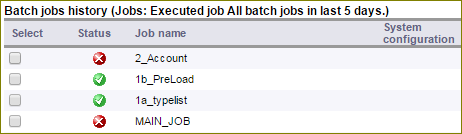
**Scenario 1:** best case, no failure in any job

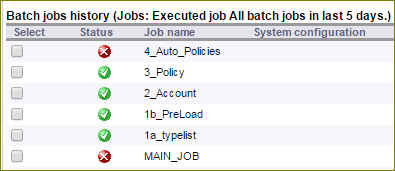
Note: the order of jobs in below snapshots follow – last in first order [observe the start-time, end-time columns]



**Scenario 2:** failure in any job will not trigger any further jobs and MAIN\_JOB depicts the failure status as well







**Value Delivered:**



* Manual efforts are no more necessary, only the main job needs to be scheduled at the desired time/frequency
* Reduced manual efforts means less of manually induced errors
* Entire ETL loads are automated: no need to manually monitor status of each job completion
* We have the obvious savings in cost and developer time: developer time saved in terms of man-hrs and also in terms of cost [since a 3rd party scheduling tool was not purchased]

**More Solution considerations which cover a wider range of requirements:**



Without the availability of a 3rd party tool [may be due to cost constraints], let us review all the major options which can be useful for a wider range of audience referring this document.

Choose the best solution based on efforts and ease of maintenance.

|  |  |  |
| --- | --- | --- |
| **SL No.** | **Approach** | **Application** |
| 1 | Batch jobs | Most widely used, for Integration with external applications, with other BODS jobs.  [already discussed in above scenario] |
| 2 | Web services | Integration with external applications based on java, C# etc, with other BODS jobs. |
| 3 | BODS metadata | For scheduling requirement based on a wide range of load statistics |
| 4 | SAP BW Process chains | Integration with other SAP products |
| 5 | Time based Scheduling | Schedule individual jobs based on approximate time take by each job by adding some extra window time, considering the status of previous job has no effect on the next job[s] |
| 6 | Scripts | Custom scripts in BODS |
| 7 | Customization | This being the most widely used, since we get a lot of options using a combination of all the above features to achieve the desired results. |

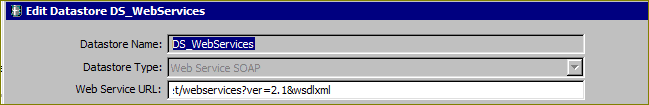
**Web Services:** Publish job[s] as webservice and this can be called from external applications or from other BODS jobs.

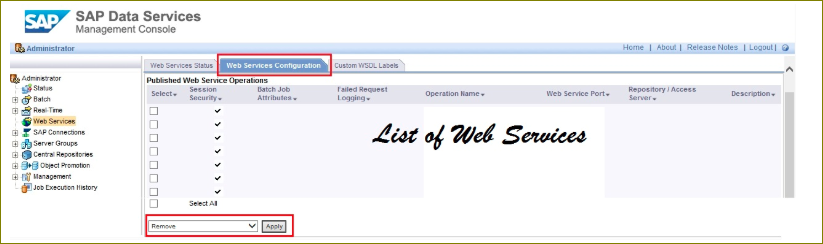


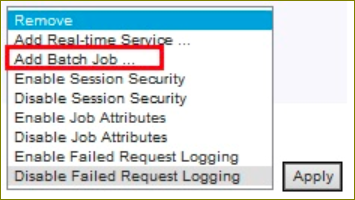
Scenario : there is an integration being done in the project landscape based on a .net or a java based application and need to integrate the bods jobs. This can be easily implemented by using the web services feature.

