International Journal of Trend in Scientific Research and Development (IJTSRD)
Volume: 3 | Issue: 2 | Jan-Feb 2019 Available Online: www.ijtsrd.com e-ISSN: 2456 - 6470

A Complete Reference for Informatica Power Center ETL Tool

Abhishek Gupta

Associate Projects, Cognizant Technology Solutions, Chennai, Tamil Nadu, India

ABSTRACT
today we are living in the world of data science, where we have to handle bulk amount of data to run any organization. To accomplish the goal of any organization it's mandatory to take right decision at the right time. For this data is maintained in the form of data ware housing and for the extract, transform and load majorly informatics power center tool is used by organization. So in this paper we have shared the complete informatics power center logics, that will be useful not only for organization's but also be useful for data scientists as a complete reference.

Keywords: *debugger, performance tuning, udf*

INTRODUCTION
INFORMATICA – the most recent Informatica tool is 10.1

1. Active and Passive transformations –
Active transformations are those transformations which have following things –
1. Number of incoming and outgoing rows are different.
2. Change in the row type. Eg. Update strategy.
3. Change in the transaction boundaries. Eg. Transaction control.

Else it's passive transformation.

2. Connected and un-connected transformations –
if the transformations are connected within pipeline then connected else it's un-connected transformation.

3. Load types in database -
In database the loading can be done through two types which are – normal load and bulk load (with indexing and without indexing).

4. Debugger –
To see how data is moving from source to target, it's required. Debugger has two windows which are – target window and instance window.

How to use debugger –
1. Press f9 – debugger starts.
2. choose any one session out of three sessions– i.s. to use =>
3. Use an existing session instance.
4. Use an existing reusable session.
5. Create a session debug instance.

Discard target data – if checked no data will go to the target.

Mapping => debugger => next instance

You can use edit break point option of the debugger. If breakpoint = true, i.s. pauses run of debugger. You can review and modify the transformation data and continue the session.

Debugger has three states – initializing, running, paused.

5. Mapplet –
Goto mapplet designer and create mapplet. Mapplet should get i/p from source definition or i/p transformation similarly it must give o/p to the o/p transformation, traditional target. Source will be one output can be multiple. It can be drag and drop. It can be expanded or un-expanded. Changes done in mapplet inherited by mapping.

Un-supported repository objects –
1. Cobol source definition.
2. Joiner, normalizer transformations
3. Non-reusable seq generator.
4. Pre and post session stored procedure.
5. Target definition
6. Xml source definition

If you are using stored procedure transformation, configure it to normal.

6. UDF –
To create complex functions using builtin functions. This can be used in multiple mappings after creation.

There are two of UDF –
1. private (can be used by all users' in repo. It can be transformations, w/fs, link condition and cmd task).
2. public (can be used only inside public udf).

Goto mapping designer => udf folder

Right click on it and click on new and then launch editor after specific entries and when passed on launch editor it show expression editor where you can define udf and save it. Now it's possible to access it through any expression editor and can be called as :udf.udf_name(param_list);

Things that are provided in udf are – name, description, public/private, expression, arguments.
1. private udf can be declared as public udf but public udf can't be declared as private udf.
2. nesting of same udf is not possible.

7. Workflow Basics –
Workflow executions can be done in two ways – 1. Serial based 2. Event based.

In workflow tasks can be placed in serial, parallel, or any required format. Start task come by default whenever workflow is created by double click on it we can edit name and description.

To show always full name of task – tools => options => general => show full name of tasks.

Connections are created so that i.s. can read/write data from/into source or target. Connections that can be created in w/f are relational, ftp, queue and application.

To create connection –
Connection menu => select connection type => a wizard will open => new => give credential to connect => close.

Components of w/f manager – task developer, worklet designer, workflow designer.

Workflow variable –
It's used to share information between two tasks. For eg. 2nd session will run only then when 1st session get completed. Double click on the link and set expression like $cmd_create_folder.status = succeeded. Next session will run only then when 1st session get satisfied.

Workflow => edit => variable.

