SAS Data Integration Studio Take Control with Conditional & Looping Transformations

Harry Droogendyk – Stratia Consulting Inc.

Introduction

- DIS is real good at linear
- not linear ?
  - we love macro, roll yer own non-linear
- why use DIS *properly* ?
  - metadata trail
  - maintenance
- non-linear Transformations
  - Loop
  - Conditional

Business Need

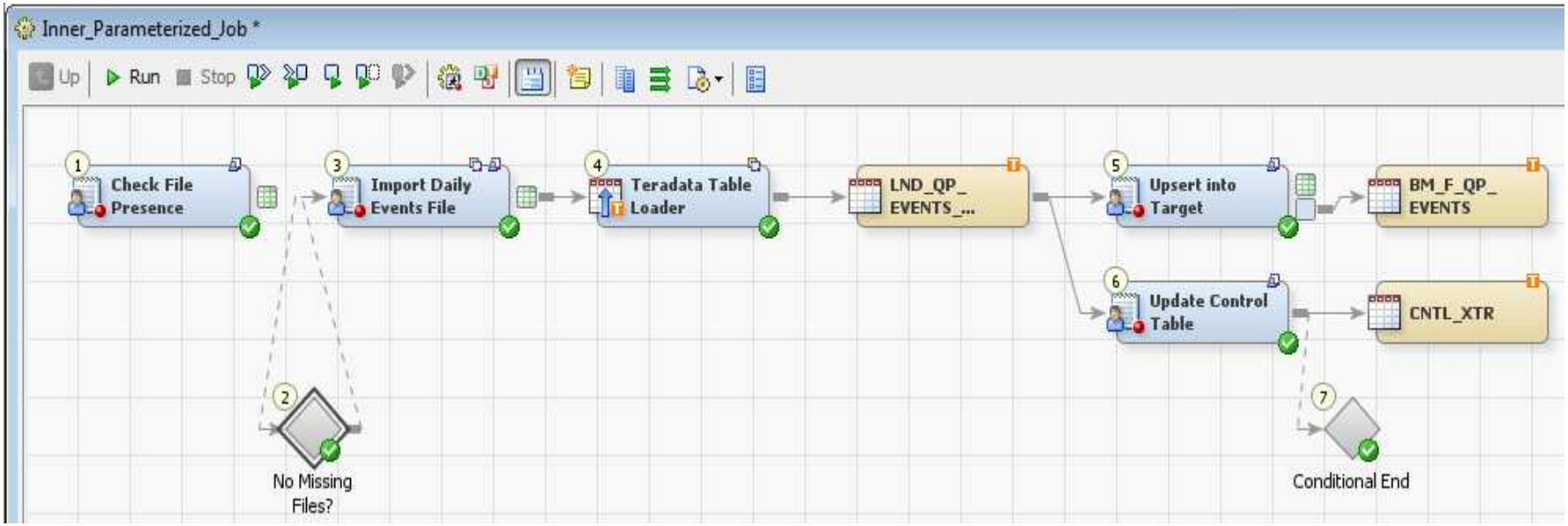
- 3rd party supplied text files
  - imported into SAS → Teradata
- daily – kinda
- must be processed in date order
  - control table
- if a day is missing...
  - send email
  - continue to look for files
  - don't import any more data

Technical Solution

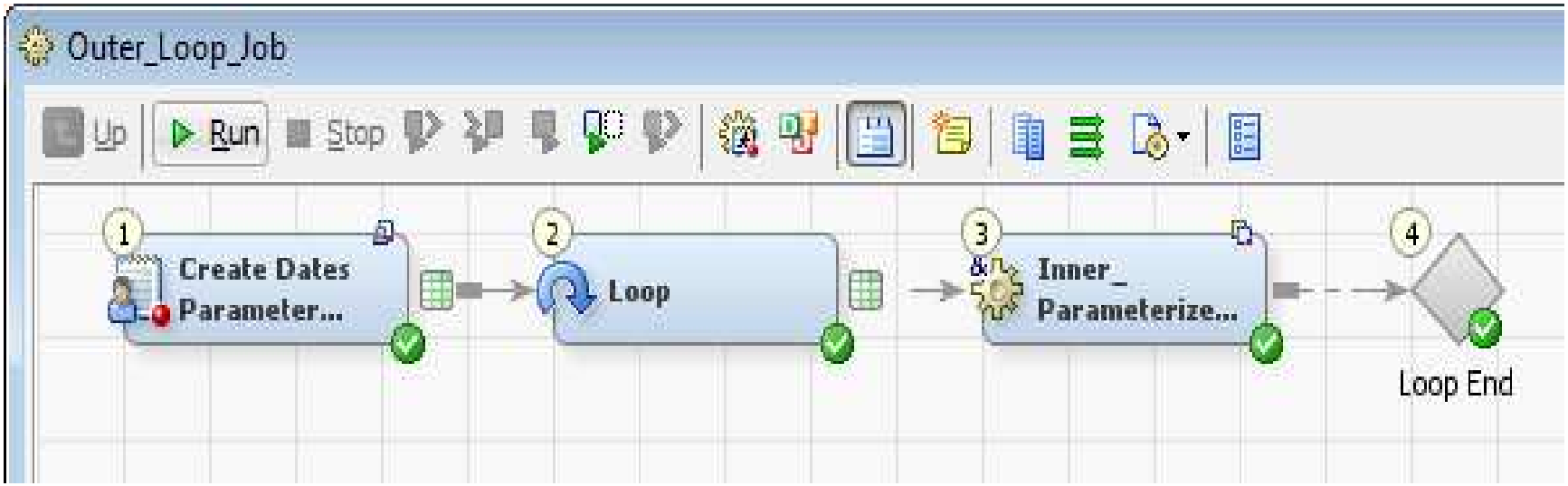
- outer job - looping
  - read control table to get last imported date
    - create driver table of dates to process - last+1 to today-1
  - loop over driver table
    - pass next date to inner, import job
    - execute inner job

- inner job – looped over
  - accept next date to process from outer job
  - check if file for this date exists
  - send email or import
  - stop importing after missing file is detected

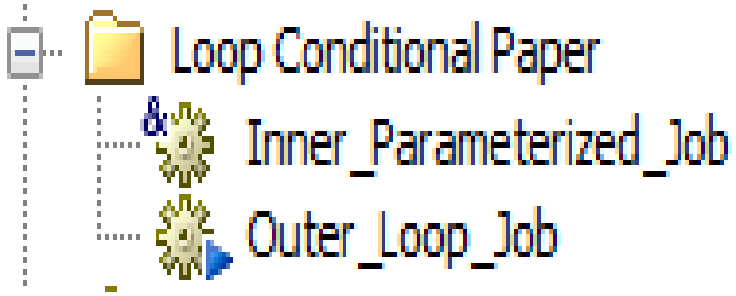
DIS Jobs


Inner_Parameterized_Job *
Up
Run
Stop
1
Check File Presence
3
Import Daily Events File
4
Teradata Table Loader
LND_QP_ EVENTS_...
5
Upsert into Target
BM_F_QP_ EVENTS
6
Update Control Table
CNTL_XTR
2
No Missing Files?
7
Conditional End