Solution Implementation:

Create a datastore of type web service and publish the desired job[s] as web service from management console



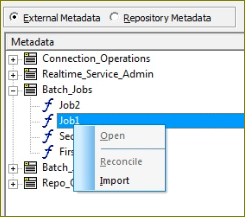




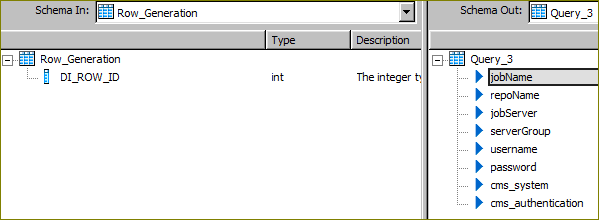
**Option1:** call this from external applications

**Option 2:** call this from other bods job[s]

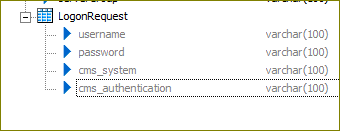
Import this as function on bods designer

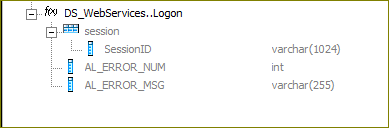


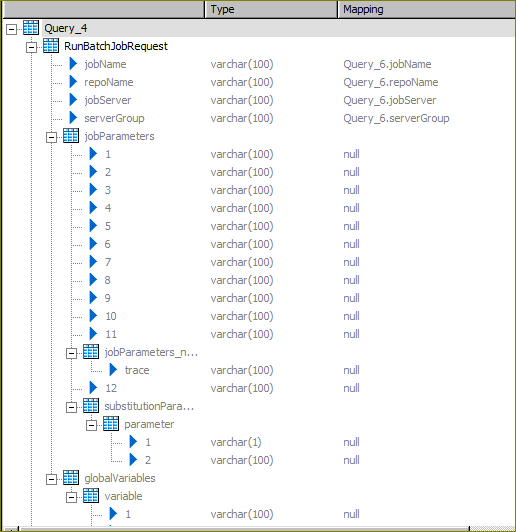
Prepare the dataflow with xml data input to call the required job from another job,

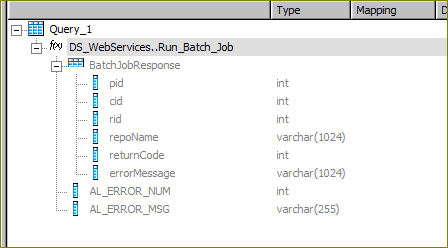


First call the login web service and get session id, and call any job as required after this step by preparing the global variables and other parameters need to trigger job.









Refer – section 3 from the SAP Integrator Guide for an exhaustive list of webservices

<https://help.sap.com/doc/PRODUCTION/ace9cae2f4cf408883eb0120b494f3dd/4.2.8/en-US/ds_42_integrate_en.pdf>

**BW Process chains:**

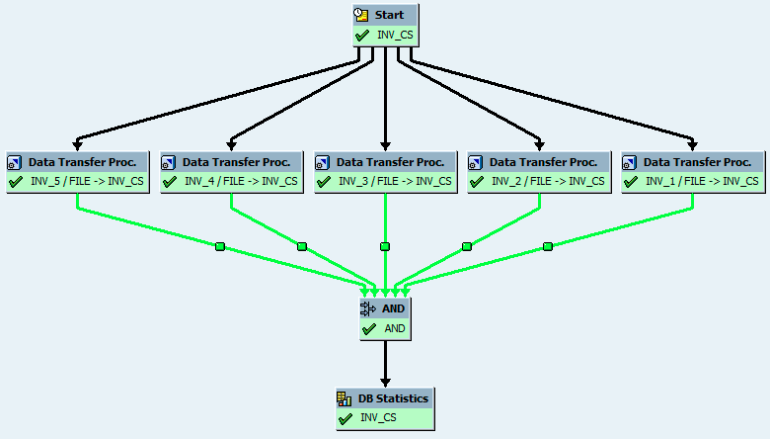
BODS jobs can be integrated with other SAP applications. Review the project landscape to find out if needed to integrate with other products and their support for one another.

