

UCMDB Self-Monitoring

Universal CMDB

November 2017

Contents

Overview	3
UCMDB Server	3
UCMDB Database	8
Universal Discovery	8
Store and Forward Server.....	11

Overview

This document lists items that you can monitor in order to understand whether UCMDB is working correctly and to help flag any anomalies that warrant further investigation.

UCMDB Server

- Monitor the UCMDB Server service
 - **Windows:** Monitor the **UCMDB_Server** service to make sure that it is running.
 - **Linux:** Search for the **UCMDBServer** process (the path to the process will resemble **opt/hp/UCMDBServer/bin/wrapper**).
 - Monitor **wrapper.exe**. This is a "watchdog" process that monitors the UCMDB_Server process.
 - Wrapper logs for UCMDB Server are saved to **hp\UCMDB\UCMDBServer\runtime\log\wrapper.log**.
 - Log entries for starting UCMDB resemble the following:


```
STATUS | wrapper | 2014/07/30 10:47:51.692 | Launching a JVM...

INFO | jvm 1 | 2014/07/30 10:47:52.896 | -XX:+CMSClassUnloadingEnabled -
XX:+CMSIncrementalMode -XX:+HeapDumpOnOutOfMemoryError -
XX:InitialHeapSize=1073741824 -XX:+ManagementServer -XX:MaxGCPauseMillis=250 -
XX:MaxHeapSize=4294967296 -XX:MaxNewSize=1073741824 -XX:MaxPermSize=201326592 -
XX:MaxTenuringThreshold=6 -XX:OldPLABSize=16 -XX:-OmitStackTraceInFastThrow -
XX:+PrintCommandLineFlags -XX:+PrintGC -XX:+PrintGCDetails -
XX:+UseAdaptiveGCBoundary -XX:+UseCompressedOops -XX:+UseConcMarkSweepGC -XX:-
UseLargePagesIndividualAllocation -XX:+UseParNewGC

INFO | wrapper | 2014/07/30 10:48:06.658 | Waiting to start...

INFO | jvm 1 | 2014/07/30 10:48:09.629 | WrapperManager: Initializing...

INFO | jvm 1 | 2014/07/30 10:48:11.351 | 2014-07-30 10:48:11,336
[WrapperSimpleAppMain] INFO - Starting cmdb server

STATUS | wrapper | 2014/07/30 10:48:13.668 | UCMDB_Server started.
```
 - Log entries for stopping UCMDB resemble the following:


```
STATUS | wrapper | 2014/07/25 20:53:30.517 | on_exit trigger
matched. Restarting the JVM. (Exit code: 1)

INFO | wrapper | 2014/07/30 10:12:13.870 | Waiting to stop...

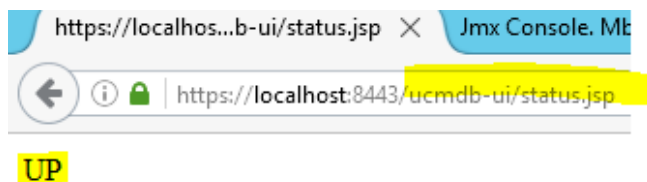
STATUS | wrapper | 2014/07/30 10:12:17.473 | <-- Wrapper Stopped

STATUS | wrapper | 2014/07/30 10:12:18.876 | UCMDB_Server stopped.
```
- Check the status of UCMDB Server. To do this, open **<protocol>://<serverName>:<port>/status** in a browser (access requires authentication). Possible statuses are **Up**, **Starting**, and **Down**.

Status		
Customer Name	Customer ID	labm3_1351
Default Client	1	Up

Detailed Status	
Component	Default Client
	labm3_1351
model_transformation	Up
reconciliation_conf	Up
topology-search-sync	Up
model	Up
classModel	Up
enrichment	Up

- If the UCMDB UI is not accessible, check that the status of the UI server is **UP**. To do this, open `<protocol>://<serverName>:<port>/ucmdb-ui/status.jsp` in a browser.



