|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Sybase ASE** | | | | | | pre-12.5.0.3 | **12.5.0.3+** | **15.0** |  |  | | Last updated: 28 December 2008 | | | | | | | **MDA/Monitoring tables in ASE** | | | | |  |  |  |  | |
| Contents  [Introduction](http://www.sypron.nl/mda.html#intro)  [Presentations](http://www.sypron.nl/mda.html#downloads) [Tools](http://www.sypron.nl/mda.html#tools) [Installation steps](http://www.sypron.nl/mda.html#install) [Changes and Enhancements to MDA tables since ASE 12.5.0.3 (updated for 15.0.3 and 15.0.1 CE)](http://www.sypron.nl/mda.html#enh) [Some MDA query examples](http://www.sypron.nl/mda.html#q)  Introduction Starting in ASE 12.5.0.3, a new feature called 'MDA tables' is available to ASE users. These MDA tables provide access to low-level monitoring information in ASE; since the MDA tables can be accessed with regular SQL select statements, they're much easier to use than products like Monitor Server/Historical Server.   The MDA tables are proxy tables, located in the master database. All MDA tables are named master..mon*XXX*, where *XXX* indicates the type of information in the table.  One example is master..monCachedObject, which contains details about the current data cache usage for different tables and indexes.   'MDA' is short for 'Monitoring Data Access', 'Monitoring and Diagnostics for ASE', 'Monitoring and Diagnostic Agent' or 'Monitoring and Diagnostic Access', depending on who you ask. As 'monitoring' seems to be a common denominator, the MDA tables are also referred to as 'monitoring tables' (although they're not normal tables, but in fact proxy tables mapped to RPCs).   Presentations  Here are some useful presentations related to MDA tables:   * [My Techwave 2003 presentation about MDA table basics](http://www.sypron.nl/tw2003.mda.zip) (190Kb). * [The more advanced Techwave 2004 presentation about MDA tables (by Peter Dorfman and Rob Verschoor)](http://www.sypron.nl/tw2004.mda.zip) (428Kb). * [The Techwave 2005 presentation about MDA tables (by Peter Dorfman, Jeff Tallman and Michael Wallace)](http://www.sypron.nl/tw2005.mda.zip) (750Kb).   Tools  These are some useful MDA-related tools:   * I wrote [some handy stored procedures](http://www.sypron.nl/sp_mda.sql) for easy access to some of the MDA tables.  Quick usage info:   + sp\_mda\_help - provides a quick way of searching for specific MDA tables or columns, based on a substring. Examples:   + -- List all cache-related MDA tables:   + sp\_mda\_help cach   + -- List all columns & parameters for a specific table:   + sp\_mda\_help monObjectActivity   + -- List all columns having something to do with SQL text:   + sp\_mda\_help null, sql   + sp\_mda\_io - monitors logical (and physical) I/O usage by T-SQL statements. Examples:   + -- Reports I/O usage since previous call   + sp\_mda\_io   + -- Reports I/O for a particular T-SQL batch:   + sp\_mda\_io "...T-SQL statement(s)..."   + sp\_mda\_wait - displays detailed information about wait events during a certain interval (default=10 seconds) Examples:   + -- displays wait information for the entire ASE server   + sp\_mda\_wait 'server'   + -- displays wait information for a specific process   + sp\_mda\_wait '<spid\_no>'   + -- displays wait information for all processes along with   + -- additional process details (such as the SQL currently   + -- being executed)   + sp\_mda\_wait 'all'   For on-line usage information, execute these procedures with '?' as the first parameter.   * Here's a simple ['collector' procedure](http://www.sypron.nl/sp_mda_collect.sql) to continuously extract data from the MDA history tables and store it in a permanent table. See the header of the script for further instructions (you'll need to create a separate database, for instance). * David Wein wrote a great procedure named sp\_\_idleReaper to automatically kill idle user processes in ASE. This procedure can be found on [CodeXchange](http://ase.codexchange.sybase.com/servlets/ProjectDocumentList?folderID=292) (a local copy is [here](http://www.sypron.nl/sp__idleReaper.sql)).   Installation steps  Before querying the MDA tables, they must have been installed first, and some ASE configuration parameters must have been set.  