PMT Architecture High Level Understanding Document

PMT – Performance Management Tool is written in perl to collect, transform and load performance, configuration, CDRs, EDRs data from network. Then the data is used for reporting purpose by other systems.

This document focuses on giving a highlevel idea on how PMT Framework is built and how it works.

In PMT to refer to a particular type of data we say flow/plugin or tag. In this document we will be using name flow.

For example:- a particular type of CDRs, like TAP CDRs which stores roaming cdrs of customers outside the countries – this is categorized under flow name MRM\_CDR.

PMT is an ETL tool with purpose of Collecting, parsing and loading the data.

In this framework above functions of collecting, parsing and loading the data are termed as roles and these roles reside under each flow, of course depending on nature of the flow a role can be in it or might not. For example:- for a csv data, there need not be a parsing role.

Roles can also be subdivided further into groups. Like a role “parse\_load” will have two groups – parse and load. Groups will have individual steps that will list out what needs to be done by the processes. All these information are stored in job definition xml in an xml database – dbxml. [ see later for job definition structure and related activities ]

We can separate out parse\_load role into two different roles itself one parse and another load. A role “load” then can have two groups within it – one is “load’ itself and another is afterloader.

When setting up PMTFramework .bash\_profile must be setup with following, this is used by the framework to locate important paths and files.

PATH=$PATH:$HOME/bin

PMT\_ROOT\_DIR="/opt/PMT"

PMT\_INSTALL\_DIR="$PMT\_ROOT\_DIR/product"

export PMT\_ROOT\_DIR PMT\_INSTALL\_DIR

PATH=/usr/bin:$PATH

PATH=$PATH:$PMT\_INSTALL\_DIR/util:/var/opt/oracle:/opt/oracle/product/11.2.0.3/bin

PATH=/opt/PMT/binary/perl/bin:$PATH

PATH=/opt/PMT/pmtsys/bin:/opt/PMT/pmtsys/scripts:/opt/PMT/bin:$PATH

export PATH

ORACLE\_SID=PMT01D ; export ORACLE\_SID

ORACLE\_HOME=/opt/oracle/product/11.2.0.3 ; export ORACLE\_HOME

ORACLE\_BASE=/opt/oracle ; export ORACLE\_BASE

NLS\_LANG=American\_america.WE8ISO8859P1 ; export NLS\_LANG

NLS\_DATE\_FORMAT="DD-MON-YY HH24:MI:SS" ; export NLS\_DATE\_FORMAT

export PMTROOT=/opt/PMT

export PMTCONFIG=/opt/PMT/etc/pmt.conf

export TNS\_ADMIN=${ORACLE\_HOME}/network/admin

export DB\_HOME=/opt/PMT/var/db/

Terminology used with roles

Collection :

When collecting files it is always important to track history of files collected so that same files are not collected every time. This history is maintained in a flat file which is defined in the configuration file of each flow separately.

This configuration file can be found under /opt/PMT/etc, with name as <FLOW\_NAME>.conf

* + - Minage : This variable defines the time (creation time ) that a file should have already before it can be picked up by processes executing role collection. This can be defined in flow specific configuration file with values as 1m ( meaning files with creation time more than 1 minute will be valid for collection ), similarly 1h meaning 1 hour, 1d meaning 1 day.
    - Maxage : This variable defines the time (creation time ) that a file should have after which it becomes invalid for collection. For example :- we don’t want to collect files older than 1 day so we specify 1d or we donot want to collect files older than 30 minutes so we specify 30m.
    - source.location : location on remote server where files can be found for collection.

It has to be defined in a particular way only.

<protocol>://username@server:<path of folder>/

For example :- sftp://mybook@localhost:/home/mybook/

* + - source.file\_pattern : specifies the pattern of the files which the process should look for and only download those.

For example :- a folder can have many different files, and can have files for different dates, and the process needs to collect files for today’s date only so the entry in this variable should be,

A.\*{{\_NOW\_|formatDateTime %Y%m%d}}.\*.xml

\_NOW\_ 🡪 this is functions defined in PMT framework which can be used to calculate time dynamically. Similarly there are other functions which can be used in the configuration files. See below for more.