DIS Jobs

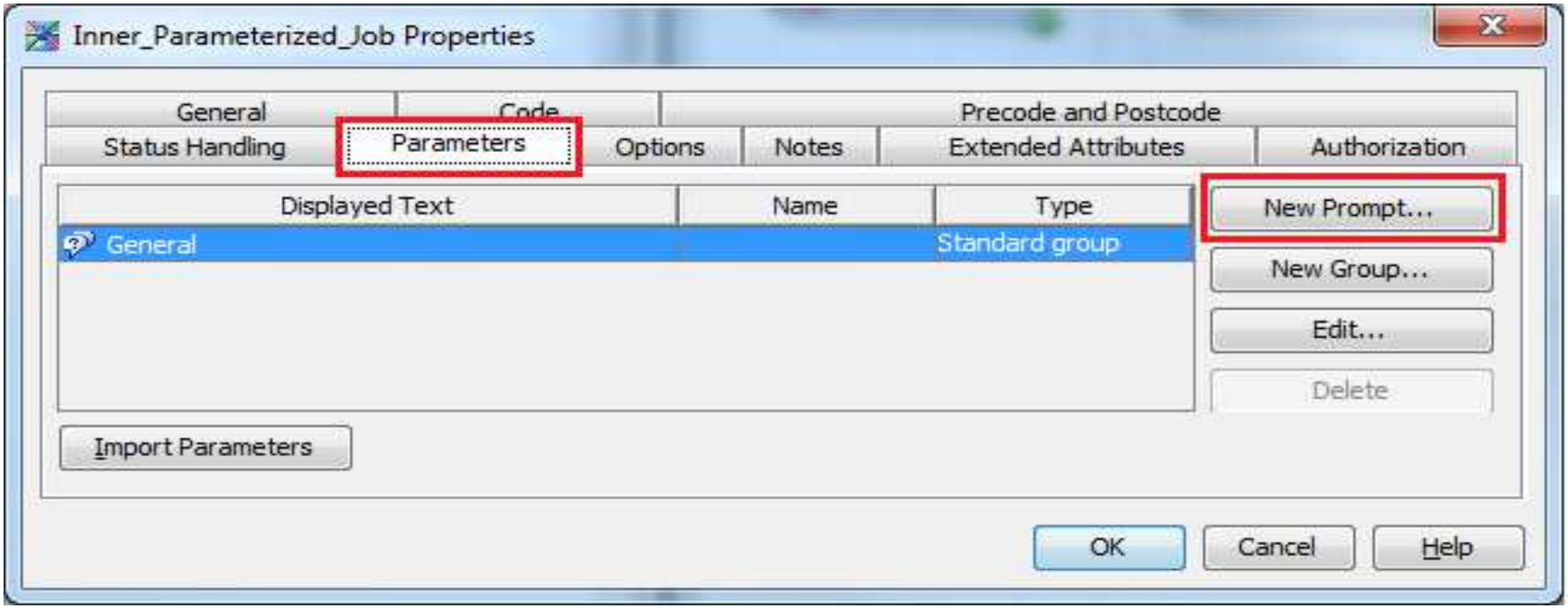

Outer_Loop_Job
Up
Run
Stop
1
Create Dates Parameter...
2
Loop
3
Inner_ Parameterize...
4
Loop End

DIS Jobs



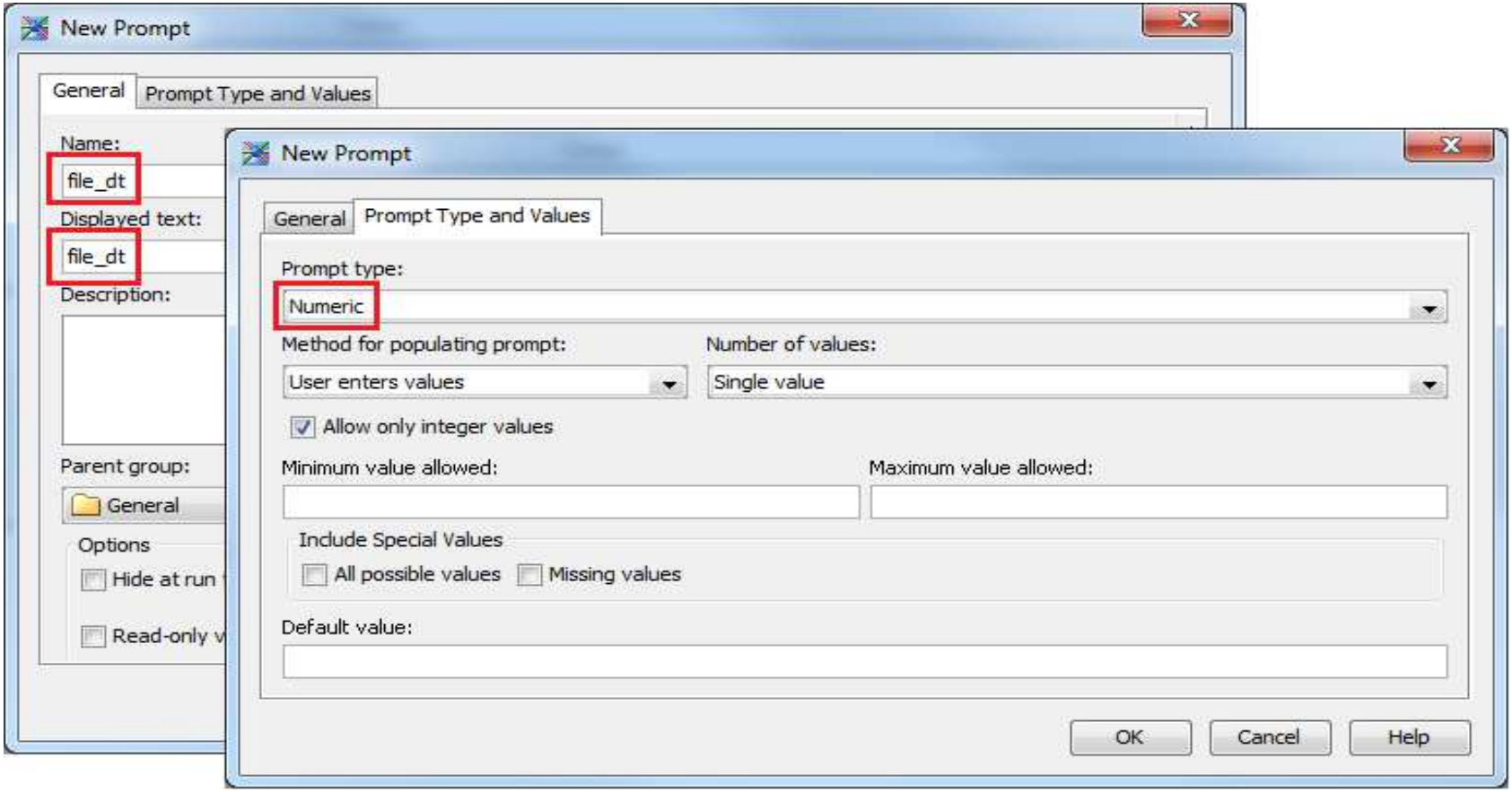
- note the icons
  - & is a job with a parameter
  - blue arrow is a deployed job

