1

Initialization and Configuration Parameters

This chapter contains detailed descriptions of the DBAip High Load Capture initialization and configuration parameters and includes the following topics:

Uses of Initialization Parameters

Initialization parameters fall into various functional groups. For example, parameters perform the following functions:

- Configure SNMP parameters
- Configure management parameters of the capture log files

Users can use configuration parameters to perform following functions (for example):

- Create new capture connections
- Specify names of the capture log files
- Created dependencies between triggering and capture connections

Most of the parameters are optional. High Load Capture will provide notification to the user when mandatory parameter is missing in configuration or parameter file.

Parameter Files

A parameter file is a file that contains a list of initialization/configuration parameters and a value for each parameter. You specify parameters in a parameter file that reflect your particular configuration. High Load Capture supports the following two types of parameter files:

- Master Parameter Files
- Capture Connection Configuration Files

Master Configuration Files

A Master Parameter File is a text file that contains generic parameters related to the overall functionality of current High Load Capture installation.

Master configuration file is set by using "**-parm=<conf_file_name>**" HLC command line option.

Capture Connection Configuration Files

A Capture Connection Configuration File is a text file that contains list of data capture connections and corresponding connection configuration parameters.

Capture configuration file is set by using "**-conf=<conf_file_name>**" HLC command line option.

Changing Parameter Values

You can change the value of a parameter by editing the initialization parameter file. In most cases, the new value takes effect the next time you restart an HLC application. However, you can change the value of some parameters for the currently running application as described in the following section.

Dynamic Parameters

Some initialization parameters are **dynamic**, that is, they can be modified by altering their values in configuration file and sending USR2 signal to the running HLC application process.

Reading the Parameter Descriptions

The parameter descriptions in this chapter adhere to the following format.

PARAMETER_NAME

Parameter type	Specifies the type of the parameter:
Master Group	Specifies the group name, that parameter belongs to
Syntax	Specifies the valid syntax for specifying the parameter.
Default value	Specifies the value this parameter assumes if not explicitly specified.
Parameter Class	Specifies whether the parameter is dynamic or static. If dynamic, its value can be changed by altering connection configuration file and sending USR2 signal to the active HLC process

For each parameter, following paragraphs further describes the parameter and the effects of different settings.

Master Configuration File Parameter Descriptions

Descriptions of the individual parameters follow in alphabetical order.

LOG_ARCHIVE_GENERATIONS

Parameter type	Integer
Mater Group	None
Syntax	‘log_archive_generations’ => ‘<value>’
Default value	3
Parameter Class	Static

Use LOG_ARCHIVE_GENERATIONS to set maximum size of the log generations to be kept

LOG_ARCHIVE_SIZE

Parameter type	Integer
Mater Group	None
Syntax	‘log_archive_size’ => ‘<value>’
Default value	5242880
Parameter Class	Static

Use LOG_ARCHIVE_SIZE to set maximum size of the capture log file

SNMP_ACTIVE

Parameter type	String
Mater Group	None
Syntax	‘snmp_active’ => ‘Y|N’
Default value	‘N’
Parameter Class	Static

Use SNMP_ACTIVE to enable SNMP traps sending

SNMP_TRAP_HOST

Parameter type	String
Mater Group	None
Syntax	‘snmp_trap_host’ => ‘<name|ip>’
Default value	localhost
Parameter Class	Static

Use SNMP_TRAP_HOST to set hostname or ip address of the SNMP trap server

SNMP_TRAP_PORT

Parameter type	Integer
Mater Group	None
Syntax	‘snmp_trap_port’ => ‘<port>’
Default value	162
Parameter Class	Static

Use SNMP_TRAP_PORT to set port of the SNMP trap server

SNMP_TRAP_COMMUNITY

Parameter type	String
Mater Group	None
Syntax	‘snmp_trap_community’ => ‘<community>’
Default value	public
Parameter Class	Static

Use SNMP_TRAP_PORT to set SNMP trap community

SNMP_TRAP_OID

Parameter type	String
Mater Group	None
Syntax	‘snmp_trap_oid’ => ‘<community>’
Default value	No default value
Parameter Class	Static

Use SNMP_TRAP_OID to set oid for the SNMP traps

SNMP_TRAP_PORT

Parameter type	Integer
Mater Group	None
Syntax	‘snmp_trap_port’ => ‘<port>’
Default value	162
Parameter Class	Static

Use SNMP_TRAP_PORT to set port of the SNMP trap server

Sample Master Configuration File

```
#
# general master configuration file
#
$VAR1 = [
        {
          'snmp_active'           => 'y'
         ,'snmp_trap_host'         => '192.168.1.140'
         ,'snmp_trap_port'         => '162'
         ,'snmp_trap_community'    => 'public'
         ,'snmp_trap_oid'          => '.1.3.6.1.4.1.24216'
         ,'log_archive_size'       => '5242880'
         ,'log_archive_generations' => '3'
        },
      ];
```

Capture Connection Configuration File Parameter Descriptions

Descriptions of the individual parameters follow in alphabetical order.