{{}} 🡪 curly braces means that this part will be dynamically calculated by the process. [ To calculate this special function expand is used defined in the Initial context used by the process. What is initialContext and the function see below]

* + - filter.class

This variables stores the module which will perform the filter of the files based on source.file\_pattern value.

* + - history\_file

Stores the history file name and path.

* + - target.directory

Directory where all the files are to be collected to on the local server.

* + - remove\_source

This variable tells the process if the collection process can delete the files on the remote server or not after collection of the files.

0 means it will not delete

1 means it will delete.

Parsing/Registering:

* + infile\_dir : this variables stores the path where the collection has been done.
  + archive\_dir : this variables stores the path where after parsing the files are to be moved.
  + archive\_error\_dir : this variables stores the path where the files are to be moved when there has been some problem in parsing.
  + ora\_datafile\_dir : this variables stores the path where ora files for sqlldr to load is generated by parser.
  + parse.<FLOW\_NAME>.concurrency : defines how many parser are to run in order to perform the parsing task.

Load:

<flow\_name>.sqlldr\_services : this variables stores the name which is to be used by loader process to pick up the database configuration and control file or par file configuration.

The configuration used by framework to run is stored in pmt.conf which is hardcoded in the scripts hence it has to be pmt.conf always.

Here are the listing of the pmt.conf

[db] 🡪 stores database information , individual flows can also have this in their personal configuration file but the name of each variable inside this block has to start with flow\_name.

<flow\_name>.ora\_username

<flow\_name>.ora\_password

<flow\_name>.ora\_sid

<flow\_name>.driver

[ssh] 🡪 stores the username and password used by the collection process used to connect to remote servers. It supports both private key authentication and username/password authentication for sftp.

To define an entry inside this block.

username@server=PRIVKEY:<private key fullpath with name>

username@server=PASSWD:<passworx>

For the first time if connecting to the remote server using sftp, it has to be done manually as it requires the server to be added to know host list.

[monitoring] 🡪 stores the connection details to the database where the tables t\_app\_monitoring and t\_app\_messaging is defined. These tables are used by the process to insert successful or failed job details. Details include number of records read and loaded in one session. Start and end time of one run.

If these details are incorrect no process in this framework will start.

db.username = PMT\_APP

db.password = APPPMTT

db.service =TDREPT

db.driver = oracle

[dbxml] 🡪 this is the xml database which is used to hold the job definition of each flow.

By default following are used.

envdir=/opt/PMT/var/db

db:pmtlog=pmt\_log.dbxml

db:pmtconf=pmt\_config.dbxml

xquerylib=file:///opt/PMT/pmtsys/lib/xquery/

[ftp] 🡪 This block stores the username and password for each remote server where the protocol used will be ftp.

Format as follows,

username@server=PASSWD:<password>

[logger] 🡪 This block stores the logging information used by the framework. Default log creation paths are

/opt/PMT/logs 🡪 This contains the logs created by running a a role of a flow , it also contains sqlldr load logs.

/opt/PMT/tmp 🡪 This directory contains the standard error logs of child processes ( PMTPLHelper )

When in case of any error both logs are to be checked for any entry.

\_default\_.logformat=normal [ fixed, should not be changed ]

\_default\_.logthreshold=DEBUG [ can be changed to INFO, trace ]

\_default\_.default\_loglevel=DEBUG [ can be changed to INFo, trace ]