- Monitor the amount of free space that is available on the server. The server creates files and uses disk space in the directory tree in which the server is installed (by default, `hp\UCMDB\UCMDBServer`). If SOLR is enabled but a standalone SOLR approach is not used, full-text search indexes (`UCMDBServer\search\`) can occupy a significant amount of disk space and grow in proportion to the data model size. Check the *Universal CMDB Support Matrix* document for more information about free space.
- Check that the UCMDB log levels are set to the default value.
 - Log levels in UCMDB are configured by changing the log settings in the `\conf\logs` directory in the UCMDB installation folder.
 - A more detailed log level (such as **DEBUG**) can impact UCMDB performance, as every action performed by the application is slowed down by disc I/O as the logs are written to the physical disk. In general, no logs should ever be set to the DEBUG level unless you are troubleshooting a specific issue.
- If you experience slow performance, check the **slow.log** files.
 - Each operation that takes more than a certain time (by default, 5 seconds; configurable in `operation.slow.log.threshold`) is written to the log.
 - Each operation that exceeds a defined timeout (configurable in `server.sync.session.timeout`) is written to the log.
 - The log is written to every 60 seconds.
 - The log includes thread names and commulative time (how long the operation has taken currently)
 - The log includes all managers, their status, and their queue (Manager name (max threads), Busy=threads; Waiting=threads). If a manager has any threads that are in the Waiting state for more than 10 minutes, UCMDB may experience a performance degradation. If there are more than 100 serving threads, UCMDB Server may run slowly.
 - Check for operations that have an elapsed time greater than 10 minutes.

```
Time: started at 2017-08-14 04:46:20.492, elapsed: 780611 ms
Operation: com.hp.ucmdb.reconciliation.datain.operation.DataInAddOrUpdateData : [ID= 183832510] [
Audit Message: [ID= 183832510] [Customer ID=1] [Changer=UCMDBDiscovery: Host Resources and Applic
Subsystem: Reconciliation Data In
Context: CMDB Context: Customer id = '1' , User ID = '911' , Caller Application = 'AutoDiscovery'
Request (sync) - ID: fe1cf0e2d0ed765c306b4d1a3c322b9a, message: General CMDB request
```

- We recommend that you increase the number of **slow.log** log files to 100. This ensures that the logs capture a longer timeframe for analysis. Make sure that enough free disk space is available.
- Historical and "Top 10 worst" operations are saved to `\runtime\log\statistics`.
 - These logs are written to every 15 minutes.
 - The logs record the operations executed, the different types of operation executed, the average time of operations, and the 10 worst operations.
 - The logs record all operations during the last 15 minutes, together with the average and maximum duration, and the number of operations. You should investigate operations that take more than 10 minutes.
 - Ignore **AutoDiscoveryOperationGetProbeTasks** 30 seconds. This is the default operation timeout.

Note

It is normal for some operations to take a long time. For example, merging a lot of data, performing a history baseline, or purging data may take 20 to 30 minutes. However, this should occur only rarely; if it occurs regularly, you should investigate further.

- Check UI statistics by invoking the **getServicesStatisticsSortByAvgDuration** JMX method.
 - If UCMDB users experience slow UI navigation or login, this JMX method can highlight potential slow areas.
 - Investigate any operations that take more than 60 seconds.

Mbean: UCMDB-UI:name=UI Server statistics. Method: getServicesStatisticsSortByAvgDuration

Service	Successful invocations	Failed invocations	Average Duration
CmdbOperationExecuterService#UcmdbMethod	4	0	1234
LicensingService#UcmdbMethod	1	0	62
SecurityService#UcmdbMethod	3	0	36
MultiTenancyService#UcmdbMethod	5	0	22
CommonService#getAppletInitialData	1	0	16
ResourceManagementService#UcmdbMethod	1	0	15
CommonService#UcmdbMethod	4	0	3
CommonService#getUserLoginInfo	1	0	0

- Check the thread count and memory usage in the **jvm_statistics.log** file.
 - Check that enough memory is assigned to the UCMDB server, based on the deployment type as recommended in the *Universal CMDB Support Matrix* document.

```
INFO - HEAP - [USAGE: 3989.8, FREE: 3.9, TOTAL: 3993.6, MAX: 3993.6]; NON-HEAP -
[USAGE: 181.6, FREE: 50.9, MAX: 240.0]; CLASSES - [Loaded: 26658, Unloaded: 1651,
Left: 25007]; THREADS - [Count: 259]
```

- Check the **cmdb.dal.log** file (**cmdb.dal.slow**) to identify queries that take more than 150 seconds.

```
2017-07-18 12:47:32,226 [RequestProcessorAsyncPool-19858] 483705ms 863448635 755639820
1193357250 309891880 704983185 1216509861 SELECT CDM_ROOT_1.CMDB_ID FROM CDM_ROOT_1
CDM_ROOT_1 ....
```