Workflow parameter –
Can be defined in the parameter file and give it's path to the workflow.

Workflow => edit => properties => parameter filename (place full path)

It can be used to define connection for source and target. Eg. $DBConnection_source place it in the file.

8. Worklet –
Group of tasks, linked with each other. Types – reusable and non-reusable.

9. Work flow Schedulers –
Can schedule when it should run. Workflow => schedulers => new => schedule.
1. run continuously.
2. run on demand.
3. run on i.s. initialization.

Other options (when working with run on i.s. initilization) for scheduler option
Run once –
1. Start option – start date/start time.
2. For scheduler options –
   A. run once
   B. run every _days _ hours _ minutes
   C. customize repeats – there are so many options.
3. A. End point – end on _ date
   B. end after _ runs
   C. forever.

Scheduling workflow –
1. Reusable schedulers – control-m
2. Non-Reusable schedulers – defined inside w/f itself. (conventional w/f)

10. Workflow Wizard –
used to generate scheduled w/f for mapping automatically.
1. workflow => wizard (give name, desc, server) => next
2. select mapping => schedule workflow => next => finish.

11. Stop and Abort –
If a session or w/f is taking much time than usual stop/abort it to free resources (cpu, memory etc.) to give it to another process. Stop/Abort can be done in multiple ways – 1. Using PMCMD command. 2. Using Control task 3. Using workflow monitor.

Difference between stop and abort –
Stopping the session –
1. stop reading data from source.
2. Wait till rollback complete.
3. Cleans buffer memory, so data remain un-written inn database.

Aborting the session – it has time out of 60 seconds. If it can't finish processing and committing data within timeout period it kills the dtm process and terminate session. This leaves the memory block and causes memory issues in server and leads to poor performance.

12. Concurrent run of workflow –
Say, when processing transactions using w/f, data belonging to multiple entities (eg. Regions) may be present in different flat-files and available for processing at the same time. The usual approach is to process them sequentially by running same w/f for each flat-file to reduce processing time it's required to run a separate instance of w/f for each flat file concurrently by using concurrent execution feature of the w/f.

A w/f configured for concurrent execution can run multiple instances concurrently.
- w/fs => edit => configure concurrent execution _ (click onn concurrent execution).

In concurrent execution configuration window will chose any one of below:-
1. allow concurrent run with same instance name.(different run-ids)
2. allow concurrent run only with unique instance name.(different instance names).

Add instance name and param file path as many that many youu want to run parallel.

Eg .
Instance name parameter file
CA_Order $PmSourceFileDir\param_ca.txt
RI_Order $PmSourceFileDir\param_ri.txt

These kind of w/f are called concurrent w/f and run concurrently.

13. Commit types –
There are 3 types of commits – source based, target based and user defined.

Target based commit –
1. Commit interval – it's the interval at which server commits data to relational target during a session.

2. Commit point depends upon buffer block size and commit interval. Server commits data based on the number of target rows and key constraints on target table.
3. during a session, server continues to fill writer buffer after it reaches the commit interval. When buffer block is full. Informatica server issues a commit command. As a result commit point generally exceeds the commit interval.
4. server commits data to each target based on pk-fk key.
   - Session => edit => properties
   A. commit type – target/source.
   B. commit interval – 10000
5. commit interval depends upon –
   A. commit interval
   B. writer wait timeout
   C. buffer blocks.
6. when you run a target based commit session, i.s. may issue commit on after or before the configured time interval. i.s. issues following process to issue commit –
   A. i.s. reaches commit interval, still continues to fill the writer buffer block, after this fills it issues commit command.
   B. if writer buffer block fills before commit interval i.s. starts to write on target. But it waits to issue commit, it issues commit when one of the following condition is true –

I. the writer idle for an amount of time specified by i.s. 'writer wait timeout' option.
II. i.s. reaches the commit interval and fills another buffer.

Source based commit –

I. configured as per (iv) above.
II. commit point is the commit interval. Server commits data based on number of source rows, from active sources in a single pipeline. A pipeline consists of source qualifier and all transformations except active transformation viz filter, router etc. and target.
III. when server runs a session, it identifies active rows for each pipeline in the mapping. The server generates a commit row from active source at every commit interval and when target receives it perform commit.

