



Best Practice : KITS Datastage Migration using BODS



1/29/16

KITS Datastage Migration using BODS

Domain/ Service/ Technology /Geo: **Retail & CPG**

Name of the Project/Support Fn: **KITS Datastage Migration**

Name of the Author: **SOUMYAJEET DHAR**

Date Created: **29th Jan 2016**

Description

Project/Context

- ❑ Best practices adopted in the creation of ETL Jobs using SAP Business Objects Data Services (BODS) v4.2 having similar functionality, business rules and target output present in existing ETL jobs developed in Datastage v7.5.

Purpose

- ❑ This document explains the best practices implemented during the development using SAP BODS by highlighting the benefits and its usage.

How we did it

Process Adopted

Below are the list of best practices that can implemented in an ETL project using BODS

1. Use of separate 'Central Repositories' for separate environments (Dev/QA/Prod).
2. Use of separate 'Local Repositories' implemented for individual developers.
3. Use of Multiple 'Substitution Parameter Configurations' for swapping execution environment with ease
4. Performance driven coding with the effective use of 'Display Optimized SQL' and 'Generate ABAP code'.

Highlights

- ☐ Easy debugging.
- ☐ Effective version history tracking among CR/MR changes with ease.
- ☐ High performance.

How we did it (Contd..)

Process Adopted

Below are the list of best practices that can implemented in an ETL project using BODS

5. Use of uniform naming convention of Project Folder, Job, Workflow, Dataflow and other components developed across multi-developer environment following standard document.

6. Use of “Beyond Compare” tool to compare output files created via both BODS job and existing Datastage job.

7. Use of raise exception code snippet to capture any exceptions caught during job execution.

8. Performing cross unit testing and code peer review of all developed components

Highlights

- ☐ Uniform Coding standard.
- ☐ Effective and robust Unit and System testing.

How we did it (Contd..)

Process Adopted

Below are the list of best practices that can implemented in an ETL project using BODS

9. Round the clock job monitoring during deployment to take immediate action on any job failure.

10. Taking backup of all existing Target Tables before deployment.

Highlights

- ☐ Organised and smooth deployment.
- ☐ Effective system recovery.

Why this is a Best Practice

Before

1. High vulnerability of a job to run in an undesired environment by mistake.
2. Frequent designer crashes loosing unsaved data.
3. High retrofitting effort to sync CR/MR changes among multiple environments.
4. Time taking job runs with low performance.

After(Benefits)

1. Easy version tracking for changes among CR/MR and new development by using separate Central repositories for separate environments.
2. Negligible work loss due to designer crashes by using separate local repositories for separate developers.
3. Less retrofitting effort to sync CR/MR changes among multiple environments.
4. High performance code at the end of Realization phase due to performance driven development with the help of 'Display Optimized SQL' and 'Generate ABAP code' functionality.

Why this is a Best Practice (Contd..)

Before

- 5. Non-uniform naming convention followed by multiple developer causing ambiguity in codes.
- 6. Comparison of target files (i.e. csv , xml) having huge data consume large bandwidth and this process is human error prone too.
- 7. Exceptions coming into catch block of any job are not handled properly within the job.
- 8. High defect occurrence during Unit testing

After(Benefits)

- 5. Maintaining uniform naming convention of Project Folder, Job, Workflow, Dataflow and other components developed across multi-developer environment following standard document enriches code.
- 6. Usage of “Beyond Compare” tool to compare huge files takes less time and leads to effective testing.
- 7. Use of raise exception code snippet in catch block capture any exceptions caught during job execution.
- 8. Performing cross unit testing and code peer review of all developed components ensure the quality of the development and reduce the chance of defect occurrence in SIT,UAT and after production deployment.

Why this is a Best Practice (Contd..)

Before

- 9. High response time in case of any Job failure during production deployment.
- 10. Huge effort spent on back-out of any job and retrieval of the historical data.

After(Benefits)

- 9. Round the clock job monitoring during production deployment reduces response time and facilitates to take immediate action on any job failure.
- 10. Taking backup of all existing Target Tables before production deployment ensures effective back-out plan and improve customer confidence on historical data.

How this may be adapted elsewhere

Replication

Proposed best practices can be used by the projects handling large amount of data with high performance needs while working with SAP Business Objects Data Services (BODS).

Soumyajeet Dhar

Soumyajeet.dhar@tcs.com



Thank You

IT Services
Business Solutions
Consulting