- Check the output from the **runSupportHandlersForAllCategories** JMX method. This method generates a lot of useful statistics about UCMDB Server.
 - For more information about this JMX method, refer to the "How to Access Support Using the JMX Console" section in the *Universal CMDB Administration Guide*.
- Check the number of active enrichments, TQLs, and pattern-based models.
 - This data can be found in the .zip file that is generated by using the Supportability Services JMX methods.
 - You can also use the **retrieveTqlNames** and **retrieveAllEnrichmentNames** JMX methods.
- Use the System Health Services JMX method to check the active TQL count.
 - It is important to check the number of active TQLs. We recommend that you have a maintenance task to delete unused and test active TQLs and deactivate those that do not need to be active.
 - You can also view this information by running the **retrieveTqlNames** JMX method or by checking the **runSupportHandlersForSpecificCategories** output (**TQL.properties** file).
- Check the cmdb.pattern.statistics.log log file, which displays TQLs and Views, together with statistics about how long each one took, the average time, the maximum time, and the result size.
 - This log shows data for active TQLs only.
 - To check all UCMDB TQLs that are calculated, check the cmdb.tql.calculation.audit.log log file.
 - Check for TQLs that take more than 600 seconds.

```
Finished calculation of pattern [JDBCDataSrc_Database]:
[JDBCDataSrc_Database Num of nodes [2] ; -10 - database ; -11 - jdbcdatasource
Num of links [1] ; -12(-11 --> -10) - join_f] Result size [12000] calculation
time = [JDBCDataSrc_Database= 920.016 ]
Finished calculation of pattern [Probe Info]: [Probe Info Num of nodes [1] ; 1
- discoveryprobegateway Num of links [0]] Result size [1] calculation
time = [Probe Info= 1.230 ]
```

- Check the active TQL status. To do this, run the **showFullReport** JMX method.

Mbean: UCMDB:service=New TQL Scheduler Services. Method: showFullReport

```
Result of listAllGates at Thu Aug 24 14:55:01 EEST 2017
In state GS3_UNLOCKED
Gate1(GS3_UNLOCKED) contains MS_SQL_Servers
Gate10(GS3_UNLOCKED) contains SAN_Computer
Gate11(GS3_UNLOCKED) contains Virtualized_MS_Cluster
Gate12(GS3_UNLOCKED) contains Non-virtualized_Sun_Cluster
Gate13(GS3_UNLOCKED) contains Non-virtualized_Veritas_Cluster
Gate14(GS3_UNLOCKED) contains Virtualized_Service_Guard_Cluster
Gate15(GS3_UNLOCKED) contains Virtualized_HACMP_Cluster
Gate16(GS3_UNLOCKED) contains Non-virtualized_HACMP_Cluster
Gate17(GS3_UNLOCKED) contains OracleConnectWebservices
Gate18(GS3_UNLOCKED) contains SAP_Double_Stack_Instance
Gate19(GS3_UNLOCKED) contains Siebel_Web_To_Middle_Tier
Gate2(GS3_UNLOCKED) contains Non-virtualized_MS_Cluster
Gate20(GS3_UNLOCKED) contains IISConnectWebservice
Gate21(GS3_UNLOCKED) contains Update_Broadcast_IP
Gate22(GS3_UNLOCKED) contains Remove_ip_to_Mainframe_if_Lpar
Gate23(GS3_UNLOCKED) contains Solaris_Zones_Networking
Gate24(GS3_UNLOCKED) contains JDBCDataSrc_Database
Gate25(GS3_UNLOCKED) contains WebService_WebModule
Gate26(GS3_UNLOCKED) contains Build_Layer2_Cis_using_PortNextMAC
Gate27(GS3_UNLOCKED) contains Link_Sender_and_Receiver_Channels
Gate28(GS3_UNLOCKED) contains WeblogicConnectWebservice
Gate3(GS3_UNLOCKED) contains Non-virtualized_J2EE_Cluster
Gate4(GS3_UNLOCKED) contains Non-virtualized_Service_Guard_Cluster
Gate5(GS3_UNLOCKED) contains Virtualized_Sun_Cluster
Gate6(GS3_UNLOCKED) contains Oracle_Servers
Gate7(GS3_UNLOCKED) contains J2EE_Cluster_-_ESX_Compliance
Gate8(GS3_UNLOCKED) contains Virtualized_Veritas_Cluster
Gate9(GS3_UNLOCKED) contains Virtualized_J2EE_Cluster

Result of listAllTqls at Thu Aug 24 14:55:01 EEST 2017
In state TS2_IDLE
Build_Layer2_Cis_using_PortNextMAC(TS2_IDLE) in gate Gate26(GS3_UNLOCKED)
IISConnectWebservice(TS2_IDLE) in gate Gate20(GS3_UNLOCKED)
J2EE_Cluster_-_ESX_Compliance(TS2_IDLE) in gate Gate7(GS3_UNLOCKED)
JDBCDataSrc_Database(TS2_IDLE) in gate Gate24(GS3_UNLOCKED)
```