User Defined Commit –
i.s. commits data based on transactions defined in mapping properties. you can also commit and rollback options in sessions.

14. Tracing Level Types –
A. Normal
B. Terse
C. Verbose Initilization
D. Verbose Data

15. Error Handling –
A. **Error Handling –** when you run a session following errors can come –
I. fatal exception – stops running workflow. Eg. Db connection error.
II. non-fatal exception – error that can be ignored by Informatica power center and cause records to drop out from target table, otherwise handled in etl logic. Eg. Pk violation, data conversion error etc.
III. user defined exception – data issues critical to data quality, which might get loaded to db unless explicitly checked for data quality.

User defined exception – we can handle this exception using-
I. error handling functions – error, abort.
II. user defined error tables.

Error () – this function causes i.s. to skip a row and issue an error message, which you define. The error message display either in session log or written to the error log table based on error logging type configuration in session. You can use error () in expression transformation to validate data. Generally, you use error() within an iif or decode function to set rules for skipping rows.

Eg. – IIF (trans_date > sysdate, error('trans_date invalid));

Abort () – stops the session and issues a specified error message to session log file or written to error log tables based on error logging type configuration in the session.

Eg. – IIF (ISNULL (creditcard_number), abort ('empty credit card'));

Non-Fatal Exception – helps in capturing errors not considered during design phase; can be handled using:-
I. default port value setting. Eg. Set default value for nulls.
II. row error logging.
III. error handling settings.

Fatal Exception – can be handled using –
Restartability – ability to restart processes at the step where it failed as well as ability to restart entire session.
W/f recovery – allows to continue processing workflow and w/f tasks from the point of interruption. During this process i.s. access w/f state, that is stored in memory or on disk based on recovery configuration. The w/f state of operation includes status of tasks in w/f and w/f variable values.

WorkFlow Recovery –
The configuration includes –
1. w/f configuration for recovery.
2. session and task configuration for recovery.
3. Recovering task from failure.

w/f configuration for recovery =>
workflow => edit => properties

I. Enable HA recovery -__
II. Automatic recovery terminated task - ___
III. Maximum automatic recovery attempts - ___

Enable Recovery – when you enable w/f for recovery, i.s. saves the w/f state of operation in a shared location. You can recover w/f if it's stops, terminates or aborts.

Note – An option HA license is required for this check box to be available for the selection. w/o HA option w/f must recover manually either in w/f monitor or using command line to recover workflow.

Suspend –
I. w/f => Edit => General => Suspend mail
II. ___ suspend on error.

When you configure a w/f to suspend on error the i.s. stores the w/f state of operation in memory. You can recover suspended w/f if a task fails. You can fix the task error and recover the w/f. if w/f is not able to recover automatically

from failure within maximum allowed number of attempts it goes to suspended state.

Session and Task Configuration for recovery –

Task => edit => Recovery Strategy

In the drop-down menu chose the recovery strategy, for each task it differs:-

1. Restart Task – available for all tasks.
2. Fail task and continue w/f – session and command tasks only.
3. Resume from task checkpoint – session task only.

When workflow recovers, it recover tasks and sessions on the basis of recovery strategy.

When you configure the session recovery strategy to 'resume from task checkpoint', i.s. stores the session state of operation in the shared location, $PMStorageDir and also it's written to the recovery tables to determine from where to begin loading data to target tables, in case of recovery.

Recovering w/f from failure –

I. Recover automatically – using HA option.
II. Recovering manually – using w/f manager or monitor.

Open w/f manager or monitor – right click and select:-

1. Recover Task. (task level)
2. Recover w/f from task. (task level)
3. Recover Workflow. (workflow level)

Unrecoverable tasks –

1. if session uses non-pass through partitioning.
2. if session uses database partitioning.
3. if you want to debug mapping, you can't perform recovery from debugger.
4. If you want to perform test data load. If you enable recovery, w/f manager disables test data load option.
5. any changes in mapping afterwards.