ACTIVE_FLAG/CONNECTION

Parameter type	String
Mater Group	CONNECTION
Syntax	‘active_flag’ => ‘Y|N’
Default value	‘Y’
Parameter Class	Dynamic

Use ACTIVE_FLAG to enable data capture connection

ACTIVATE_ON_STARTUP/CONNECTION

Parameter type	String
Mater Group	CONNECTION
Syntax	‘activate_on_startup’ => ‘Y|N’
Default value	‘N’
Parameter Class	Dynamic

Use ACTIVATE_ON_STARTUP to begin data collection immediately upon startup of HLC

ALERT_CAPTURE_CYCLE/CONNECTION

Parameter type	Double
Mater Group	CONNECTION
Syntax	‘alert_capture_cycle’ => ‘<cycle>
Default value	No default value
Parameter Class	Dynamic

Use ALERT_CAPTURE_CYCLE to set data collection cycle after alter event (caused by ALERT_THRESHOLD_ABS or ALERT_THRESHOLD_PCT parameter settings)

ALERT_CAPTURE_PERIOD/CONNECTION

Parameter type	Double
Mater Group	CONNECTION
Syntax	‘alert_capture_period’ => ‘<period>
Default value	No default value
Parameter Class	Dynamic

Use ALERT_CAPTURE_PERIOD to set period of data collection after alter event (caused by ALERT_THRESHOLD_ABS or ALERT_THRESHOLD_PCT parameter settings). During ALERT_CAPTURE_PERIOD data collection cycle is set to ALERT_CAPTURE_CYCLE, even if alert condition is no more active.

ALERT_SNMP_TRAP/CONNECTION

Parameter type	String
Mater Group	CONNECTION
Syntax	‘alert_snmp_trap’ => ‘<Y|N>
Default value	N
Parameter Class	Dynamic

Use ALERT_SNMP_TRAP to send SNMP trap upon alert event

ALERT_THRESHOLD_ABS/CONNECTION

Parameter type	Double
Mater Group	CONNECTION
Syntax	‘alert_threshold_abs’ => ‘<period>
Default value	No default value
Parameter Class	Dynamic

Use ALERT_THRESHOLD_ABS to set absolute alert threshold to evaluate every data item collected by capture connection. Upon alert event, special records recorded into capture log file and data collection period if altered from CYCLE to ALERT_CAPTURE_CYCLE value

ALERT_THRESHOLD_PCT/CONNECTION

Parameter type	Double
Mater Group	CONNECTION
Syntax	‘alert_threshold_pct’ => ‘<period>
Default value	No default value
Parameter Class	Dynamic

Use ALERT_THRESHOLD_PCT to set percent alert threshold to evaluate every data item collected by capture connection. Upon alert event, special records recorded into capture log file and data collection period if altered from CYCLE to ALERT_CAPTURE_CYCLE value

CAPTURE_GROUP/CONNECTION

Parameter type	Integer
Mater Group	CONNECTION
Syntax	‘capture_group’ => [‘<cap_conn_cs_list>,]
Default value	No Default Value
Parameter Class	Dynamic

Use CAPTURE_GROUP to set list of capture connections (comma separated list) that can be controlled by triggering connection. If CAPTURE_GROUP is not set, triggering connection is controlling all capture connections.

CONNECTION

Parameter type	Group
Mater Group	None
Syntax	‘connection’ => {…}
Default value	No default value
Parameter Class	Dynamic

Use CONNECTION parameter to define a list of capture connection configuration parameters

CONNECTION_TYPE/CAPTURE

Parameter type	String
Mater Group	CAPTURE
Syntax	‘connection_type’ => ‘<c_type>’
Default value	No default value
Valid Values	capture, trigger
Parameter Class	Static

Use CONNECTION_TYPE parameter to define a type of the data collecting connection. Valid values are “**capture**” which defines connection that only collects data from the data source and “**trigger**”, which defines connection, that in addition to pure data collection is also capable of controlling other capture connections.

CHARTING_CONCENTRATOR_INDEX/CONNECTION

Parameter type	Integer
Mater Group	CONNECTION
Syntax	‘charting_concentrator_index’ => ‘<index>
Default value	No default value
Parameter Class	Dynamic

Use CHARTING_CONCENTRATOR_INDEX to set index of the data item in the row of collected data, that would be used by HLC GUI to group other columns.