- Check the CPU usage for the UCMDB process.
 - In Linux environments, this is logged in the **os_statistics.log** file.


```
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root      29472  5.0  63.3 8621540 5071408 ?        S1    00:32   30:43
/opt/hp/UCMDB/UCMDBServer/bin/jre/bin/ucmdb_server -server
```
 - In Windows environments, use Task Manager or other monitoring tools. This information is not stored in the UCMDB logs.
- Check for model consistency using the newDbTool database tool. This is documented in the *Universal CMDB Administration Guide*.
 - Alternatively, you can run the use **findBrokenLinks**, **deleteBrokenLinks**, and **checkModelConsistency** JMX methods.
 - The recommended approach is to use the **newDbTool.bat** tool (not in preview mode) to clean any inconsistent CIs.
- You can check what UCMDB is doing at a specific time by invoking the following JMX methods. The JMX methods reveal which managers are running, together with a full stack trace. This enables you to see which UCMDB operations are running, and their callers.

- **viewOperationsInformation**
- **viewFormattedServerSnapshot**
- **viewServerSnapshot**
- **viewThreadDump**
- **viewLocksInfo**

- Check the **cmdb.reconciliation.audit.log** file for the total time of discovery bulks.

```
INFO - [ID=1612311824] [Customer ID=1] [Changer=UCMDBDiscovery: MZ_Basic node_Host
Connection by WMI] [total time=120.625
```

- The **cmdb.reconciliation.audit.log** file contains information to help you troubleshoot potential slow data-in issues.
- Check the data-in throughput by using the following JMX methods (aging and automatic probe deletion is not included in these statistics):
 - **viewDiscoveryProcessingStatistics**
 - **exportDiscoveryProcessingStatisticsToExcel**
 - For more information about the meaning of the data saved in the Excel file, refer to the "Reconciliation" of the "Supportability Handlers" topic in the *Administration guide*.

	A	B	C	D	E	F	G	H	I	J
		Total Time in hours	Total CIs Number	Rate (CIs/sec)	Daily Usage(% from 24h)	Total Model time	Total Identify time	Total DataIN time	Average CIs per bulk	Number of bulks per day
1	Date									
2	2017-05-20	14.538	3489497	66.672	60.577	4.851	5.094	4.573	4,683.000	745
3	2017-05-21	9.514	1796008	52.436	39.643	3.307	3.384	2.783	3,733.000	481
4	2017-05-22	4.205	947891	62.615	17.521	1.488	1.678	1.026	2,273.000	417
5	2017-05-23	10.430	2004571	53.385	43.460	4.001	3.380	3.015	3,678.000	545
6	2017-05-24	11.117	2189145	54.698	46.322	3.783	4.350	2.942	3,888.000	563
7	2017-05-25	2.912	589093	56.188	12.135	1.095	1.225	0.585	2,337.000	252
8	2017-05-26	7.306	2358333	89.661	30.443	1.815	2.622	2.848	4,211.000	560
9	2017-05-27	7.094	1710448	60.960	22.265	2.444	2.244	2.187	2,644.000	480

UCMDB Database

- Check that the UCMDB Database is configured based on the recommendations in the *Database guide*.
- Check if there are any UCMDB Database connection from JMX by using the **viewDbConnectionSummary** JMX method.
 - There should be no connections. Database connection leaks are displayed for many hours, until the server restarts.