16. Target Load Plan –

if there are multiple targets then to define which is the load order of targets.

17. Constraint Based Loading –

if there is constraint based dependency between tables like pk-fk relationship then first table having pk will loaded then table having fk will be loaded.

18. Unit testing vs System testing –

unit testing => Integration testing => System testing.

19. Push Down Optimization –

to convert etl logic to the sql and fire it against the database to improve performance. PDO are of three types – 1. Source based 2. Target Based 3. Full PDO. Source based means fire sql against source. Target based means fire sql against target. A common use case for full pushdown optimization is when the source and target databases are in the same relational database management system. For example, configure full pushdown optimization to push all the transformation logic from a Teradata source to a Teradata target. When you run a session configured for full pushdown optimization, the Integration Service analyzes the mapping from the source to the target. When the Integration Service reaches a downstream transformation that it cannot push to the target database, it generates and executes SQL statements against the source or target based on the transformation logic that it can push to the database. Push Down Optimization Viewer is there to see which transformation will go to source/target.

20. Flat Files –

I. it can be: -
1. delimited flat-file
2. fixed-width flat file

II. Direct and Indirect flat-file: -

you need to declare following things in session, while working with flat-file –

1. Source file name – file name, being loaded to target.
2. Source file directory – directory name, where file is resided.
3. Source file type – direct/Indirect.

III. In the indirect loading file path name list is given in the file and this file path is given in the session, it's used to handle say 100 files in source side so it can be used:-

eg. > cat cust_list.dat

$PMSourceFileDir/cust_us.dat

$PMSourceFileDir/cust_uk.dat

$PMSourceFileDir/cust_india.dat

IV. Expression transformation can be used to generate flag and transaction control transformation to perform tc_commit_before, tc_commit_after and so on when to generate files in the target side.

V. whenever cobol file is placed Informatica add normalizer transformation automatically, If not add it.

VI. remove duplicates –

I. for relational tables – I) by sql override. II) in S.Q. select 'distinct' option.
II. for flat-files or other sources – I) sorter with distinct II) aggregator with group by funda.

21. Command Line Utility –

Informatica provides four built-in command line programs or utilities to interact with Informatica features. They are PMCMD, PMREP, INFACMD and INFASETUP. PM stands for Power Mart.

1. PMCMD is used to perform following tasks: -

I. Start task
II. Start w/f from specific task.
III. Stop, Abort w/f and sessions.
IV. Schedule the w/f.

2. Common Syntax for PMCMD –

I. PMCMD cmd_name –

service Informatica_integration_service –d domain_name –u user_name –p password –f folder_name –w workflow_name;

CMD_NAME =>

1. Scheduleworkflow
2. Startworkflow
3. Stopworkflow
4. Abortworkflow

II. Start w/f from specific task –
PMCMD starttask –service Informatica_integration_service –d domain_name –u user_name –p password –f folder_name –w workflow_name –startfrom task_name;

iii. Abort task –
PMCMD aborttask –service Informatica_integration_service –d domain_name –u user_name –p password –f folder_name –w workflow_name task_name;

3. PMCMD is program command utility to communicate with Informatica servers. PMCMD command is used to control Informatica repository events through unix, when Informatica server is on unix server.

Location of PMCMD –
<installed drive>\informatica\server\bin\pmcmd

Syntax –
PMCMD command_name [-option1] arg1 [-option2] arg2
Some of the commands used in Informatica –
Pmcmd> connect –sv service –d domain_name –u user_name –p password
Pmcmd> startworkflow/ getworkflowdetails/ gettaskdetails/ stoptask/ getsessionstatistics/ scheduleworkflow/ unscheduleworkflow -f folder_name workflow;

Pmcmd> disconnect –sv service –d domain_name –u user_name –p password

4. Pmcmd command with parameter file –
Pmcmd startworkflow –s $server_name –u $user_name –p $password –f $folder_name –paramfile param_file_name –wait $wkf

5. **to run pmcmd command, it's required to set following given environment variables –**
I. INFA_HOME – absolute path to dictionary of Informatica client.
II. INFA_DOMAINS_FILE – absolute path to the domains infa file.
III. INFA_USER – username to connect to Informatica.
IV. INFA_PASS – password to go with username in INFA_USER.