\_default\_.default\_logdomain=SYSTEM [ fixed, should not be changed ]

logthreshold = {{CMDPARAM/logthreshold|default {{CONFIG/logger/{{SYSTEM/RUN/FLOWCD}}.logthreshold | default {{CONFIG/logger/\_default\_.logthreshold}}}}}} 🡪 *calculated at runtime. First the framework picks up this entry and finds out where to look for actual value. A flow can have its separate logging information. This line times the framework to first look for command line parameters supplied with the run command if nothing found then look for the flow specific variable definition if this is also not present then only it looks for \_\_default\_\_.logthreshold. When nowhere a value can be found a default keyword can be used to be used as last option.*

logformat = {{CMDPARAM/logformat|default {{CONFIG/logger/{{SYSTEM/RUN/FLOWCD}}.logformat | default {{CONFIG/logger/\_default\_.logformat}}}}}}

default\_logdomain = {{CMDPARAM/default\_logdomain|default {{CONFIG/logger/{{SYSTEM/RUN/FLOWCD}}.default\_logdomain | default {{CONFIG/logger/\_default\_.default\_logdomain}}}}}}

default\_loglevel = {{CMDPARAM/default\_loglevel|default {{CONFIG/logger/{{SYSTEM/RUN/FLOWCD}}.default\_loglevel | default {{CONFIG/logger/\_default\_.default\_loglevel}}}}}}

compress=Y

filename={{CMDPARAM/logfile | default {{SYSTEM/RUN/FLOWCD}}.{{SYSTEM/RUN/ROLE}}.{{\_PROCESSID\_}}.{{SYSTEM/RUN/RUNSEQ}}.{{\_NOW\_|formatDateTime %Y%m%d%H%M%S}}}}

logdir={{CMDPARAM/logdir|default {{ENV/PMTROOT}}/logs}} 🡪 *this variable defines the log directory*

[settings] 🡪 used by framework to find the actual values. These are mandatory.

load.queue\_batchsize={{CMDPARAM/load.queue\_batchsize| default {{CONFIG/settings/load.{{SYSTEM/RUN/FLOWCD}}.queue\_batchsize| default {{CONFIG/settings/\_default\_.load.queue\_batchsize}}}}}}

parse.concurrency={{CMDPARAM/parse.concurrency| default {{CONFIG/settings/parse.{{SYSTEM/RUN/FLOWCD}}.concurrency| default {{CONFIG/settings/\_default\_.parse.concurrency}}}}}}

load.concurrency={{CMDPARAM/load.concurrency| default {{CONFIG/settings/load.{{SYSTEM/RUN/FLOWCD}}.concurrency| default {{CONFIG/settings/\_default\_.load.concurrency}}}}}}

Similarly there is

[parameters] block.

infile\_dir = {{ CMDPARAM/infile\_dir|default {{CONFIG/parameters/{{SYSTEM/RUN/FLOWCD}}.infile\_dir|default {{CONFIG/parameters/\_default\_.infile\_dir}}}}}}

infile\_pattern = {{ CMDPARAM/infile\_pattern|default {{CONFIG/parameters/{{SYSTEM/RUN/FLOWCD}}.infile\_pattern|default {{CONFIG/parameters/\_default\_.infile\_pattern}}}}}}

archive\_dir = {{ CMDPARAM/archive\_dir|default {{CONFIG/parameters/{{SYSTEM/RUN/FLOWCD}}.archive\_dir|default {{CONFIG/parameters/\_default\_.archive\_dir}}}}}}

archive\_error\_dir = {{ CMDPARAM/archive\_error\_dir|default {{CONFIG/parameters/{{SYSTEM/RUN/FLOWCD}}.archive\_error\_dir|default {{CONFIG/parameters/\_default\_.archive\_error\_dir}}}}}}

ora\_datafile\_dir = {{ CMDPARAM/ora\_datafile\_dir|default {{CONFIG/parameters/{{SYSTEM/RUN/FLOWCD}}.ora\_datafile\_dir|default {{CONFIG/parameters/\_default\_.ora\_datafile\_dir}}}}}}

To Inovke any role in a flow script PMTRunner is to be used.

This script identifies the role and flow and then starts the necessary processes to complete the action asked for.

The syntax to invoke the script is

PMTRunner –flowcd <flowname> --role <rolename>

This script is present at location

/opt/PMT/pmtsys/scripts/ and is considered as CORE part of the framework.

CORE part of the framework consist of all the modules which are required to run the framework. Following are the paths to core modules,

/opt/PMT/pmtsys/lib/pmt

/opt/PMT/pmtsys/scripts