For example, in capture connection, that collectes absolute values of the SQL “select EVENT, count(*), sum(WAIT_TIME) from v$sesison_wait group by event”, to use “EVENT” (index 1) as charting concentrator, parameter needs to be set as: 'charting_concentrator_index' => '1'

CYCLE/CONNECTION

Parameter type	Double
Mater Group	CONNECTION
Syntax	‘cycle’ => ‘<capture_cycle >’
Default value	No default value
Parameter Class	Dynamic

Use CYCLE to set length of the data capture cycle of capture connection

DATABASE/PARAMS

Parameter type	Group
Mater Group	PARAMS
Syntax	‘database’ => ‘’ ‘database’ => ‘usr/pwd@tns’
Default value	No default value
Parameter Class	Static

Use DATABASE to define a connection string to be used by High Load Capture to connect to the database. Empty string instructs HLC to attempt to connect as SYSDBA; otherwise HLC would connect to database using provided username, password and database tnsnames alias.

DATA_SOURCE

Parameter type	Group
Mater Group	Connection
Syntax	‘data_source’ => {…}
Default value	No default value
Parameter Class	Dynamic

Use DATA_SOURCE parameter to define a list of data source configuration parameters

DELTA_FLAG/CONNECTION

Parameter type	Integer
Mater Group	CONNECTION
Syntax	‘delta_flag’ => ‘<index>
Default value	-1
Parameter Class	Dynamic

Use DELTA_FLAG to set index of the data item in the row of collected data that would be used as a “join” column in delta collection mode to subtract results of current data sample from the results of the previous data sample. If DELTA_FLAG is set to -1 or not used, data collection is done in a absolute value mode

FILE/CONNECTION

Parameter type	String
Mater Group	CONNECTION
Syntax	‘file’ => ‘<log_file_name>’
Default value	No default value
Parameter Class	Dynamic

Use FILE to define name of the connection capture log file

FILTER_THRESHOLD_ABS/CONNECTION

Parameter type	Double
Mater Group	CONNECTION
Syntax	‘filter_threshold_abs’ => ‘<period>
Default value	No default value
Parameter Class	Dynamic

Use FILTER_THRESHOLD_ABS to set absolute filter threshold to evaluate every data item collected by capture connection and filter all rows where every data item is lower then parameter value.

NAME/CONNECTION

Parameter type	String
Mater Group	None
Syntax	‘name’ => ‘<name>’
Default value	No default value
Parameter Class	Static

Use NAME parameter to define a name of the capture connection

NAME/PARAMS

Parameter type	Group
Mater Group	PARAMS
Syntax	‘name’ => ‘<external_name>’
Default value	No default value
Parameter Class	Static

Use NAME to define name of the external script of program to be executed by High Load Capture

NAME/DATA_SOURCE

Parameter type	String
Mater Group	DATA_SOURCE
Syntax	‘name’ => ‘<ds_type>’
Default value	No default value
Valid Values	database, external
Parameter Class	Static

Use NAME parameter to define a type of the data source. Valid values are “**database**” which defined data source as a data retrieved from the database and “**external**” as a data retrieved from the external script or program

PARAMS/DATA_SOURCE

Parameter type	Group
Mater Group	DATA_SOURCE
Syntax	‘params’ => {…}
Default value	No default value
Parameter Class	Static

Use PARAMS to define a list of data source parameters bases on a data source type.

SQL/PARAMS

Parameter type	Group
Mater Group	PARAMS
Syntax	‘sql’ => ‘<sql>’
Default value	No default value
Parameter Class	Static

Use SQL to define a SQL query to be executed by High Load Capture

Sample Capture Connection Configuration File

```
#
# Sample capture configuration file
#
$VAR1 = [
        {
         'connection' => {
           'data_source' => {
           'params' => {
                'database' => '',
                'sql' => 'select count(*) CNT, \'SESS\' SESS from v$session where status = \'ACTIVE\'' },
            'name'   => 'database'
           },
           'file' => '/export/home/oracle/capture/act_sess.log',
           'connection_type' => {
             'name' => 'triggering',
           },
           'capture_group' => [ 'UPTIME_LOAD' ],
           'cycle' => 5,
           'alert_threshold_abs' => 2,
           'activate_on_startup'  => 'y',
           'active_flag' => 'y'
        },
         'name' => 'ACTIVE_SESSIONS'
        },
        {
         'connection' => {
            'data_source' => {
              'params' => { 'name' => './uptime.ksh' }
              ,'name' => 'external'
            }
            ,'file' => '/export/home/oracle/capture/uptime.log'
            ,'connection_type' => { 'name' => 'capture' }
            ,'cycle' => 1
            ,'alert_capture_cycle' => 0.5
            ,'activate_on_startup' => 'y'
            ,'active_flag' => 'y',
            ,'charting_concentrator_index' => 1
         },
         'name' => 'UPTIME_LOAD'
        },
];
```