Audit Control & TWS Excel App

for EDW

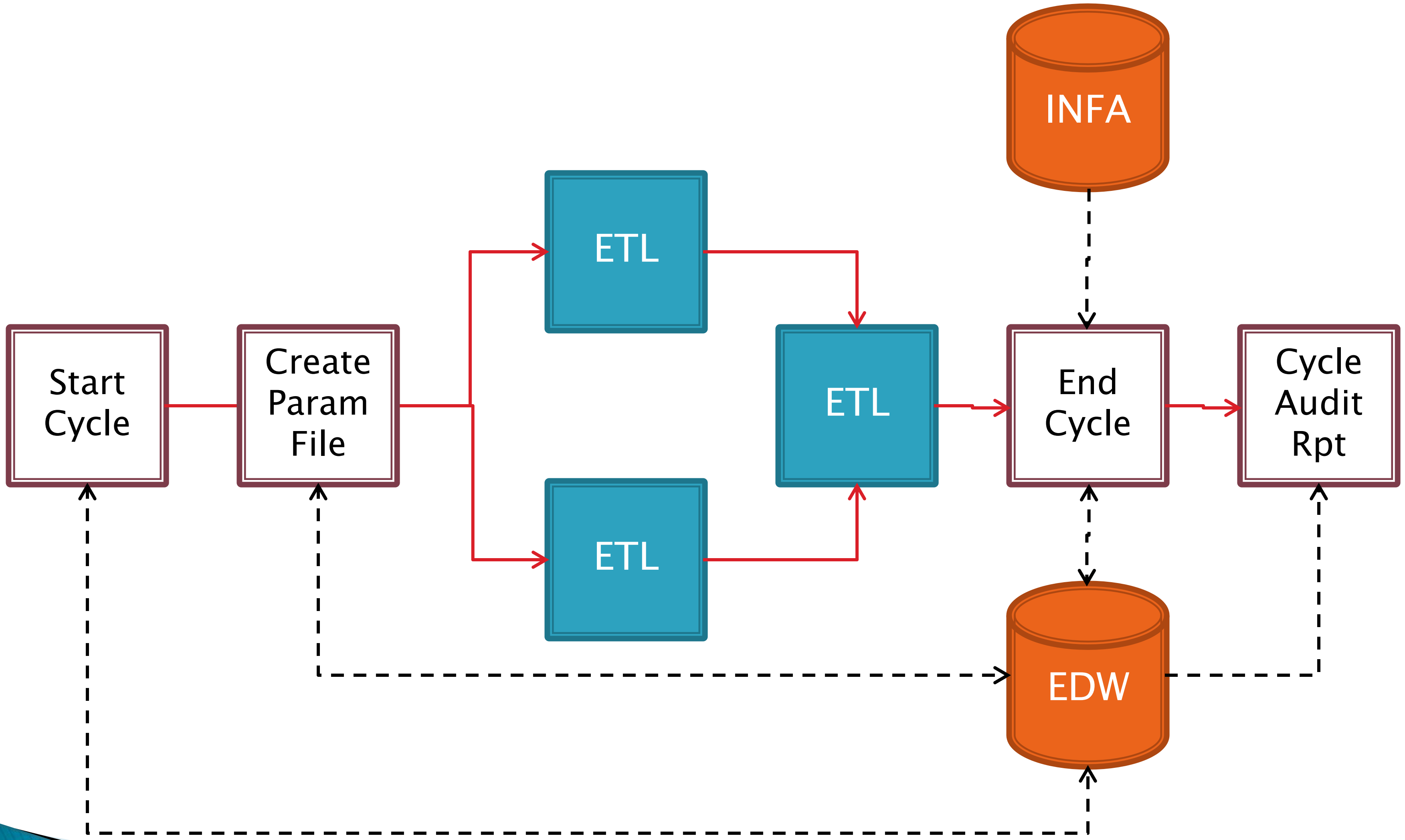
Sajjan Janardhanan
01/24/2013

Agenda

- Maintaining high level audit info
- Creating dynamic parameters for ETL
- Gathering ETL load information
- ETL load/audit report
- TWS Excel App