Creating Inner Job



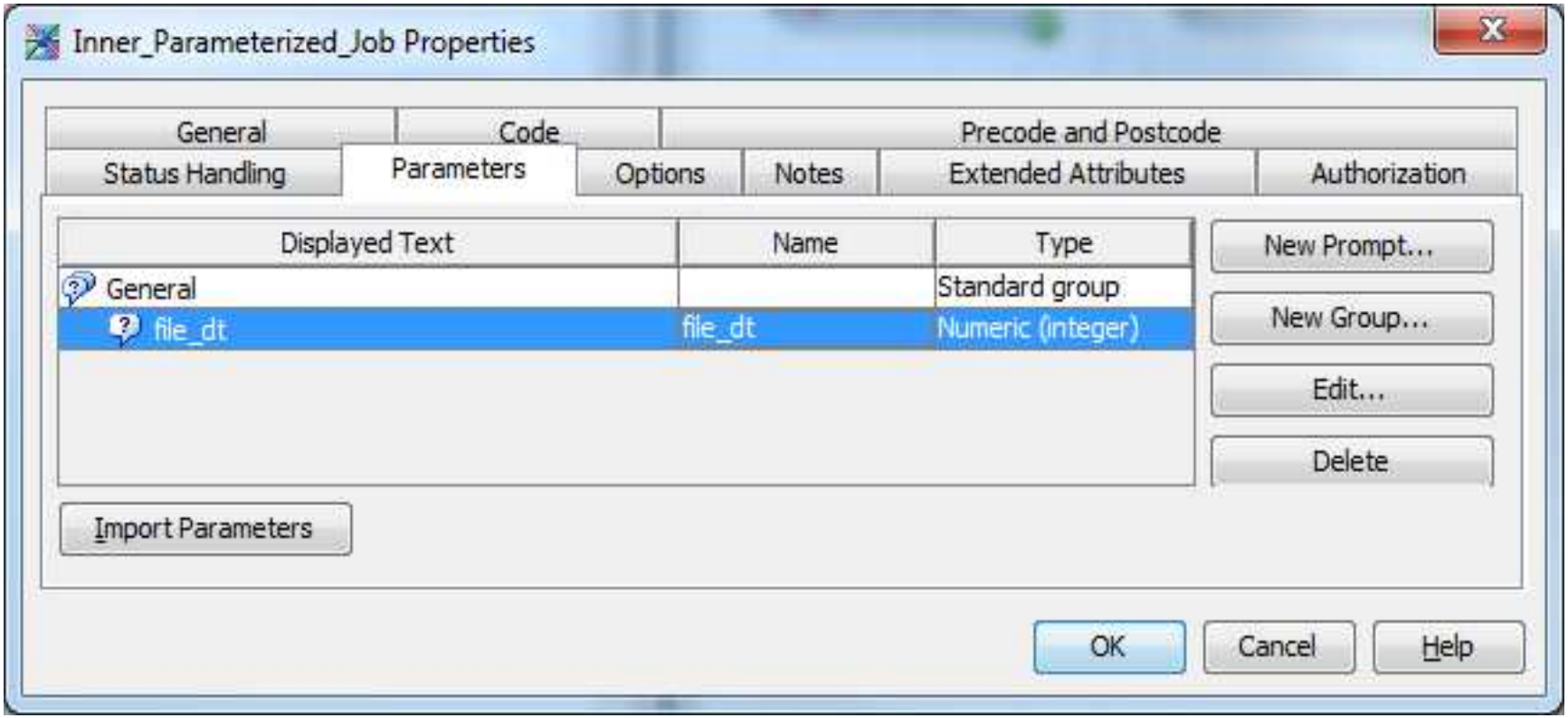
- Job Properties
- prompts aren't magic, simply macro vars

Creating Inner Job



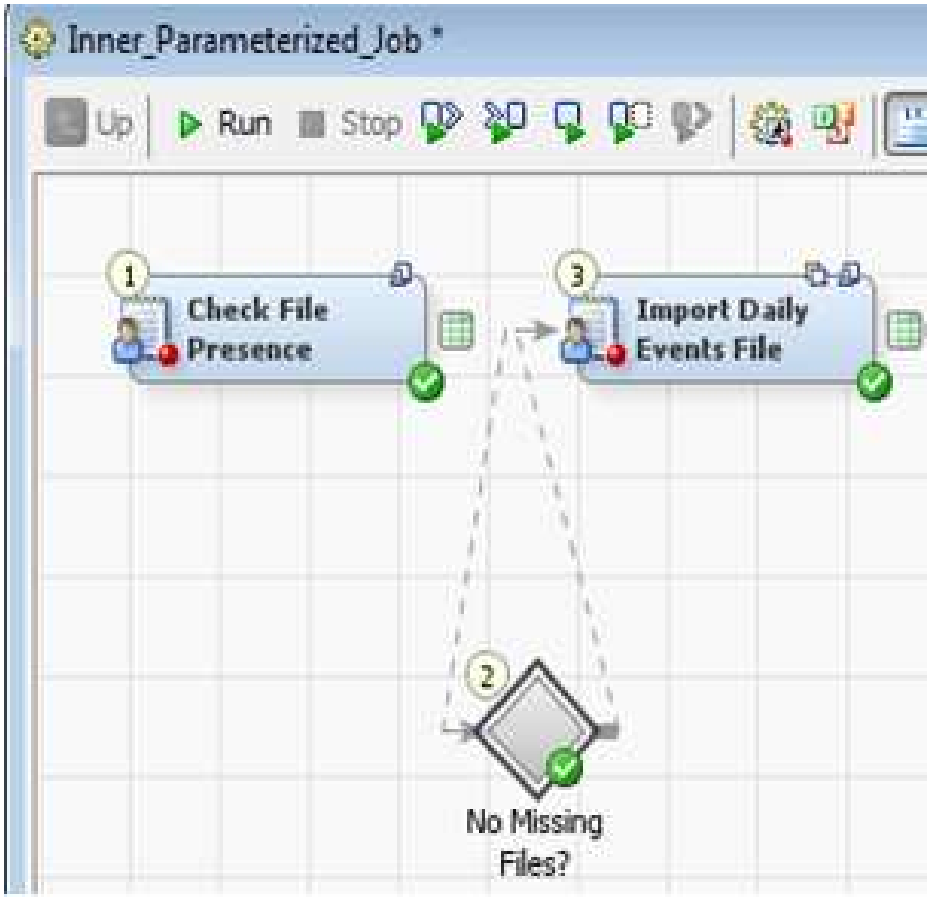
- name must be macro variable name in code

Creating Inner Job



- completed parameter

Inner Job – Set Condition



- Check File Presence – code node

Inner Job – Set Condition

- using the job parameter

```
%let fileloc    = /data/sharedall/mifeeds/quickplay;
%let filename   = QuickPlay_BELL-DAILY-Events-TV5-TEST-%sysfunc(putn(&file_dt, yymmddn8.))*.csv;

%put Looking for filename: &filename;

filename ls pipe "ls &fileloc./&filename";
```

```
data _files_to_process;
    length path_file $256;
    infile ls;
    input;
    put _infile_;

    * if the asterisk is found in the ls output, it means the file was not found ;
    if index(_infile_,'*') = 0 then do;
        path_file = _infile_;
        output;
    end;
run;
```