Tip – set environment variables inside your script so that you won't require each user to set-up it's environment variables to use script.

6. Connecting through pmcmd – (connecting to i.s.) –
C:\user\ramakantt> pmcmd [enter]
Pmcmd> connect [enter]
Username>danger [enter]
Usersecuritydomain: [enter]
Password: [enter]
Domainname: danger [enter]
Servicename: infm_integration [enter]
Connected to integration service: [infm_integration]
Pmcmd>

22. Different types of cache memory: -
1. persistent vs non-persistent cache – if after completion of session, cache stays/hold the data it's persistent else not.
2. Static vs dynamic cache – if cache changes during run time it's dynamic else it's static cache.
3. named vs un-named cache and shared vs non-shared cache – if the cache memory is designed for one look up in one mapping then it will be used by all other look ups in the same mapping and then it's un-named cache. But it's area of usage is limited for only one mapping. If you want to use it for another mapping, then click on the 'persistent cache' and make it as persistent and give the 'name to the cache' and call it to the other mapping from the session properties by using name. so it will be used across mappings. It's called as shared cache else it's called as non-shared cache.

23. Parameters and Variables –
1. to go to parameters/variables.
Mapping => parameters and variables
Set following values, that are as: -
I. name - $$<parameter_name>
II. type – parameter/variable
III. datatype
IV. iv)prescision
V. scale
VI. aggregation – max/min
VII. initial value – any value.

2. after declaration, you can use parameters/variables inside expression transformation.
3. how to make parameter file –
I. open a notepad and type (folder.workflow.session)
eg. – [folder_name.wf:workflow_name.st:session_name]
II. place parameter values inside of this and save with extension .prm;

4. I. edit the session and in properties tab give param file name with full path.
II. edit w/f and in properties tab also set param file name.
5. parameter – constant value; variable – changes between sessions.
6. I.S. looks values in following order –
I. value in param file => ii) value in pre-session variable assignment => iii) initial values defined in repo (for variables).

7. can be seen by => session, right click => view persistent value.

8. power center default values: -
Data type values
String empty string
Number 0
Date time 1/1/1

9. I.S. holds two values for a mapping during session run for variable => 1. Start value 2. Current value.
10. different types of variable functions –
Setvariable, setmaxvariable, setminvariable, setcountvariable;

11. Use of mapping variable –
I. to pass value from one session to another.
II. Incremental data load.
III. in lookup override where you want to use dynamic sql override.

12. $ - session variable; $$ - mapping variable; $$$ - system variable;

24. Designer tips and tricks –

1. Search tools – in tool bar.
2. Auto link ports by position and/or by name. (right click to workspace, layout - autolink)
3. propagating port attributes (right click on column and propagate it)
4. Dependencies – right click on tables and ___; mapping => Dependencies
5. Link path – to view forward/backward link paths from/to a particular port.
6. Overview window – view =>overview window.
7. Iconize workspace objects – iconize mapping objects.
8. Table definition options – (tools => options => table tab).
9. Copying designer objects – ctrlC+ctrlV
10. Compare objects.

25. Sessions –

Using session properties you can configure various session characteristics like pre and post sql scripts, log file name, path and memory properties.

Session => right click => edit => properties tab

1. I. treat source rows as –insert/update/delete/data driven
   II. commit interval
   III. Enable test load __; no. of rows set to test= 1;
   IV. Session log file name and session log file directory
   V. Log file options 6. Error handling
2. Mapping properties – give flexibility to fine tune memory allocated to Informatica for performance optimization in 'config object' tab.
3. mapping and source/target properties – i) set connections. ii) place table suffix and table name.
4. session => success or failure of session task – (in general tab).