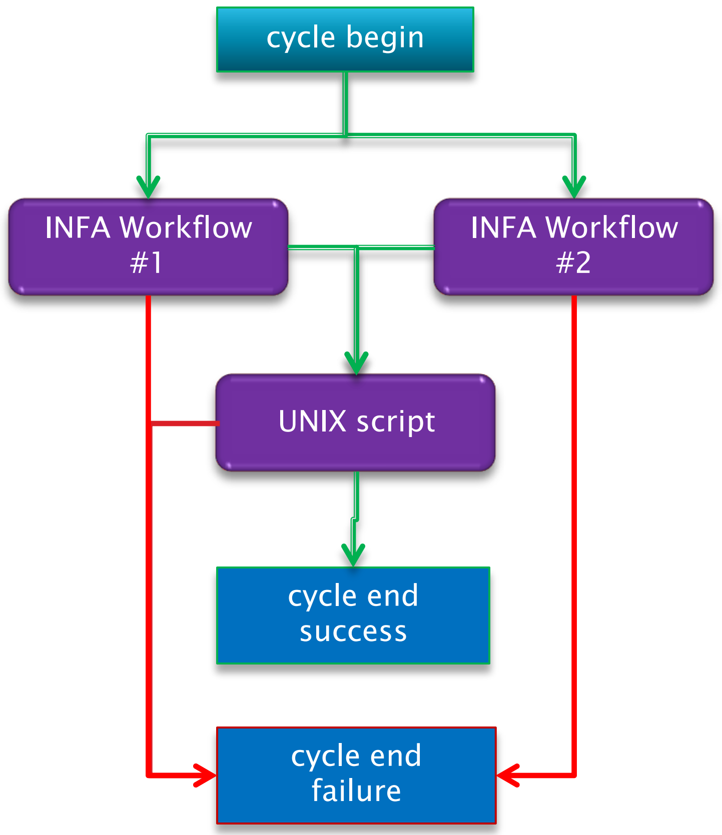
- Generic UNIX scripts
- Parameter driven
- Implemented for
  - LIS
  - MIDAS Core Measures
- Easy maintenance
- Objects in ETLSVC schema

Overview of Audit–Control



Maintaining high level audit info ..1

- Provides the following
  - Process Name
  - Subject Area
  - Status
  - Start Date
  - Complete Date
  - Manual Notes
- Audit table – T_EDW_PROCESS_AUDIT
- UNIX wrapper scripts to insert or update the audit table
  - cycle_begin.sh
  - cycle_end_success.sh
  - cycle_end_failure.sh
- Parameters are same for these 3 wrapper scripts
  - Process Type – INFA,ORCL,UNIX or OTH
  - Process Name
  - Subject Area
- Examples:
  - cycle_begin.sh INFA MIDAS_V2 MIDAS
  - cycle_end_success.sh INFA MIDAS_V2 MIDAS
  - cycle_end_failure.sh INFA MIDAS_V2 MIDAS
- Wrapper scripts call the main script – cycle_main.sh
- Script location – /dw001/app/edwetl/scripts/common
- Log files for all scripts are created in the /log folder within the script location mentioned above



Green lines indicate a success path.
Red lines indicate a failure path.

Maintaining high level audit info ..2

- Begin a load cycle
  - Script used is "cycle_begin.sh"
  - Makes a new open entry in the audit table, where COMPLETE_DTE is NULL and STATUS_DESC = 'RUNNING'
  - Checks for a prior open entry, before making a new open entry
    - A prior open entry could mean that the prior run did not complete in its entirety
    - Some data clean-up may be required, before the job is run again
    - This checks helps notify or even remind the support personnel that the prior run was not successful and therefore, further research is required before a re-run
  - Ends the script in failure if a prior open entry exists in the audit table
  - Manual update of COMPLETE_DTE to a date value is required, before a re-run of the process
  - Although not mandatory, it would be a good idea to leave comments in the NOTES_TXT column, while performing such an update

```
$ cycle_begin.sh INFA MIDAS MIDAS_V2
```

Subject Area	Process Name	Start Dte	Status	Complete Dte
MIDAS	MIDAS_V2	20130123 @ 4PM	RUNNING	<null>

If the process ends in failure, the open entry will be updated by script "cycle_end_failure.sh", which will be discussed later. The updated record is given below.

MIDAS	MIDAS_V2	20130123 @ 4PM	FAILED	<null>

If the script is run again for the same subject area & process name, the logic would force the script into failure, because an open entry exists in the audit table for the same subject area & process name. Closing an open entry would be to assign a value to the COMPLETE_DTE column.

MIDAS	MIDAS_V2	20130123 @ 4PM	FAILED	20130123 @ 5PM

Rerunning the script to begin a cycle for the same process will now complete successfully. This is illustrated below.