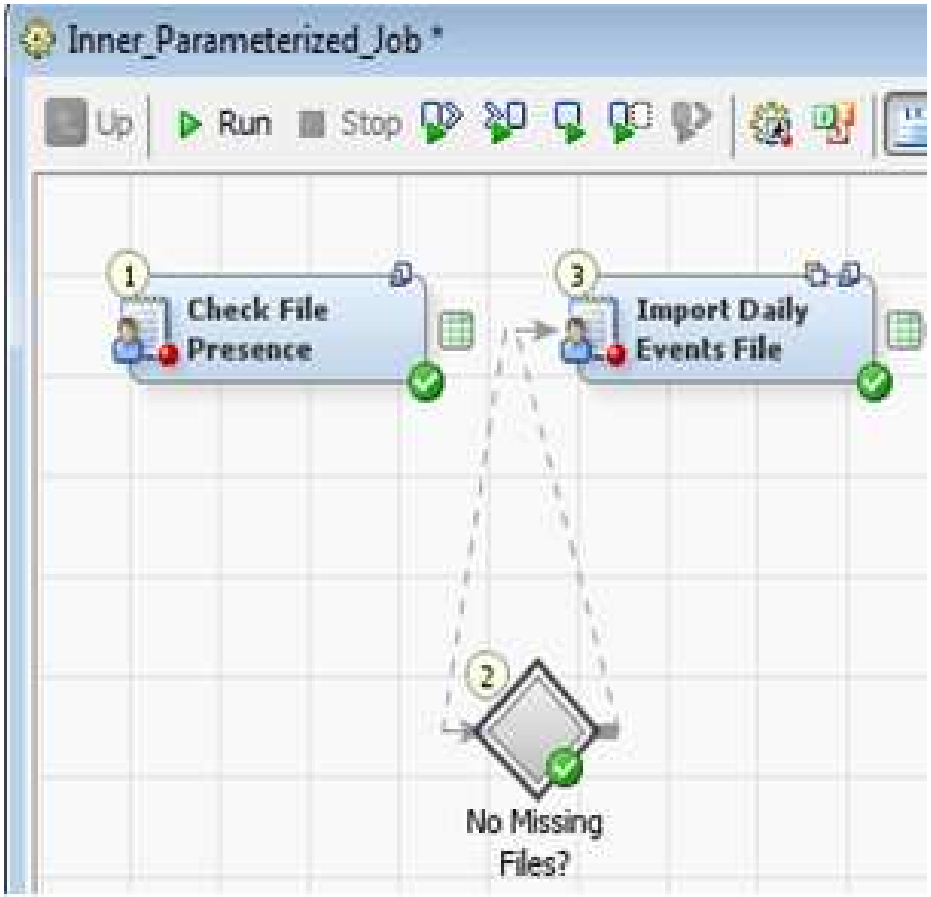
```
%check_files
```

Inner Job – Set Condition

- checking if file exists

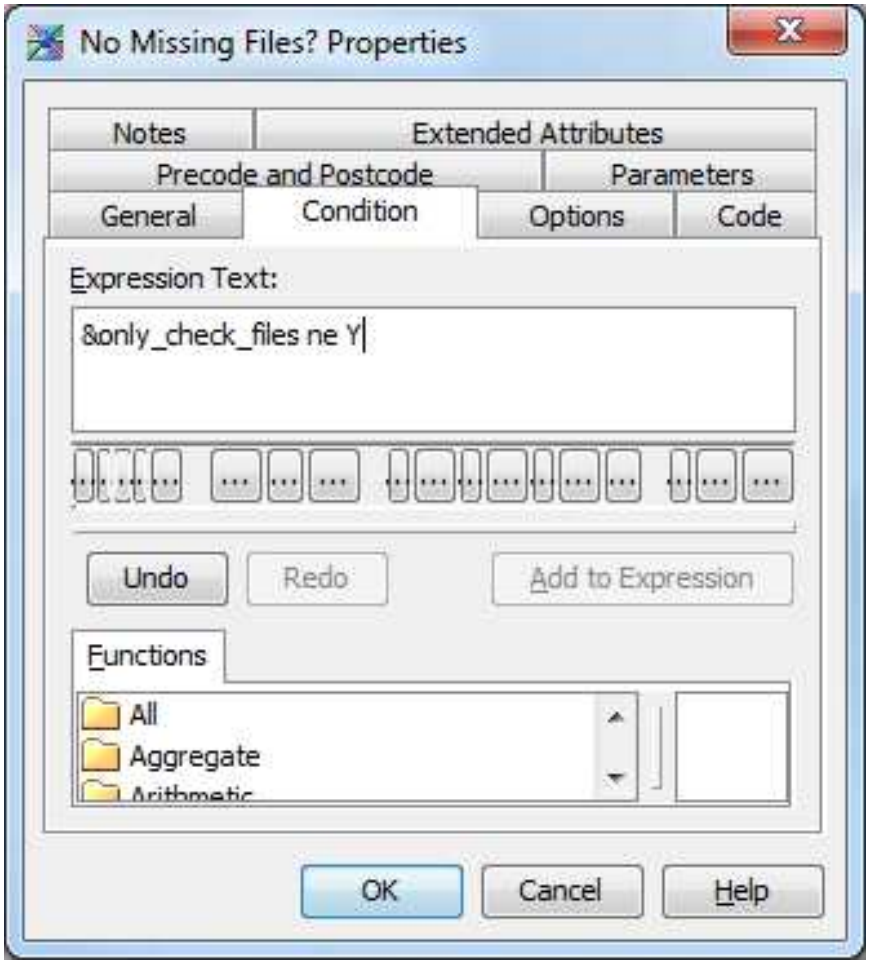
```
%macro check_files;

    %global files_found
            only_check_files;

    %let files_found = %attrn(_files_to_process,nobs);

    %if &files_found = 0 %then %do;

        %email(
           Report_Key    = Quickplay CSV Issue,
           email_from    = %str(t                    ),
           Subject       = Missing data file &filename,
           greeting      = %str(Quickplay folks:),
           message       = %bquote(Please (re)send &filename., not found on Bell servers.),
           bye           = %bquote(Regards,`Bell Client Insights` `&etls_jobname)
            );

        /*
            The looping process is still dependent on files with consecutive dates being processed.  If a
            file is missing, going back to process it is a manual process, so set a switch so all we do
            from here on is check for the presence of files.
        */

        %let only_check_files = Y;

    %end;

%mend;
```

Inner Job – Check Condition



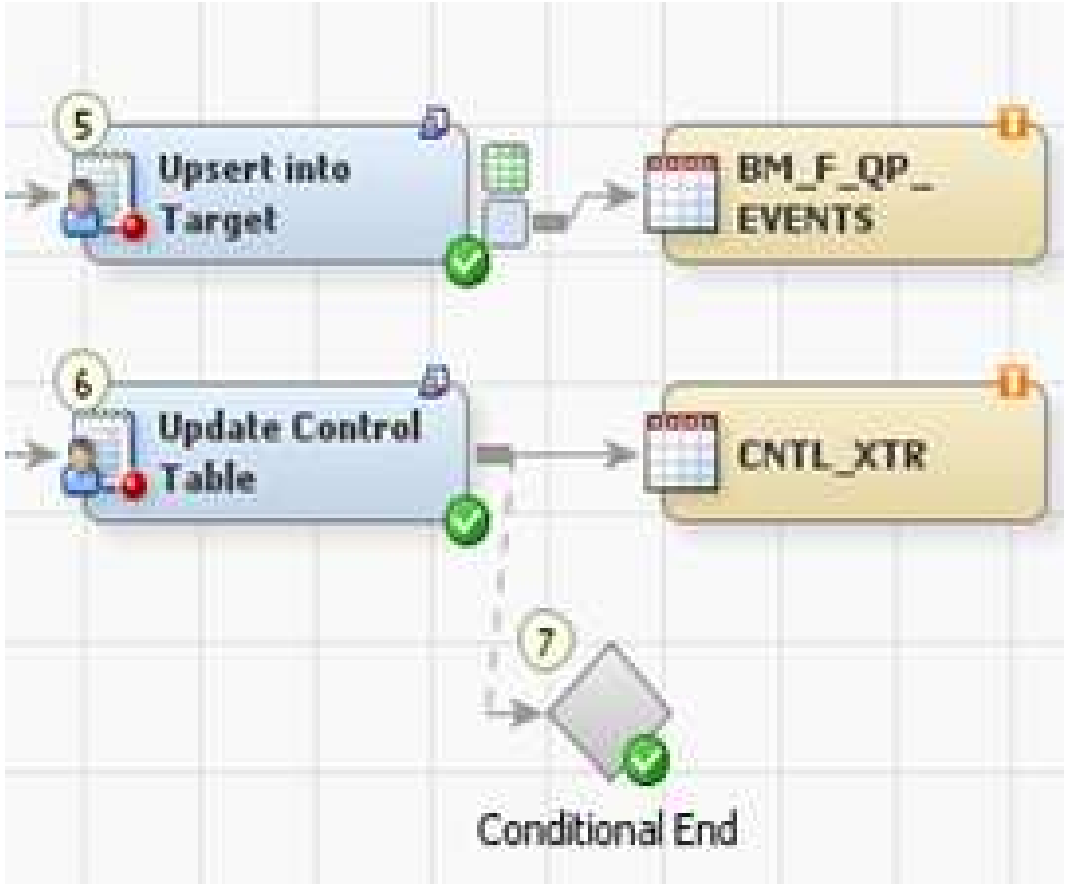
- No Missing Files? **Conditional Start** Transformation