I. fail parent if task fails
II. fail parent if tast does not run.
III. disable this task.

26. shared folder –

Create shared folder in Informatica –

1. if a new folder being designed create => new folder in repo manager, and then right click on it and place 'IS SHARED' as yes.
2. if folder is already existing, then right click on it and set properties. 'IS SHARED' is just a property to show it's shared. Enable 'allow shortcut ___', to make folder as shared.

Folder creation in repo manager –

Folder => create => set properties => name, description, owner, group, permission.

Note – import and export can be done by xml files.

27. Version Control –

It's a paid utility. It works upon three things

I. check in
II. check out
III. undo checkout we can also see versioning history.

28. Partitioning –

We can create/define up to 64 partitions at any point in a pipeline. By partitioning we can make parallel reader writer and transformation threads that is more than 1. This increase the performance by making the parallelism. There are two types of partitioning first is static and second is dynamic partitioning.

Type of static partitioning –

1. Hash Key Partitioning 2. Round Robin Partitioning 3. Pass Through Partitioning 4. Key Range Partitioning.

Type of dynamic partitioning –

1. Based on source partitioning
2. Based on number of CPUs.
3. Based on number of nodes in grid
4. Based on number of partitions (defined in the session parameter $dynamicpartitioncount >1)

Note –

1. dynamic partition can be used if source qualifier not include the source query or source filter.
2. dynamic partition can't be used with xml source or target.
3. dynamic partition can't be used for session where manual partition has been done else session will fail.
4. dynamic partition uses same connection for each partition.

Note – instead of debugger we can use the verbose data as tracing level.

29. Incremental aggregation –

In this incremental aggregation is performed. For example, we want to have load of data as 10,20,30 so the aggregate value will be 10+20+30/3 = 20, it will be stored in the cache memory now the next data has come which is 30 so now in the incremental aggregation it would be 20+30/2 = 25 so now it's 25 so on.

Note –

1. don't delete cache files or change in the no. of partitions.
2. if you are using percentile or median function, don't use incremental aggregation because i.s. uses system memory to process these functions in addition to cache memory configured in session property.

30. Aggregate cache in aggregator transformation –

Power center server stores data in the aggregate cache until it completes aggregate calculations.

Note – incremental aggregation make session invalid, if mapping include aggregator with transaction control transformation.

Reinitilize aggregate cache – overwrite whole cache memory once again, till this is done so it's required to disable it after data loading.

31. How to use persistent cache –

Check following options in lkp –

1. look up cache persistent - ____
2. cache file name prefix- _____ (write the name)
3. recache from look-up source - _____ (to rebuild cache)

[after session first time run, uncheck 3rd option so that table can be used after build of cache from session to session.]

32. Index and Data cache –
1. these caches are stored in $PMCacheDir with *.idx and *.dat format respectively. After successful execution of session Informatica server deletes these cache files. It's used in aggregator, rank, joiner and lkp transformations.

33. Worklet variables –
1. pre variable worklet assignment 2. Post variable worklet assignment

34. There are different types of transformations –
Aggregator, Expression, filter, router, joiner, look up, rank, sorter, update strategy, sequence generator, source qualifier, union, transaction control, sql and normalizer transformation

35. There are different types of tasks –
Command, session, email, event raise, event wait, control, link, decision, timer, assignment task.

Note –
1. Joiner transformation is known as blocking transformation.

36. Performance Tuning –
To see bottlenecks below things can be used –
I. session thread statistics.
II. session performance counter.
III. workflow monitor properties.

I. Session thread statistics – there are three types of threads – Reader, Writer and Transformation thread.

Gathering thread statistics => we can gather three informations of it;

I. Run Time ii) Idle Time iii) Busy time = [(Runtime-idletime)/runtime] * 100 iv) thread worktime
II. Gathering performance counters –

For this tick on the session properties –
1. Collect performance data.
2. Write data to repository____.

Understanding performance counter –
w/f monitor => session => run properties.
a non-zero count for read-from-disk and write-to-disk indicate sub optimal settings – for transformation index or data caches. This may indicate to tune session performance cache manually.

A non-zero count for error rows:- indicates you should eliminate transformation error to increase performance.

Error rows – remove error increase performance.