MIDAS	MIDAS_V2	20130123 @ 4PM	FAILED	20130123 @ 5PM
MIDAS	MIDAS_V2	20130124 @ 4PM	RUNNING	<null>

Maintaining high level audit info ..3

- Ending a load cycle on success
  - Script used is "cycle_end_success.sh"
  - Closes an existing open entry in the audit table, that is COMPLETE_DTE = sysdate and STATUS_DESC = 'SUCCEEDED'
  - Checks for a prior open entry, before closing
  - Ends the script in failure if an open entry does not exist in the audit table for that subject area and process name
  - Additionally, copies the data for the day's run into the table T_EDW_PROCESS_LOAD_STATS from the INFA MX-Views using V_INFAMXVIEW_LOAD_STATS
  - The view V_INFAMXVIEW_LOAD_STATS makes use of DB-Links to access the MX-Views @ INFA repository

Subject Area	Process Name	Start Dte	Status	Complete Dte
MIDAS	MIDAS_V2	20130123 @ 4PM	RUNNING	<null>

```
$ cycle_end_success.sh INFA MIDAS MIDAS_V2
```

MIDAS	MIDAS_V2	20130123 @ 4PM	SUCCEEDED	20130123 @ 5PM

- Running this script for a subject area & process closes an existing open cycle entry and updates the COMPLETE_DTE. This is illustrated above.
- If an open cycle entry does not exist when this script is run, the logic within the script would force the execution to end in failure.
- Running the cycle_begin.sh script for this subject area & process would create a new open cycle, illustrated below.

MIDAS	MIDAS_V2	20130123 @ 4PM	SUCCEEDED	20130123 @ 5PM
MIDAS	MIDAS_V2	20130124 @ 4PM	RUNNING	<null>

Maintaining high level audit info ..4

- Ending a load cycle on failure
  - Script used is "cycle_end_failure.sh"
  - This is similar to it's sibling "cycle_end_success.sh", except that the cycle is left open, but the status is updated to 'FAILED'. In other words, no updates are made to the column COMPLETE_DTE, but STATUS_DESC is set to 'FAILED'
  - Similarly, load information is fetched from the MX-Views in the INFA repository and loaded into the table T_EDW_PROCESS_LOAD_STATS using the view V_INFAMXVIEW_LOAD_STATS via DB-links

Subject Area	Process Name	Start Dte	Status	Complete Dte
MIDAS	MIDAS_V2	20130123 @ 4PM	RUNNING	<null>

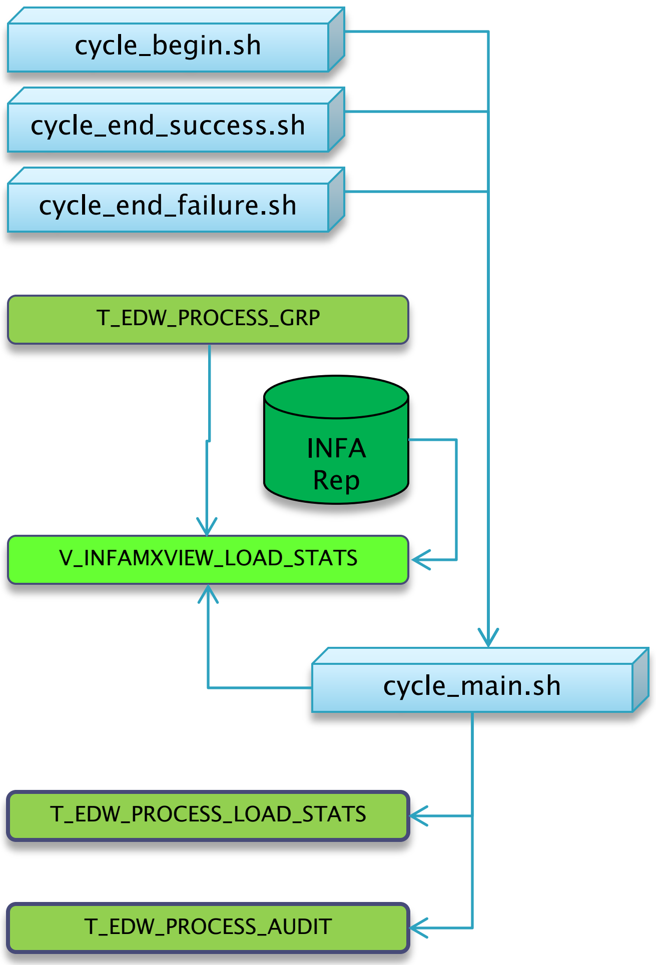
```
$ cycle_end_failure.sh INFA MIDAS MIDAS_V2
```

MIDAS	MIDAS_V2	20130123 @ 4PM	FAILED	<null>

- If an open cycle entry does not exist when this script is run, the logic within the script would force the execution to end in failure.
- Running the cycle_begin.sh script for this subject area & process would end in failure, because the cycle is still open

