ETL Revision History and its importance

by Sudheer Sharma - Wednesday, May 05, 2010

http://dwhnotes.com/archives/1291

What do you think of this post?

Awesome (0) Interesting (0) Useful (0)

Well, I'm an ETL consultant and I mostly work at clients place for short / long term assignments. As soon as I get the assignment with the proper environment setup I have to jump into ETL activities like, writing the ETL code, changing the existing code as per new requirements or what soever, enhancements etc.,

When it comes to the development, I always strongly believe in writing an ETL code from the scratch is much easier than understanding the existing code written by XYZ. Sometimes….. no… no… many times in consulting world, you may touch others code and you might change any object job, workflow, dataflow, table, script, function, etc.,

Assume that you need to work on one of the object – job/workflow/dataflow, and it was developed by XYZ, who already left the organization for some reasons. Now the actual problem comes here, there was no proper documentation to describe the flow of the job, or the appropriate document was misplaced or no one knows the exact location of the document or in other words the document was not updated after the initial draft or something like that, finally for whatever reasons there was no document which describes the actual flow of the job. And no other person in the organization can guide you to make you to understand the flow of the job.

There might be many questions raise in your mind

- How many times that dataflow got revised?
- Who modified it? and when was the last modified date ?
- What is the latest version of the code and where it is available?
- What are the base documents (ETL Design docs and mapping docs) to build the dataflow and where can you find these documents (URLs to Teamsite/ sharepoint location)?
- What is the latest Build number? what were the defects handled in the last build?

You might get answers for these questions from Defect tracking tools, Code version controlling tools etc., but can you get answers as soon as you received the code??? No way… you have to do all your analysis (that would cost good amount of time) then you may need to gather information from different people in the team, who can help you out.

If you have a REVISION HISTORY inside your ETL code, probably it may answer you some of these

questions immediately in one go such as

- Who created the code and when it was created.
- Who modified(if at all) and when it was modified by.
- A brief description of previous change in the dataflow.
- What are the defects fixed in the current version of the code.

You can use a script step in BODI before the parent workflow. However, different organizations use different templates, some people use BODI annotations inside/outside the dataflow and some uses scripts before the dataflow/workflow etc.

Here is the revision history/change management template that I would like to share. If you have similar thing in your job then you could easily track out things without any pain and in no time.

```
######################################################################################
###########
#PROGRAM NAME   : JOB_CUSTOMER_DIM
#PROGRAM DESCRIPTION   : Job to load Customer dimension
#CREATED ON   : 04/04/2006
#CREATED BY   : Sudheer Sharma Goda
#VERSION NUMBER   : 1.0
#CHANGES MADE   : New Code
#Design Doc Reference   : URL of the share point location
#Mapping Doc Reference   : URL of the share point location
######################################################################################
###########

######################################################################################
###########
##################################### CHANGE MANAGEMENT
##########################################
######################################################################################
###########

######################################################################################
###########
#CHANGE DATE   : 05/05/2006
#CHANGED BY   : Sudheer Sharma Goda
#VERSION NUMBER   : 1.1
#CHANGES MADE   : 1- Added new query transformation(qry_getkeys) to get the keys
#   : 2- Replaced source customer table with customer_type
#   : 3- Applied lookup function in 2nd qry transform
#   : 4-
#   : 5-
#
######################################################################################
```

\###########

What do you think of this post?

Awesome (0) Interesting (0) Useful (0)

_____

PDF generated by Kalin's PDF Creation Station