Read from disk and write to disk => if not equals to zero, increase cache size to increase performance.

Session bottleneck using session log file – when i.s. initializes a session, it allocates a block of memory to hold source or target data. Not having enough buffer memory for dtm process, slow down etl process and causes large fluctuations in performance. If not so we get warning; then work upon it.

In Informatica at following level performance tuning can be done –
1. source level
2. target level
3. mapping level
4. session level
5. system level

Session bottleneck using workflow monitor –
Session => run properties => partition details

I. cpu%
II. memory usage
III. swap usage.

System level –
This bottleneck can be seen by i) cpu ii) memory usage iii) swap usage.

Session bottleneck –
Due to small cache size, small buffer memory and small commit intervals can cause session bottleneck.

To remove this problem –
I. use terse mode in tracing level.
II. disable high precision if not required.
III. use pipeline partitioning.
IV. increase commit interval.
V. use bulk loading, external loading etc.
VI. run sessions concurrently, if they are independent from each other.

Db optimization –
I. optimize target db.
II. increase n/w packet size.
III. if source/targets are flat files it should present in the system where Informatica server is resided.

Target bottleneck –
when much time is consumed in writing target db. You can identify target bottlenecks by configuring the session to write a flat-file target. If performance increases it's target bottleneck.

To increase performance –
I. drop index and key constraints.
II. increase checkpoint interval.
III. use bulk load, external load etc.
IV. increase db n/w packet size.
V. optimize target db.

Source bottleneck –
Identifying source bottlenecks –
I. using filter transformations.
II. Read test session.
III. by extracting query and run against db and check for all.

To improve performance -
I. optimize query using optimizer hint.
II. use conditional filters, at source db itself.
III. increase db n/w packet size.
IV. configure source db torun parallel queries.
V. connect to oracle db using ip protocol.
VI. proper indexing for group by or order by clause.

At mapping level –

I. reduce transformations and delete un-necessary links.
II. for transformations that uses data cache eg. Joiner, aggregator limit connected i/o ports.
III. consider single pass reading. (for one source and multiple mappings).
IV. optimize sql overrides to raise performance.
V. increase partitioning in session.
VI. optimize the expressions, joiner, aggregators, look ups etc. for example
VII. sort data before sending it to the transformations.
VIII. reduce number of i/o nodes. And so on.

37. References -

[1] Informatica-10.1-developer tool guide https://kb.informatica.com/proddocs/Product%20Documentation/5/IN_101_DeveloperToolGuide_en.pdf

[2] Informatica-10.1- developer transformation guide https://kb.informatica.com/proddocs/Product%20Documentation/5/IN_101_DeveloperTransformationGuide_en.pdf

[3] Informatica-10.1-developer mapping guide https://kb.informatica.com/proddocs/Product%20Documentation/5/IN_101_DeveloperMappingGuide_en.pdf

[4] Informatica power center-10.1-transformation guide https://kb.informatica.com/proddocs/Product%20Documentation/5/PC_101_TransformationGuide_en.pdf

[5] Informatica power center-10.1-designer guide https://kb.informatica.com/proddocs/Product%20Documentation/5/PC_101_DesignerGuide_en.pdf

[6] Informatica power center-10.1-developer workflow guide https://kb.informatica.com/proddocs/Product%20Documentation/5/IN_101_DeveloperWorkflowGuide_en.pdf

[7] Informatica power center-10.1-getting started https://kb.informatica.com/proddocs/Product%20Documentation/5/PC_101_GettingStarted_en.pdf

[8] Informatica ETL: A Beginner's Guide To Understanding ETL Using Informatica PowerCenter https://www.edureka.co/blog/informatica-etl/

[9] **Informatica Tutorials & Interview Questions** https://tekslate.com/tutorials/informatica/

[10] **INFORMATICA TUTORIAL: Complete Online Training** https://www.guru99.com/informatica-interview-questions.html

[11] Informatica PowerCenter (Version 10.1) https://kb.informatica.com/proddocs/Product%20Documentation/5/PC_101_DesignerGuide_en.pdf