Mbean: KPI:service=Dashboard. Method: viewDbConnectionSummary

```
Total number of connection is: 4
The timeout for a connection is: 600000
Connection org.apache.commons.dbcp2.PoolableConnection@6ef8b95f was opened at : Thu Aug 24 14:48:42 EEST 2017 by the thread qtp1245001884-1321
Stack Trace:
[
    sun.management.ThreadImpl.getThreadInfo(Native Method),
    sun.management.ThreadImpl.getThreadInfo(ThreadImpl.java:176),
    sun.management.ThreadImpl.getThreadInfo(ThreadImpl.java:139),
    com.mercury.topaz.cmbd.server.util.concurrent.ThreadUtil.getStackTrace(ThreadUtil.java:117),
    com.mercury.topaz.cmbd.server.manage.dal.OpenConnectionDetails.(OpenConnectionDetails.java:28),
    com.mercury.topaz.cmbd.server.manage.dal.CmdbDalConnection.(CmdbDalConnection.java:96),
    com.mercury.topaz.cmbd.server.manage.dal.CmdbDalConnectionFactory.create(CmdbDalConnectionFactory.java:33),
    com.mercury.topaz.cmbd.server.manage.dal.ConnectionPoolManager.getConnection(ConnectionPoolManager.java:490),
```

- Check if there are any inconsistencies between the UCMDB class model and the database tables by using the **rebuildModelDBSchemaAndViews** JMX method.
 - If inconsistencies exist, run the **rebuildModelDBSchemaAndViews** and **rebuildModelViews** JMX method.
 - The class model must be aligned with the Database tables and views.

Mbean: UCMDB:service=DAL services. Method: rebuildModelDBSchemaAndViews

```
All tables in DB are aligned with the class model
```

- Check if there are any missing indexes. To do this, run the **showAndRebuildMissingIndexes** JMX method.
 - One of the common root causes for a slow environment is index fragmentation.
 - Check the *Universal CMDB Database Guide* for details about how to monitor index fragmentation and how to create an index rebuild plan based on the discovery schedule and load of the UCMDB server.
- Check the UCMDB connection to the Database by using the **testConnectionToDB** JMX method. The method displays a basic summary of few DB SQL statements.
 - Use the method to establish a baseline. Run the method when the system is busy and when it is idle. This can highlight possible UCMDB database configuration issues.
- Fix any history tables inconsistencies by using the **alignHistoryForType** JMX method.
 - If there are inconsistencies in the history tables (for example, missing columns) data in operations will fail.
- Delete any unused history tables from the Database to save disk space on the Database server. To do this, run the **deleteUnboundHistoryTables** JMX method.

Universal Discovery

- Monitor the following Windows services on the Data Flow Probe server to make sure the services are running:

- "UCMDB_Probe_XmlEnricher" (Display name: "HP Universal Discovery XML Enricher")
- "UCMDB_Probe" (Display name: "UCMDB Probe")
- "UCMDB_Probe_DB" (Display name: "UCMDB_Probe_DB")
- It is necessary to monitor the amount of free space available on the probe. The probe creates files and uses disk space in the directory tree in which the probe is installed (by default, **C:\hp\UCMDB\DataFlowProbe**). Scan files (especially if software scanning is enabled) can occupy a significant amount of disk space. If the free disk space is less than 100 Mb, you should free more space.
- When you encounter problems, refer to the log files (by default, these are located in **C:\hp\UCMDB\DataFlowProbe\runtime\log**). Logs that typically contain information about fatal errors are WrapperProbeGw.log and probe-error.log. If a log contains an entry that resembles the following example, you must address the issue:

```
<2017-08-23 10:13:12,594> 50626250 [FATAL ] XXXXXXXXX
```

- Monitor the probe status using the **MAM:service=Discovery manager::monitorDiscoveryProbes** UCMDB server JMX method. The method displays information that resembles the following:

```
RUBMYDGLNO
```

```
Monitors Discovery Probe
```

```
Last Report Time: No reports yet
Reported 0 CIs
```

```
Last Access Time: 08/24/2017 13:55:30
```

Monitor the Last Access Time value. If the last access time is not updated for one minute, the probe is disconnected.

- Run the **JobsInformation > viewJobsStatuses** probe JMX method.
 - This JMX method displays the number of stuck triggers in the probe.
 - When the number of stuck threads reaches 8, the probe will restart.
 - Monitor this JMX method to be alerted about probe restarts.