Inner Job – Check Condition



- continue if TRUE

Inner Job – Check Condition



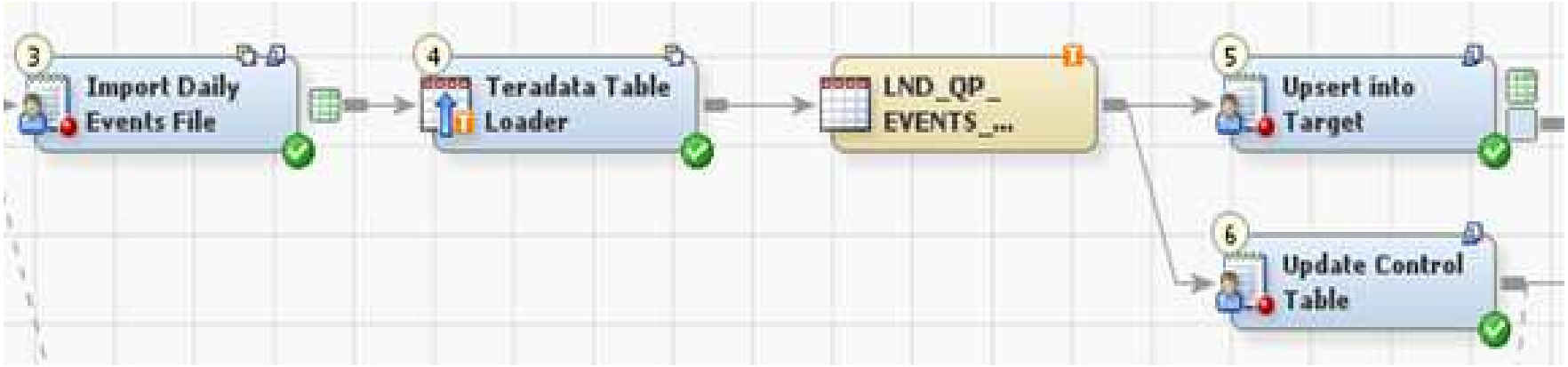
- Skip to **Conditional End** if FALSE

Inner Job – Check Condition

```
1   /*=============================================================================*
2    * Step:            No Missing Files?                    A58T40E2.BY001UWS *
3    * Transform:       Conditional Start                                      *
4    * Description:                                                           *
5    *=============================================================================*/
6
7   %let transformID = %quote(A58T40E2.BY001UWS);
8   %let trans_rc = 0;
9   %let etls_stepStartTime = %sysfunc(datetime(), datetime20.);
10
11  %macro etls_conditionW6VDGVD;
12     %local etls_conditionTrue;
13     %let etls_conditionTrue = %eval(&only_check_files ne Y);
14     %if (&etls_conditionTrue=0) %then
15     %do;
16        %put ETLS_DIAG: Condition flow did NOT execute, condition was &only_check_files ne Y;
17        %goto exitetls_conditionW6VDGVD;
18     %end;
19     %else
20     %do;
21        %put ETLS_DIAG: Condition flow did execute, condition was &only_check_files ne Y;
22     %end;
23  %exitetls_conditionW6VDGVD:
24  %mend etls_conditionW6VDGVD;
25
26  %etls_conditionW6VDGVD;
```

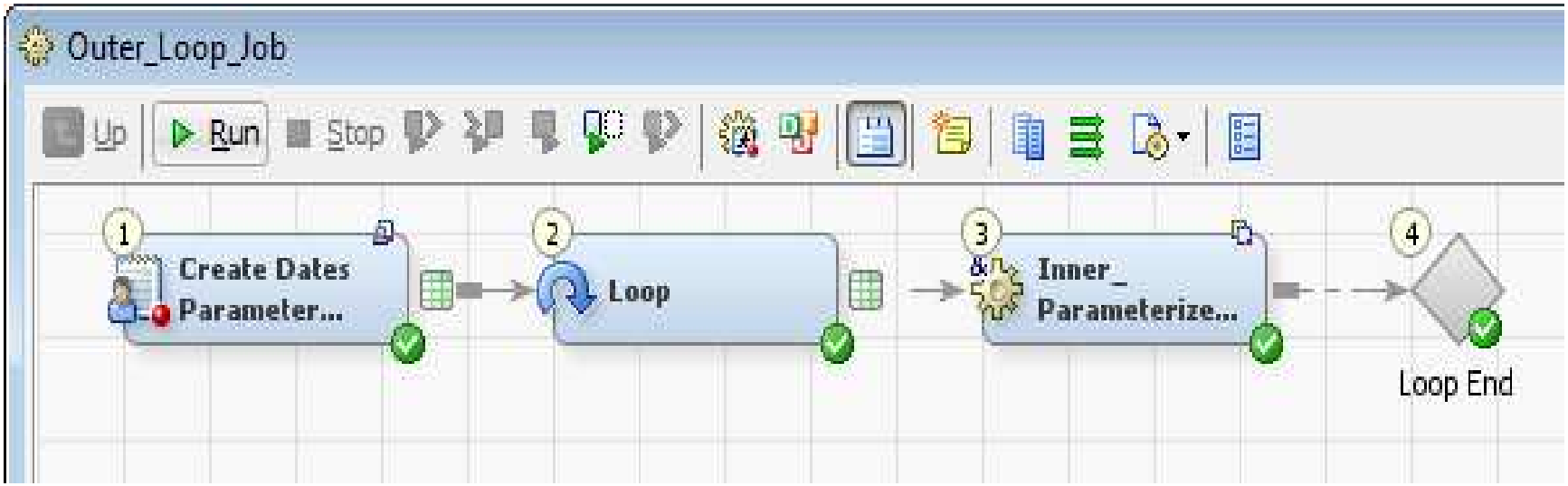
- Conditional Start generated code

Inner Job – real work



- if TRUE, the real work occurs
  - everything between Conditional Start and Conditional End

Outer Job



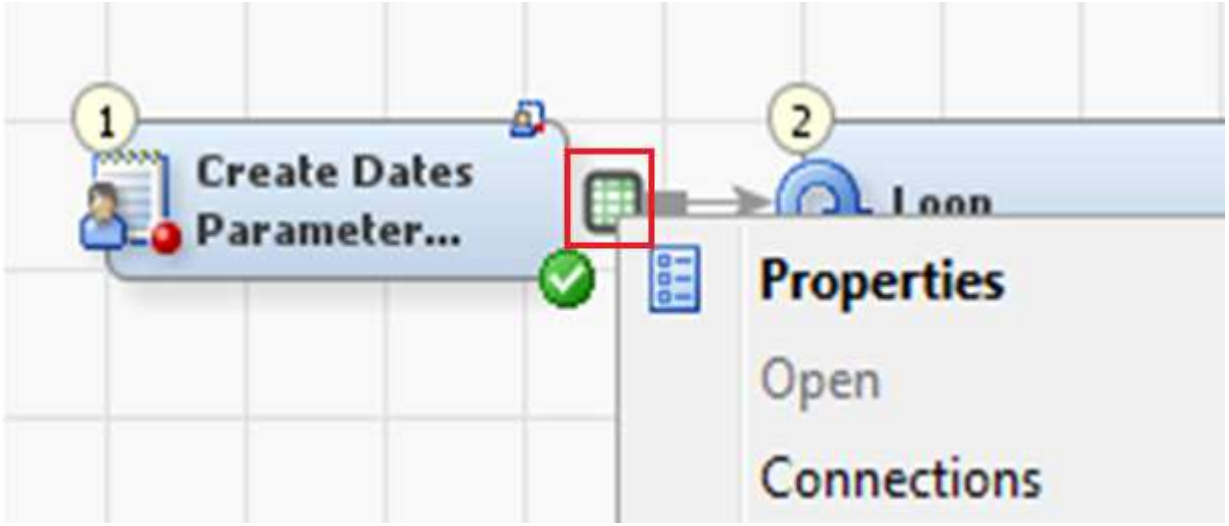
- Create Dates Parameter – code node
- Loop
- inner job
- Loop End

Outer Job – Create Dates Parms

```
/*  CNTL_XTR records the last successful run, bump it by one.  */

%let _xtrFromDTM = %sysfunc(datetime(),datetime19.);

proc sql noprint;
    select max(xtrToDTM)
      into :_xtrFromDTM
      from mi_cntl.cntl_xtr;
quit;

data qp_dts;

    do file_dt = datepart("&_xtrFromDTM"dt ) + 1 to today() ;
       output;
    end;

run;
```