BW Process chains has a support for bods: can trigger and monitor bods jobs and also can be configured to send out notifications.

Scenario:

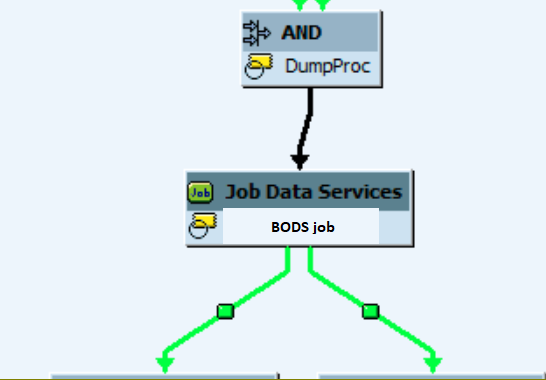
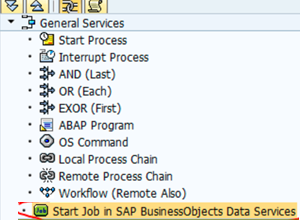
A requirement to trigger setup loads in SAP ECC and upon completion to trigger the data migration loads from bods.

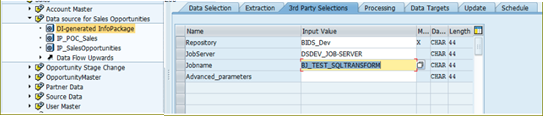
Solution Implementation:



Create a new process chain to call the process/job in ecc to delete and load the set up tables and add the subsequent steps to trigger bods jobs.

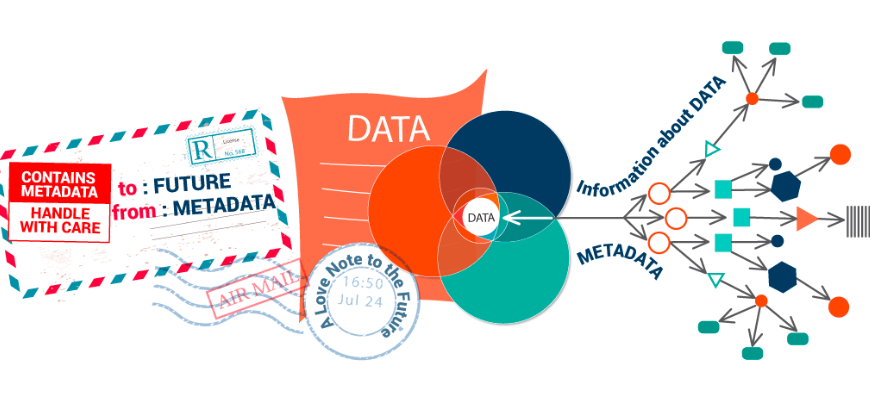
Below snapshot shows the settings to implement the same. Add the variant and we also have the option to add success/failure notification with details as well.



**BODS Meta Data:**

Can query the metadata objects like - ALVW\_HISTORY, this has a lot of applications due to the ocean of information that can be queried from metadata objects.



Scenario:

The scheduling order of the jobs to be decided based on the execution status and a set of rules is decided initially like the one shown in below table.

|  |  |
| --- | --- |
| Decision for next step in scheduling | dependent job that needs to get triggered |
| Job 1 - success | Job 2 |
| Job 1 – failure | Skip job 2 and go to job 3 |
| Job 1 – certain specified warnings | Job 2 |

Solution Implementation:

Create a datastore for the specific BODS repository by giving all the required details – call it BODS\_01REPO. Import the required views [say ALVW\_HISTORY, AL\_HISTORY, AL\_AUDIT, AL\_AUDIT\_INFO, AL\_HISTORY\_INFO etc], based on what information that is required.

Query these table/views to find the desired parameters [Example AL\_HISTORY has a “status” code column that provides the various states of a job execution], and based on the requirement take the appropriate next action using customized scripting.