Mbean: Local_VMUCMDB36:type=JobsInformation. Method: viewJobsStatuses

The current time is:Sun Sep 01 15:38:32 EDT 2013

Discovery Jobs Information table

No.	Job Name	Status	Triggered CIs	Errors & Warnings	Last invocation	Next invocation	Last Total run duration (seconds)	Avg run duration (seconds)	Recurr
1	MZ_CSO_MZ_1_Activityv1_CSO_InProgressJob1	Done/Total Triggers (0/1) Threads: 0 (1 stuck threads)	1	1	9/1/13 3:37 PM	9/2/13 3:24 PM	0	0	1
2	MZ_CSO_MZ_1_Activityv1_CSO_prepareDataJob	Scheduled	1	0	9/1/13 3:36 PM	9/2/13 3:23 PM	21	21.265	1

- Run the **GwTaskResultDistributer** Probe JMX method.
 - Check "NumberOfUnsentResults". This is the number of pending bulks waiting in the probe to be sent to the server.
 - Drill down to the **GwTaskResultDistributer > viewUnsentResults** JMX method:
 - Check how long each bulk is waiting in the probe queue before it is sent to server.
 - Bulks may wait too long in the probe queue because of a server-side problem.

Mbean: Probe_VMUCMDB36:type=GwTaskResultDistributer. Method: viewUnsentResults

Probe Mgr	Time Stamp	Task ID	Job Id	Num of objects to add or update	Num of objects to delete	action
VMUCMDB36	Sun Sep 01 16:45:34 IDT 2013	MZ_CSO_MZ_2_Activity1_CSO_InProgressJob3	MZ_CSO_MZ_2_Activity1_CSO_InProgressJob3	5000	0	view result/
VMUCMDB36	Sun Sep 01 16:45:37 IDT 2013	MZ_CSO_MZ_2_Activity1_CSO_InProgressJob3	MZ_CSO_MZ_2_Activity1_CSO_InProgressJob3	5000	0	view result/
VMUCMDB36	Sun Sep 01 16:45:40 IDT 2013	MZ_CSO_MZ_2_Activity1_CSO_InProgressJob3	MZ_CSO_MZ_2_Activity1_CSO_InProgressJob3	10000	0	view result/
VMUCMDB36	Sun Sep 01 16:45:45 IDT 2013	MZ_CSO_MZ_2_Activity1_CSO_InProgressJob3	MZ_CSO_MZ_2_Activity1_CSO_InProgressJob3	5000	0	view result/
VMUCMDB36	Sun Sep 01 16:45:48 IDT 2013	MZ_CSO_MZ_2_Activity1_CSO_InProgressJob3	MZ_CSO_MZ_2_Activity1_CSO_InProgressJob3	5000	0	view result/

If the number of unsent results is increasing and is more than 1 million, you should investigate further.

- Monitor the probe JVM heap by checking the **jvm_statistics.log** file (located in **\HPUCMDB\DataFlowProbe\runtime\log**). The log contains content that resembles the following:

```
<2017-07-12 12:58:58,660> [DEBUG] [Timer-1] (JVMMonitorInfo.java:13) - HEAP - [USAGE: 241.5, FREE: 253.5, TOTAL: 494.9, MAX: 1979.8]; NON-HEAP - [USAGE: 41.7, FREE: 0.9, TOTAL: 42.6, MAX: 1024.0]; CLASSES - [Loaded: 7093, Unloaded: 0, Left: 7093]; THREADS - [Count: 89]
```

If the free size of HEAP is less than 10, increase the amount of memory allocated to the probe's JVM.

- Check the **probeGW-taskResults.log** file:

- Example:

```
<2014-07-24 12:08:37,925> 145673 [INFO ] (TaskResultsSenderThread.java:166) - Process Result Time Statistics - Total Time:5651, Results size:6, Time To get Tasks:9, Time to send results: 78, Time until server gets the results:90132, Time to get results status:5016, Time to process Id Mapping:98, Time to update warnings:0, Time to activate automatic deletion:378, Time to inform the sent results is:4, Time to finish deleting results:25, Total time to handle Successful results:505
```

- The above entry is printed for each group of discovery results bulks. Each entry includes the following measurements:

- "Time until server gets the results": Select results from DB (in probe side)
- "Time to get results status": Server side work - reconciliation\model update
- "total time to handle Successful results": Update probe DB table

- Check the result processing queue time on the server side in the **mam.autodiscovery.results.stat.log** log file (located in **\HPUCMDB\UCMDBServer\runtime\log**).

```
2017-07-12 16:47:12,046 INFO [Process Results Thread-Test Import Vector From File] - Processing result of 'Unix-Daily' from probe: 'USPMVAPP557' took 3948msec. Waiting time (in result processing queue): 133349
```

If the waiting time is more than 120000, you should investigate further.