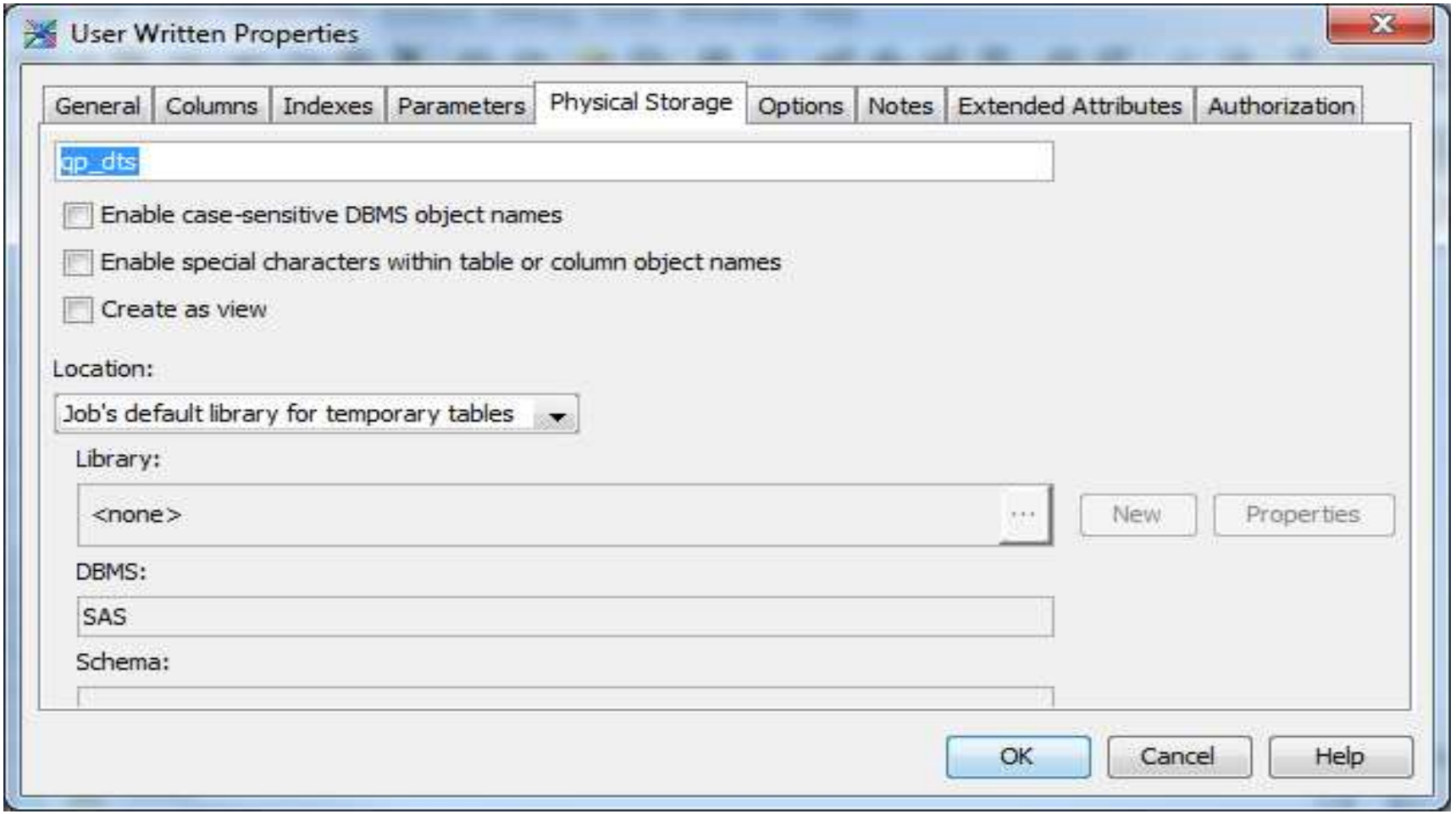
- read Control table to get last date
- data set with dates to process

Outer Job – Create Dates Parms



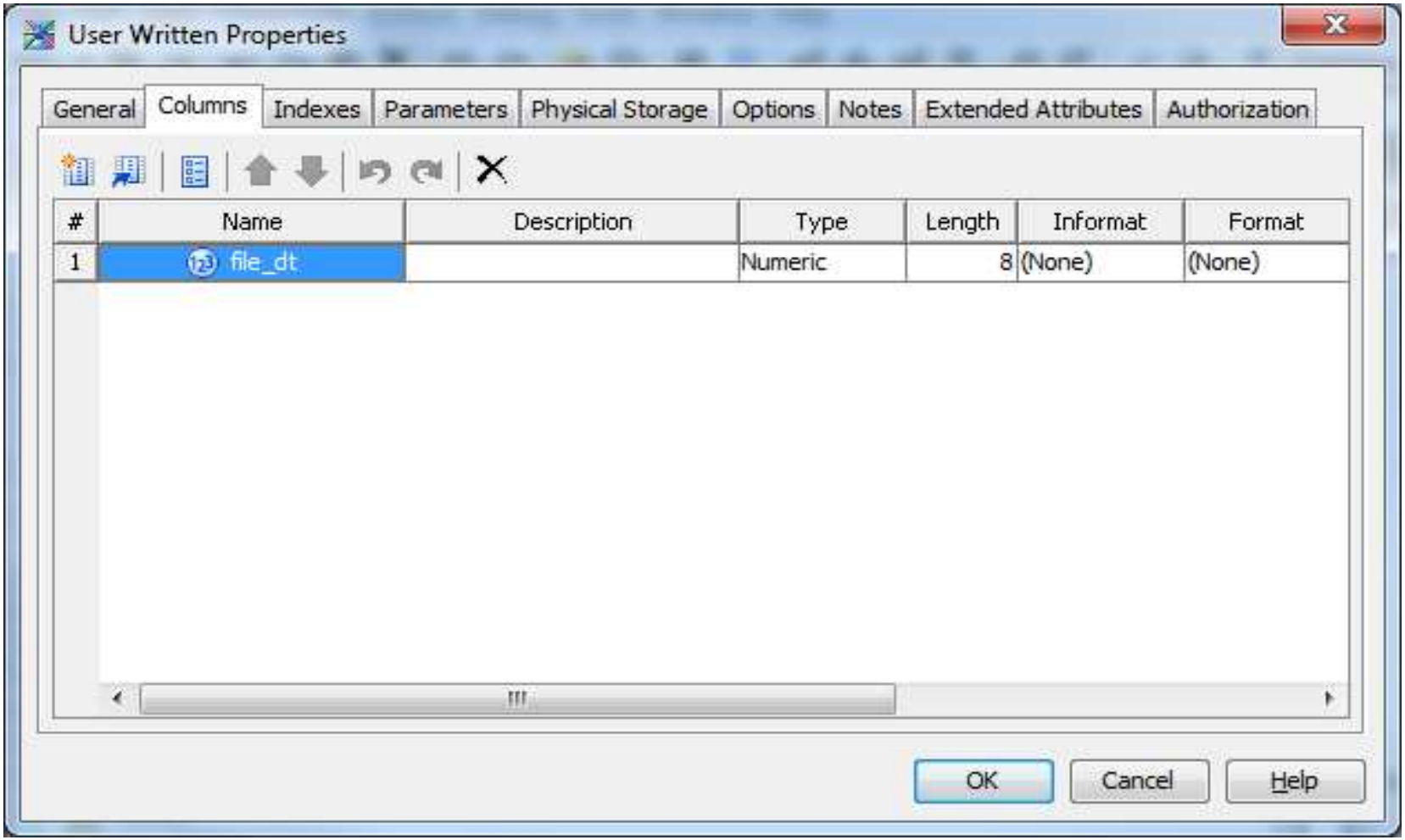
- right click on green grid, output data set of xform
  - properties
- when the job runs, output data set will have *n* rows