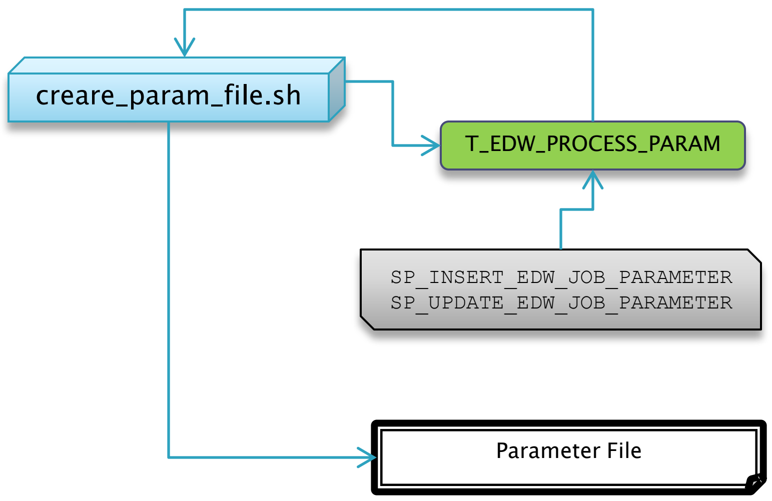
Maintaining high level audit info ..5

- Other objects used to maintain audit information
  - T_EDW_PROCESS_GRP
    - Helps in grouping multiple INFA workflows under a single process
    - This table is maintained manually
    - This table comes into play for –
      - Fetching load information from INFA MX views
      - Preparation of a load report that is sent by email
  - V_INFAMXVIEW_LOAD_STATS
    - This has been discussed in detail in slide #10



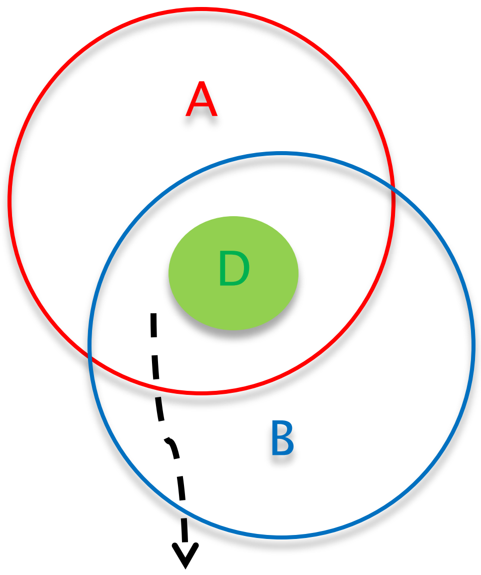
Creating dynamic parameters for ETL

- Useful under following circumstances
  - Delta loads requiring storing the last ETL run date
  - Source files don't have a constant name
  - Any reusable ETL used behave differently based on a parameter
- The objects in play are listed below
  - UNIX script: – create_param_file.sh
  - Table:– T_EDW_PROCESS_PARAM
  - Procedures:–
    - SP_INSERT_EDW_JOB_PARAMETER
    - SP_UPDATE_EDW_JOB_PARAMETER
- Input parameters
  - INFA folder name
  - INFA workflow name
  - Absolute target file path
- Uses spooling logic



Gathering ETL load information

- Fetched from MX views in the INFA repository
- Uses the public DB-Link ETLSVCRO_INFAREP9.WORLD defined in the EDW database to access the INFA repository
- The MX views that have the ETL load information are REP_SESS_LOG and REP_SESS_TBL_LOG
- The view V_INFAMXVIEW_LOAD_STATS is used to fetch the load information from the MX views, which is not already loaded in T_EDW_PROCESS_LOAD_STATS
- Structure of the view
  - Fetch recent records from REP_SESS_LOG (A)
  - Fetch recent records from REP_SESS_TBL_LOG (B)
  - Fetch all records from T_PROCESS_GRP (C)
  - Fetch all the records from T_EDW_PROCESS_LOAD_STATS (D)
  - A join B join C where not exists in D for the current date



This VENN diagram illustrates the view V_INFAMXVIEW_LOAD_STATS. The data in the intersection between A & B, but the matching from D is returned and loaded into the table T_EDW_PROCESS_LOAD_STATS