- Check the queue size in the **mam.autodiscover.log** log file (located in **\HPUCMDB\UCMDBServer\runtime\log**).

```
2017-08-22 20:20:25,275 INFO [notification_publish_task0-130] - [DISPATCH] Adding dispatch task to queue. New queue size: 846. Task: Redispatch Trigger Cis
```

- Enable the probe server Database performance log. To do this, follow these steps:
 1. Open the `\HP\UCMDB\UCMDBServer\conf\log\mam.properties` file, and then add the following code:

```
#####
### DB.performance.appender          ##
#####
log4j.category.mam.collectors.dal=DEBUG,db.performance.appender
log4j.appender.db.performance.appender=com.mercury.topaz.cmdb.shared.base.log.BetterRollingFileAppender
log4j.appender.db.performance.appender.File=${logs.dir}/db.performance.log
log4j.appender.db.performance.appender.MaxFileSize=${def.file.max.size}
log4j.appender.db.performance.appender.MaxBackupIndex=${def.files.backup.count}
log4j.appender.db.performance.appender.layout=org.apache.log4j.PatternLayout
log4j.appender.db.performance.appender.layout.ConversionPattern=%d %-5p - %m%n
```

2. Locate the `db.performance.log` file (located in `\HP\UCMDB\UCMDBServer\runtime\log`).

```
2017-08-02 15:35:48,031 DEBUG -
context=dalcollectors;operation=setCIs;duration=500349
```

The unit of duration is milliseconds. If one operation exceeds 2 mins, investigate further.

Store and Forward Server

- **Windows:**
 - Monitor the following two Windows services to make sure the services are running:
 - "hpudStoreAndForward" (Display name: "HP Universal Discovery Store and Forward Server")
 - "hpudApacheSF" (Display name: "HP Universal Discovery Store and Forward Web Server")
- **Linux:**
 - Monitor the following Store and Forward server processes to make sure they are still running:
 - storeNforward
 - httpd (embedded Apache web server daemon)
- The Store and Forward server also serves the XML status document showing the current status of the server over HTTP (or HTTPS depending on how the server is configured). The status is available at the following URL:

[http\[s\]://<StoreAndForwardServer>:<port>/server_status.xml](http[s]://<StoreAndForwardServer>:<port>/server_status.xml)

If you view the server status in a browser, a user-friendly HTML page is displayed. However, monitoring software can read the XML itself. For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="server_status.xsl"?>
<StoreAndForwardStatus version="1.0">
  <statusDirBytes>0</statusDirBytes>
  <incomingDirBytes>0</incomingDirBytes>
  <inProgressDirBytes>0</inProgressDirBytes>
  <resumeDirBytes>0</resumeDirBytes>
  <freeDiskSpaceBytes>50882215936</freeDiskSpaceBytes>
  <numIncomingFiles>0</numIncomingFiles>
  <numInProgressFiles>0</numInProgressFiles>
```

```
<numResumeFiles>0</numResumeFiles>  
<modifiedTime>06/22/14 08:23:30</modifiedTime>  
</StoreAndForwardStatus>
```

- It is important to monitor the amount of free disk space available on the disk that the store and forward server uses for its data. If there are network connectivity problems and the store and forward server is not able to send the files on a timely basis, the files will start to accumulate and consume disk space. The directories that are used for storage are configured in the config.ini file used by the store and forward server:

```
[BaseDirectories]  
dataDir=xxx  
incomingDir=yyy
```

More details can be found in the UCMDB documentation.

Micro Focus Trademark Information

MICRO FOCUS and the Micro Focus logo, among others, are trademarks or registered trademarks of Micro Focus (IP) Limited or its subsidiaries in the United Kingdom, United States and other countries. All other marks are the property of their respective owners.

Company Details

Company name: Micro Focus International plc

Place of registration: England and Wales

Registered number: 5134647

Registered address: The Lawn, 22-30 Old Bath Road, Berkshire, RG14 1Q