Outer Job – Create Dates Parms



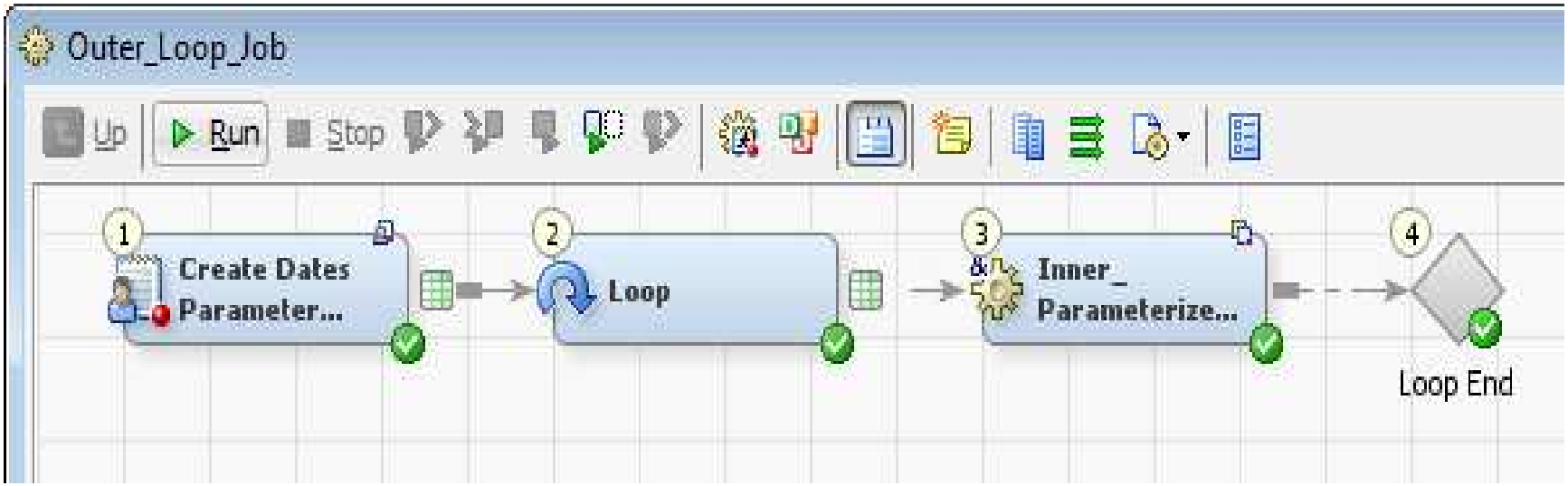
- name the output data set

Outer Job – Create Dates Parms



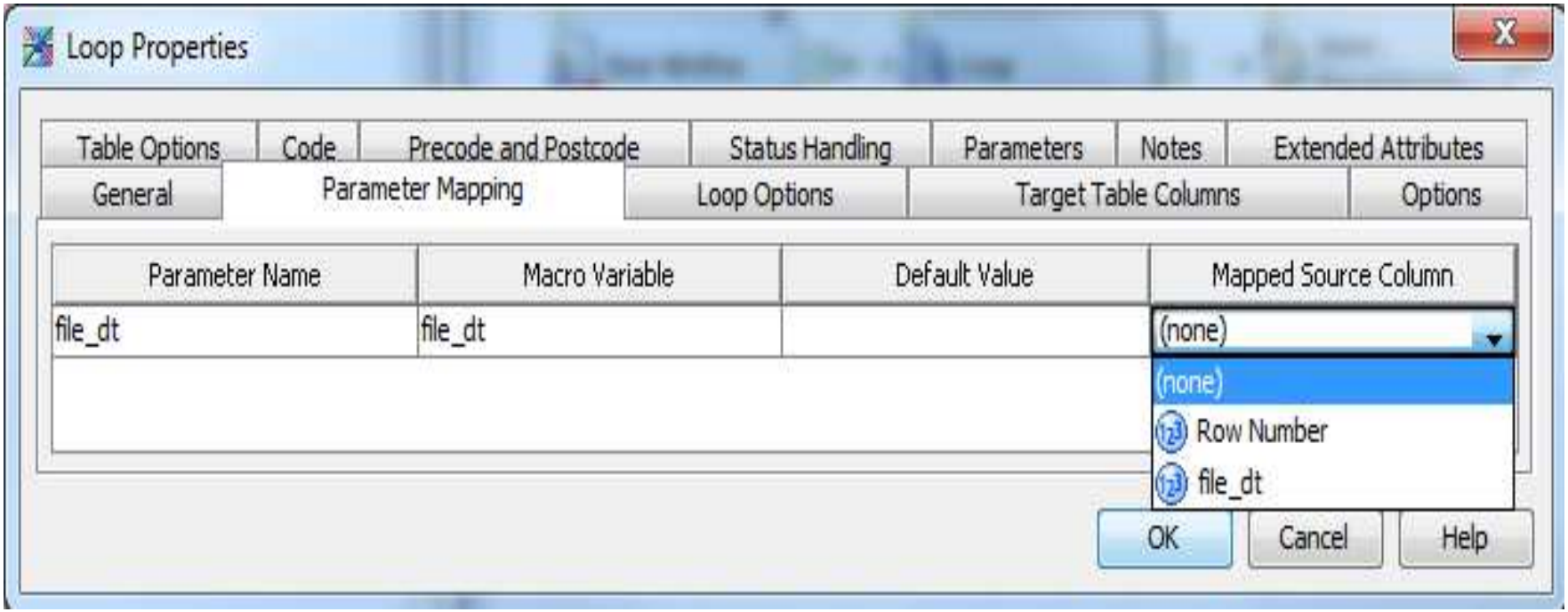
- define the file_dt column

Outer Job – Loop, Inner Job



- order is important – DIS will help you
  - parameter data set
  - Loop transformation
  - Inner Job

Outer Job – Loop, Inner Job



- DIS "sees" data set columns and Inner Job parameter
  - Loop Properties
  - map the parameter
  - select source column
- add Loop End, save

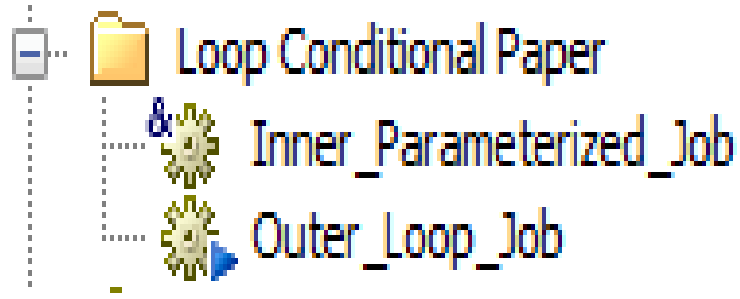
Outer Job – Loop code

```
276  %macro etls_loopW764KHV;
277     %local etls_filePrefix;
278     %let etls_filePrefix = ;
279
280     %macro etls_processToLoopW764KI5(parameterTable=, row=, handleName=rmt);
281        %local etls_parmvars;
282        %etls_getParameterNames(parameterTable=&parameterTable,
283           parameterVariableMacro=etls_parmvars,
284           startingColumnNumber=1);
285        %local &etls_parmvars;
286        %etls_getParameters(parameterTable=&parameterTable, row=&row,
287           startingColumnNumber=1);
288        %let etls_previousFilePrefix = &etls_filePrefix;
289        %local etls_filePrefix;
290        %let etls_filePrefix = &etls_previousFilePrefix.&handleName;
291        %macro etls_jobW764KIF;
292
293           /* Setup to capture return codes  */
294           %global job_rc trans_rc sqlrc syscc;
295           %let sysrc = 0;
296           %let job_rc = 0;
297           %let trans_rc = 0;
298           %let sqlrc = 0;
299           %let syscc = 0;
300           %global etls_stepStartTime;
301           /* initialize syserr to 0 */
302           data _null_; run;
303
304           %macro rcSet(error);
305              %if (&error gt &trans_rc) %then
306                 %let trans_rc = &error;
307              %if (&error gt &job_rc) %then
308                 %let job_rc = &error;
309           %mend rcSet;
310
311           %macro rcSetDS(error);
312              if &error gt input(symget('trans_rc'),12.) then
313                 call symput('trans_rc',trim(left(put(&error,12.))));
314              if &error gt input(symget('job_rc'),12.) then
315                 call symput('job_rc',trim(left(put(&error,12.))));
316           %mend rcSetDS;
317           /*************************************************************************
318            * Job:            Inner_Parameterized_Job              A5FPY8BO.C0000PFV *
319            * Description:                                                           *
```