Refer to the illustration in slide #8

ETL cycle audit report

- Mostly the final step in a cycle
- Fetches data from the following objects
  - T_EDW_PROCESS_LOAD_STATS
  - V_EDW_PROCESS_LOAD_RPT
- Input parameters
  - Subject Area
  - Process Name
  - Number of delinquency days
  - Email Address
- The first 2 parameters are mandatory
- Provides the following reports
  - Data delinquency report – List of tables for which data was not received beyond a certain threshold
  - Logical rejection report – List of tables that contain records, which could not be loaded into the IDR or the mart
  - Load report – A comprehensive report of tables that were loaded in the current cycle with the records loaded
- The data delinquency report is skipped if the 3rd parameter is zero or if only 2 parameters are passed
- If the 4th parameter is not passed or if only 2 parameters are passed, the email address is set to EdwEtlSupport@BaylorHealth.edu
- The script name is cycle_audit_report.sh located at /dw001/app/edwetl/scripts/common
- The log file of this script is sent as the load/audit report
- Parameter driven & generic

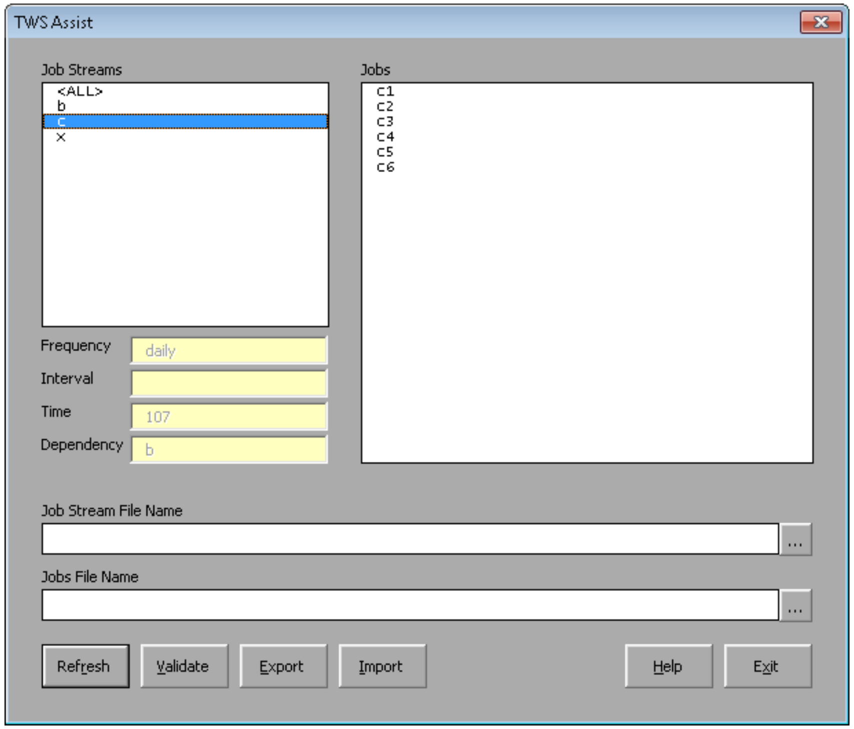
OutlookItem

Other generic functions

- The following generic functions are in etl_func_env.sh at /dw001/app/edwetl/scripts/common .
  - Function_get_logon (not recommended)
    - Fetch password from the hidden file ".etl.logons" located at "/dw001/app/edwetl/scripts/logons"
    - Input parameter = user name
  - Function_get_db_pswd (recommended)
    - Fetch password from the hidden file ".etl.logons" located at "/dw001/app/edwetl/scripts/logons"
    - Input parameter = user name & database name
  - Function_identify_env
    - Sets the name of the environment & EDW database to the global variables. This eliminates the need to change scripts when they are promoted to the next environment.
    - Input parameters = none
  - Function_archive_file
    - Creates a folder with the name of current date in the format YYYYMMDD for archiving inbound or outbound files
    - Input parameter = path in which the archive folder should be created
  - Function_purge
    - Removes sub-directories and files from archive directory based on archive limit.
    - Input parameter = retention limit in number of days
  - Function_notify
    - Function used to trigger email notification
    - Input parameters = email body, email subject, email recipients
  - Function_log – Displays common log output and exits on [ERROR] with email notification.
  - Function_orasql – Executes oracle SQL with a single row and column return value
  - Function_orasql_commit – Executes oracle SQL transaction(s) to be committed
  - Function_orasql_spool – Executes oracle SQL transaction(s) to be spooled to a file

Excel App for TWS

- Easy creation of the following Tivoli objects
  - Jobs
  - Job Streams
  - Dependencies
  - Schedule
- Saves time
- Minimizes errors
- More enhancements planned
- Used for MIDAS core measures
  - Schedule prepared within a day for many jobs, job streams and dependencies



Frequency	daily
Interval	
Time	107
Dependency	b