Follow these installation steps:  -- First, ensure that the configuration parameter  -- 'enable cis' is set to 1 (if not, an ASE  -- restart is needed)  sp\_configure 'enable cis', 1  go  -- Add 'loopback' server name alias (assuming @@servername  -- is also defined in the interfaces file)  -- (NB: this step is no longer required in 15.0 ESD#2 or later)  use master  go  sp\_addserver loopback, null, @@servername  go  -- Test this configuration:  -- (NB: this step is no longer required in 15.0 ESD#2 or later)  set cis\_rpc\_handling on  go  --  -- Alternatively, run:  -- sp\_configure 'cis rpc handling', 1  -- ...and disconnect/reconnect your session  exec loopback...sp\_who -- note: 3 dots!  go  -- Install the MDA tables. Important: do NOT run this  -- script with 'sqsh' as it'll give errors: 'sqsh' sees  -- a '$' as the start of a sqsh variable, and this messes  -- up the native RPC names, since these start with a  -- '$' as well.  -- Solution: either usq 'isql' as below, or run 'sqsh'  -- with the '-Lexpand=0' option to disable sqsh's  -- variable expansion feature (thanks to Paul Harrington  -- for this tip).  -- (NB: as of ASE 15.0.2, this script is part of 'installmaster')  isql -U sa -P yourpassword -S YOURSERVER \  -i $SYBASE/$SYBASE\_ASE/scripts/installmontables  -- Assign 'mon\_role' to logins allowed MDA access  -- (this also applies to the 'sa' login!)  use master  go  grant role mon\_role to sa  -- grant to other logins or roles here, as needed  go  -- Test basic MDA configuration:  -- (note: you may need to disconnect/reconnect first  -- to activate 'mon\_role' when you just granted this  -- role to the login you're currently using)  select \* from master..monState  go  -- Now enable all configuration parameters;  -- these are all dynamic (except the last one)  -- For all 'pipe' tables, the number of  -- messages is set to 100 here, but you may want  -- to choose a large size.  --  sp\_configure "enable monitoring", 1  go  sp\_configure "sql text pipe active", 1  go  sp\_configure "sql text pipe max messages", 100  go  -- keep this disabled if you don't need query plan info  -- and you want to limit performance impact  --sp\_configure "plan text pipe active", 1  go  sp\_configure "plan text pipe max messages", 100  go  sp\_configure "statement pipe active", 1  go  sp\_configure "statement pipe max messages", 100  go  sp\_configure "errorlog pipe active", 1  go  sp\_configure "errorlog pipe max messages", 100  go  sp\_configure "deadlock pipe active", 1  go  sp\_configure "deadlock pipe max messages", 100  go  sp\_configure "wait event timing", 1  go  sp\_configure "process wait events", 1  go  sp\_configure "object lockwait timing", 1  go  sp\_configure "SQL batch capture", 1  go  sp\_configure "statement statistics active", 1  go  sp\_configure "per object statistics active", 1  go  --  -- As of ASE 15.0.2, also run the following one:  --  sp\_configure "enable stmt cache monitoring", 1  go  -- This is the only static parameter. Set to  -- a higher value (the setting is in bytes  -- per user connection) if you're expecting  -- a lot of (or long) SQL batches  sp\_configure "max SQL text monitored", 2048  go  -- The following option must be enabled only when  -- using DBXRay, so it is not relevant when only  -- using the MDA tables directly. It is mainly  -- included here for completeness and to pre-empt  -- your questions...  sp\_configure "performance monitoring option", 1  go  -- Now you're ready to use the MDA tables. Have fun!  Changes and Enhancements to MDA tables since ASE 12.5.0.3  The MDA tables were first introduced in ASE 12.5.0.3. This section lists the enhancements to the MDA tables in subsequent versions of ASE. I'll try to keep this list up-to-date.   **ASE 12.5.1 IR** -- 5 new columns:   * monErrorLog.State - 'state' of an error * monOpenDatabases.QuiesceTag - the tag specified with 'quiesce database' (if any) * monOpenDatabases.SuspendedProcesses - number of currently suspended processes due to log-full condition in this database * monProcessWorkerThread.FamilyID - for parallel queries, the spid of parent process * monProcessWorkerThread.ParallelQueries - total # parallel queries attempted   **ASE 12.5.2 IR** -- 2 new columns:   * monProcessObject.TableSize - table size in Kbyte * monProcessActivity.WorkTables - total number of work tables created by the process * Note: the uninitialized milliseconds in monSysStatement.StartTime / EndTime have been fixed in 12.5.2   **ASE 12.5.3 IR** -- 4 new columns:   * A column ServerUserID has been added to monProcessActivity, monProcessSQLText and monSysSQLText; this column is the login's 'suid'. * monProcessProcedures.LineNumber - the line in the procedure currently being executed   In addition, as of 12.5.3, monOpenObjectActivity contains details about tables and indexes only. Prior to 12.5.3, this table could contain rowsa row for an executed stored procedure, but these details (like the Operations column) were not reliable.  **ASE 12.5.3 ESD#2** -- 4 new columns:   * monEngine.Yields - #times this engine yielded to the Operating System * monEngine.DiskIOChecks - #times this engine checked for asynchronous disk I/O * monEngine.DiskIOPolled - #times this engine polled for completion of outstanding asynchronous disk I/O. * monEngine.DiskIOCompleted - #asynchronous disk I/Os that were completed when this engine polled   **ASE 15.0** -- 2 new tables and various new/changed columns:   * The new table monOpenPartitionActivity reports monitoring statistics at partitition level * The new table monLicense shows the details for the license keys that are active in this server   New columns in monEngine:   * HkgcMaxQSize - maximum #items that can be queued for HK garbage collection in this engine * HkgcPendingItems - #items yet to be garbage-collected by the HK in this engine * HkgcHWMItems - maximum #pending items queued for HK garbage collection at any instance of time since server restarted * HkgcOverflows - #items that could not be queued for HK garbage collection due to queue overflows   New columns in monCachedObject:   * PartitionID, PartitionName - partition name and ID * TotalSizeKB - the total size of the object (table or index)   New columns in monOpenObjectActivity:   * DBName - the databasename corresponding to DBID   New/changed columns in monProcessObject:   * PartitionID, PartitionName - partition name and ID * TableSize has been changed to PartitionSize - this reflects the size of the partition for the object   **ASE 15.0 ESD#2** -- 5 new columns   Perhaps the most important enhancement in ASE 15.0 ESD#2 is the new 'materialized' option with which the MDA proxy tables are created. In 15.0 ESD#2, the MDA tables no longer use the 'backdoor' connection back into to the server itself and consequently, the 'loopback' server name alias is no longer needed either. This new feature reduces some of the overhead of querying the MDA tables. There's nothing you have to do to benefit from this new feature other than running the 'installmontables' script that comes with 15.0 ESD#2.   New columns in monLocks:   * BlockedState - identifies whether a lock is being blocked or is blocking others * BlockedBy - for blocked locks, identifies the session this lock is being blocked by   New columns in monSysStatement:   * RowsAffected - the number of rows affected by the statement, similar to @@rowcount * ErrorStatus - the SQL return status of the statement, similar to @@error   New column in monProcessStatement:   * RowsAffected - the number of rows affected by the statement, similar to @@rowcount   **ASE 15.0.1, esd#1, #esd2, esd#3** -- no changes were made  **ASE 15.0.1 Cluster Edition** -- various changes, 15 new tables (updated: 28 December 2008)  The following changes are for ASE Cluster Edition only -- up to ASE 15.0.3 (the classic, non-cluster-edition flavour of ASE, that is), none of the following applies.   * A new column InstanceId has been added to various MDA tables.   15 new MDA tables were added in ASE CE:   * monCIPC, monCIPCEndpoints, monCIPCMesh, monCIPCLinks contain statistics about communication over the cluster interconnect (CIPC) * monLogicalCluster, monLogicalClusterAction, monLogicalClusterInstance, monLogicalClusterRoute, monSysLoad, monWorkload, monWorkloadPreview, monWorkloadProfile, monWorkloadRaw contain information about the logical cluster configuration and the workload manager features. * monTempdbActivity contains information about activity in instance-specific local temporary databases (i.e. not in global temporary databases like tempdb). * monClusterCacheManager is for internal diagnostics only.   **ASE 15.0.2** -- 2 new tables, 11 new columns   First, the definition of the MDA tables is moved into the installmaster script, so that they're automatically installed/updated when installmaster is run (you \*do\* run this after installing an EBF , right?). Note that installmontables is still available as a separate script, but this is intended as a template for special cases like setting up the MDA tables in a different server or database.   New tables:   * The new tables monCachedStatement and monStatementCache report monitoring statistics about the statement cache. These tables are controlled by the new configuration parameter 'enable stmt cache monitoring'.   New columns in monSysStatement:   * Ssqlid - a unique identifier of a SQL statement, maps to monCachedStatement.SSQLID * HashKey - the hash key value for the SQL text of a SQL statement, maps to monCachedStatement.HashKey   New columns in monOpenObjectActivity and monOpenPartitionActivity:   * HkgcRequests, HkgcPending, HkgcOverflows - information about the Housekeeper's garbage collection activity for an object or partition   New columns in monLocks and monDeadLock (these columns contain diagnostic information about a lock, for support purposes only):   * SourceCodeID was added to monLocks * HeldSourceCodeID and WaitSourceCodeID were added to monDeadLock   Among various bugfixes in 15.0.2, one that is worth mentioning is the number of table scans or index scans on a table can now be reliably derived from monOpenObjectActivity.UsedCount for rows with IndexID = 0. Previously, this value was not correct as it included accesses via a clustered index as well.  **ASE 15.0.2 #esd2, 15.0.2 esd#4** -- no changes (NB: 15.0.2 esd#3 was not released)  **ASE 15.0.2 #esd5** -- 12 new columns   New column in monEngine:   * MaxOutstandingIOs - the max.# of I/Os pending for each engine   New column in monProcessNetIO:   * NetworkEngineNumber - engine handling the network IO for this SPID   New column in monProcessProcedures:   * StatementNumber - the statement in the stored procedure currently being executed   New columns in monOpenDatabases:   * LastCheckpointTime - date/time of the start of the last checkpoint for this database * LastTranLogDumpTime - date/time of the start of the last log dump for this database   New column DBName was added to the following tables:   * monLocks * monProcessStatement * monSysStatement * monSysPlanText * monCachedStatement   New column ServerUserID was added to the following tables:   * monProcess * monProcessWaits   **ASE 15.0.2 esd#6** -- no changes were made  **ASE 15.0.3** -- 5 new tables, 2 new columns, 1 column removed (updated: 28 December 2008)   New tables:   * The new tables monSQLRepActivity and monSQLRepMisses report statistics about SQL Statement replication (note that this new feature requires Replication Server 15.2 as well). * Three new (and so far, undocumented) tables monPCIBridge, monPCISlots, monPCIEngine have been added. These tables are related to new internals in 15.0.3, and for the time being, these are likely meaningful for TechSupport only.   New columns in monSysStatement:   * ProcNestLevel - the nesting level on which the statement executed * StatementNumber - a sequence number, starting at 0, for each statement in a batch or stored procedure, in the order of their actual execution.   One column, TableCount, was removed from monCachedStatement.  Some MDA query examples  Below are some examples of queries against the MDA tables -- adjust as needed.  Please note that this is not aiming for completeness, but merely an attempt to get you started. For more examples, see the various [presentations](http://www.sypron.nl/mda.html#downloads) available above.  --====================================================================  -- find out what queries are running right now  select \* from master..monProcessSQLText  -- ... and the corresponding resource usage  select \* from master..monProcessStatement  -- keep in mind that the information in these tables changes  -- very rapidly, so the tables may not always match!  --====================================================================  -- when tempdb is full and you cannot select from sysprocesses, syslocks  -- etc. anymore, still find out what's running :  select \* from master..monProcess  --====================================================================  -- show recent queries that ran longer than 100 millisec  -- or took more than 1000 I/Os  drop table #tsql, #ts  go  declare @kpid int  select @kpid = kpid from master..sysprocesses where spid = @@spid  select \* into #tsql from master..monSysSQLText where KPID != @kpid  select \* into #ts from master..monSysStatement where KPID != @kpid  select KPID, BatchID, LineNumber, LogicalReads,  datediff(ms, StartTime, EndTime) millisec, WaitTime from #ts  where datediff(ms, StartTime, EndTime) > 100  or LogicalReads > 1000 order by 1,2,3  go  -- to find the corresponding SQL text, pick a KPID from the above  -- query's output:  select \* from #tsql where KPID=  order by BatchID, SequenceInBatch  go  --====================================================================  -- find seemingly unused indexes in the current database:  select "Database" = db\_name(DBID),  "Table" = object\_name(ObjectID, DBID),  IndID = IndexID, si.name  from master..monOpenObjectActivity oa, sysindexes si  where oa.ObjectID = si.id  and oa.IndexID = si.indid  and UsedCount = 0  and OptSelectCount = 0  and ObjectID > 99  and IndexID > 1 and IndexID != 255  and DBID = db\_id() -- remove this to run server-wide  order by 1,2  --====================================================================  -- The big batch below calculates delta values between two  -- invocations. To initialise, run this once in this session:  select dt=getdate(), \* into #monOOA  from master..monOpenObjectActivity  go  -- Now run this batch repeatedly:  set nocount on  select dt=getdate(), \* into #monOOA\_new  from master..monOpenObjectActivity  where object\_name (ObjectID, DBID) not like '#monOOA%'  select secs = datediff(ms, t.dt, n.dt),  n.ObjectID, n.DBID, n.IndexID,  LogicalReads = n.LogicalReads - t.LogicalReads,  PhysicalReads = n.PhysicalReads - t.PhysicalReads,  Operations = n.Operations - t.Operations,  LockWaits = n.LockWaits - t.LockWaits,  RowsInserted = n.RowsInserted - t.RowsInserted,  RowsUpdated = n.RowsUpdated - t.RowsUpdated,  RowsDeleted = n.RowsDeleted - t.RowsDeleted,  LockRequests = n.LockRequests - t.LockRequests  into #monOOA\_delt2  from #monOOA t, #monOOA\_new n  where t.ObjectID =\* n.ObjectID  and t.DBID =\* n.DBID  and t.IndexID =\* n.IndexID  -- calculate sorting totals  select ObjectID, DBID, LogicalReads = sum(LogicalReads),  PhysicalReads = sum(PhysicalReads), Operations=sum(Operations),  LockWaits=sum(LockWaits)  into #monOOA\_sort  from #monOOA\_delt2  group by ObjectID, DBID  select t.\*, sort\_LogicalReads = s.LogicalReads,  sort\_PhysicalReads = s.PhysicalReads,  sort\_Operations = s.Operations,  sort\_LockWaits = s.LockWaits  into #monOOA\_delta  from #monOOA\_delt2 t, #monOOA\_sort s  where t.ObjectID = s.ObjectID  and t.DBID = s.DBID  -- calc total LIO  declare @sum\_lio numeric(10,1)  select @sum\_lio = sum(LogicalReads)  from #monOOA\_delt2  print "sum\_lio = %1!", @sum\_lio  if @sum\_lio in (0, NULL) select @sum\_lio = 1  -- select the final data  set rowcount 35  print ""  print "Tables/indexes consuming most I/O:"  select TableName = object\_name(ObjectID, DBID), IndexID, LogicalReads,  IOPercent= convert(numeric(4,1),100\*LogicalReads/@sum\_lio),  PhysicalReads, Operations, LIOperOp=  convert(int,ceiling(case Operations when 0 then 0  else sort\_LogicalReads/(1.0\*Operations) end)),  LockWaits, RowsInserted, RowsUpdated, RowsDeleted  from #monOOA\_delta  where (100\*LogicalReads/@sum\_lio) >= 0.1  order by 4 desc, 1,2  print ""  print "Most frequently executed/accessed procedures/views:"  select TableName = object\_name(ObjectID, DBID), Operations  from #monOOA\_delta  where Operations > 0 and LogicalReads = 0  order by Operations desc, 1  print ""  print "Tables with lock contention:"  select TableName = object\_name(ObjectID, DBID), IndexID, LockWaits,  LockRequests, LogicalReads, IOPercent=  convert(numeric(3,1),100\*LogicalReads/@sum\_lio),  PhysicalReads, Operations, RowsInserted, RowsUpdated, RowsDeleted  from #monOOA\_delta  where LockWaits > 0  order by LockWaits desc, LogicalReads desc, 1,2  set rowcount 0  truncate table #monOOA  insert #monOOA select \* from #monOOA\_new  drop table #monOOA\_new, #monOOA\_delta, #monOOA\_delt2, #monOOA\_sort  set nocount off  go  --====================================================================  More information coming...  The MDA tables are still a new area, with many unexplored applications. More information will be added to this page, so check back here regularly. |