- 316 lines of goobly-gook
- Inner Job code is in Loop xform

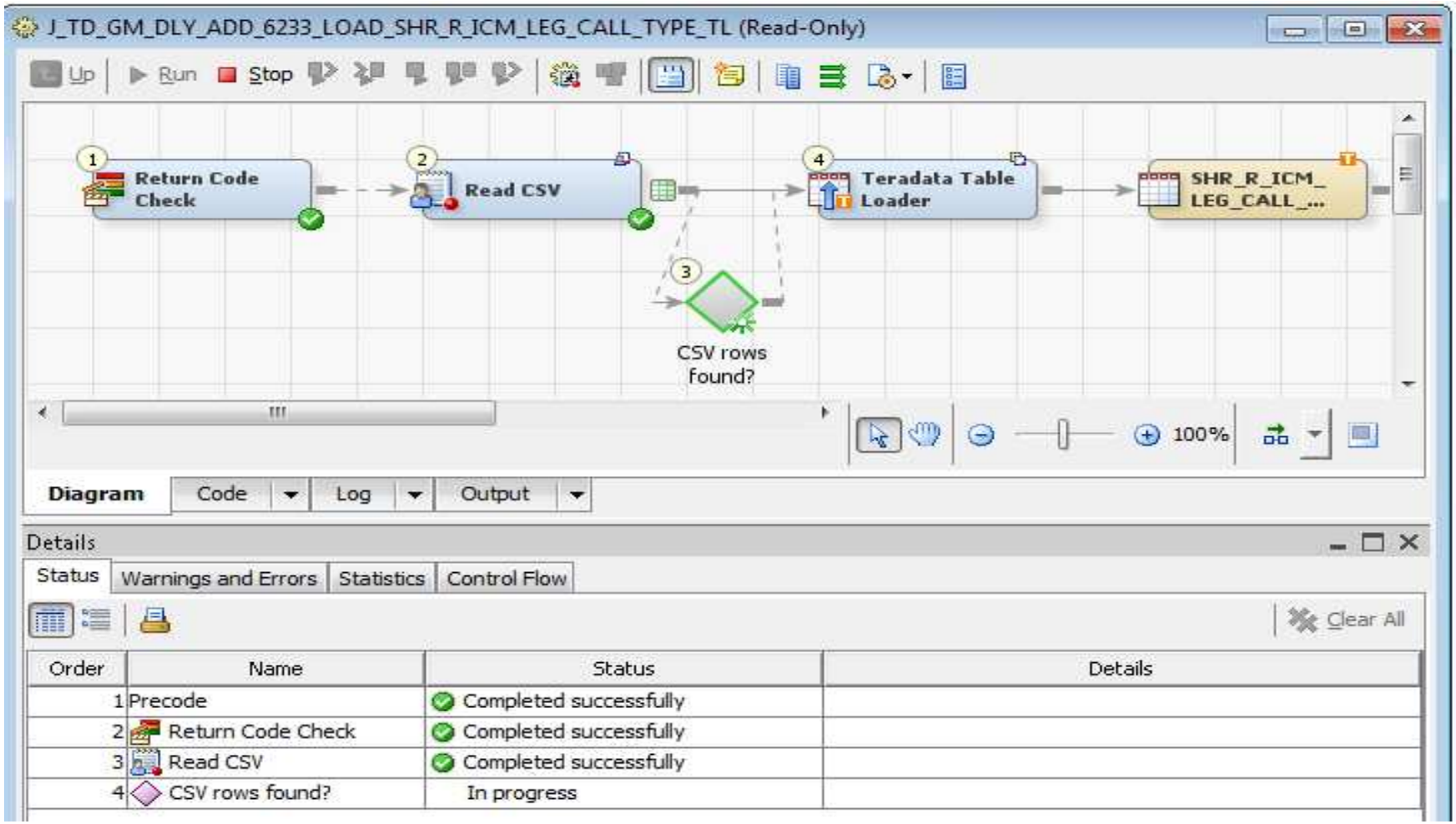
Finished Result

- generated Loop transformation code *included* Inner Job code
- recall the Job folder display
  - only the Outer Job was deployed
  - if changes are made to either job...
    - must re-deploy Outer Job

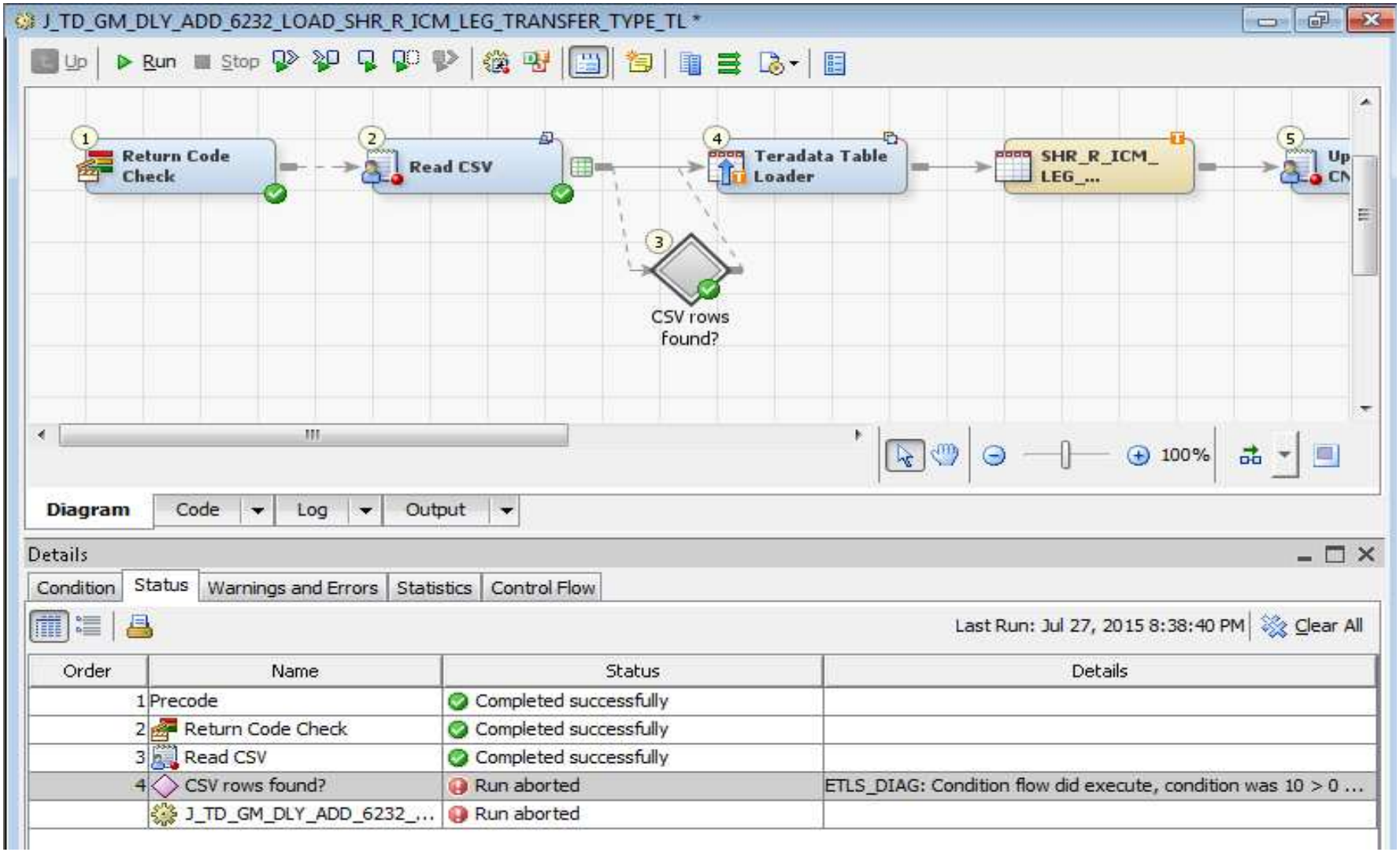


- multiple jobs within a loop
- multiple parameters
- execute loops in parallel in grid environment

Unexpected Behavior - kinda



Unexpected Behavior - kinda



Wrap

- DIS Loop & Conditional Transformations
  - stay in the DIS envelope
    - metadata trail
    - maintenance
  - non-linear processing

Harry Droogendyk harry@stratia.ca
www.stratia.ca