|  |
| --- |
| AseMon – The GUI Version |
| Users Guide |
| Version x.0 |

|  |
| --- |
| Goran Schwarz  6/21/2010 |

[What is this tool all about 4](#_Toc281993410)

[Different monitor modes 4](#_Toc281993411)

[GUI (Online) mode 4](#_Toc281993412)

[In-Memory history 4](#_Toc281993413)

[Offline mode 4](#_Toc281993414)

[Command Line parameters 5](#_Toc281993415)

[Storing Performance Counter data in a Persistent Counter Storage 6](#_Toc281993416)

[Viewing Performance Counter data from a Persistent Counter Storage 6](#_Toc281993417)

[Connecting to ASE 7](#_Toc281993418)

[Connecting to ASE and Storing Performance Counters in a database 8](#_Toc281993419)

[Connecting to a offline storage. 9](#_Toc281993420)

[System Performance Counter Sets 11](#_Toc281993421)

[Summary 20](#_Toc281993422)

[Objects 20](#_Toc281993423)

[Processes 23](#_Toc281993424)

[Databases 24](#_Toc281993425)

[Temp Db 25](#_Toc281993426)

[Waits 26](#_Toc281993427)

[Engines 27](#_Toc281993428)

[System Load 28](#_Toc281993429)

[Data Caches 29](#_Toc281993430)

[Pools 30](#_Toc281993431)

[Devices 31](#_Toc281993432)

[IO Sum 32](#_Toc281993433)

[IO Queue 33](#_Toc281993434)

[Spinlock Sum 34](#_Toc281993435)

[Sysmon Raw 35](#_Toc281993436)

[Cached Procedures 36](#_Toc281993437)

[Procedure Cache 37](#_Toc281993438)

[Cached Objects 38](#_Toc281993439)

[Errorlog 40](#_Toc281993440)

[Deadlock 41](#_Toc281993441)

[Proc Cache Module Usage 42](#_Toc281993442)

[Proc Cache Memory Usage 43](#_Toc281993443)

[Statement Cache 44](#_Toc281993444)

[Statement Cache Details 45](#_Toc281993445)

[Active Objects 46](#_Toc281993446)

[Active Statements 47](#_Toc281993447)

[Blocking 48](#_Toc281993448)

[Missing Statistics 49](#_Toc281993449)

[User Defined Counter Sets 50](#_Toc281993450)

[Using the ‘User Defined Counter’ Wizard 52](#_Toc281993451)

[Adding entries manually to the configuration file 52](#_Toc281993452)

[Reading ‘offline data’ using the GUI 52](#_Toc281993453)

# What is this tool all about

# Different monitor modes

## GUI (Online) mode

This is the GUI mode, where you connect to a ASE and monitor what is happening in the system right now.

### In-Memory history

## Offline mode

This mode is primarily for storing monitored data in a database, which can be viewed by someone at a later stage. The stored data can be viewed from the AseMon GUI or directly from “Persistent Counter Storage” database using your own reports or SQL statements.

The most efficient way to store data in the Counter Persistent Storage database is to do it via the non GUI mode, or daemon mode. This persists the desired/specified counter sets in a database

# Command Line parameters

usage: asemon [-c <cfgFile>] [-t <filename>] [-h] [-v]

[-U <user>] [-P <passwd>] [-S <server>]

[-n <cfgFile|cmNames>] [-s <seconds>] [-d <dbname>]

options:

-c,--config <cfgName> The Main Config file

-t,--tmpConfig <filename> Config file where temporary stuff are stored.

-h,--help Usage information.

-v,--version Display AseMon and JVM Version.

-U,--user <user> Username when connecting to server.

-P,--passwd <passwd> Password when connecting to server. (null=noPasswd)

-S,--server <server> Server to connect to.

Switches for offline mode:

-n,--noGui <cfgFile|cmNames> Do not start with GUI.

instead collect counters to a database.

cfgFile = <a config file for offline sample>

which can be generated with the wizard.

cmNames = <small|medium|large|all>.

or a comma separated list on CMNames.

-s,--sampleTime <seconds> time between samples.

-d,--h2dbname <dbname> H2 dbname/file to store offline samples.

# Storing Performance Counter data in a Persistent Counter Storage

Performance Counters can be stored in a database, this so we can view the information at a later stage.

This module was developed so that Technical Support or other Sybase personal can view the sampled performance counters, this so they do "offline" analysis and therefore they may not need to go to the customer site.

The default storage engine used is H2 (<http://www.h2database.com>).

This is a database written in java and it's shipped with AseMon (just a small jar, with a footprint of 1-2 MB) Main advantage with this database is that it's platform independent, so customer can sample the performance counters on Unix/Linux, transfer the database file onto another platform, probably to Windows and analyze the information on that platform.

Other databases can be used to store Performance Counters, simply by changing the JDBC Driver and URL. But you will need to test if/how it works. Please send me feedback about when databases you have tried, if you had problems, send info about the problems so I can fix it ☺

Storing Performance Counters can be done in 2 ways

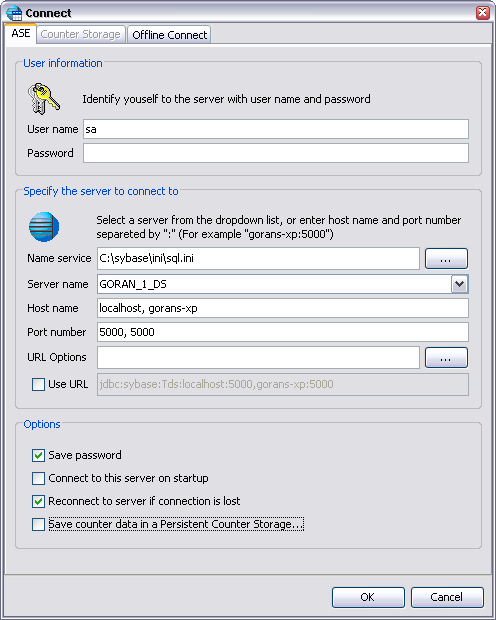
1. From the AseMon GUI itself.  
   This enables us to look at the sampled counters while it's stored in the database at the same time.
2. Start AseMon in non-gui mode and just store the counters  
   This means less overhead, but it's hard to do "online" analysis at the same time

# Viewing Performance Counter data from a Persistent Counter Storage

The GUI can connect to a Persistent Counter Storage and read it’s data, see the chapter “[Connecting to a offline storage](#_Connecting_to_a)” for more details.

# Connecting to ASE

Well we need to connect to ASE if we want to monitor it, so here is the GUI dialog for that.  
You can also use Command Line Parameters -U, -P, -S as an alternative to this dialog.



Actual URL used when connecting to ASE.  
Click the Checkbox if you want to edit/use the raw URL in this field.

JConnect URL Options, press … shows a list.

Name of the ASE server

Hostname and Port number where the ASE is located. If several “query” rows exists in the sql.ini/interfaces file, they will simply be comma separated in the host/port list.

Sql.ini or interfaces file to use as ASE Name service

Options:

**☑** Save password  
Saves the password in encrypted form in the file: ~/.asemon/asemon.save.properties

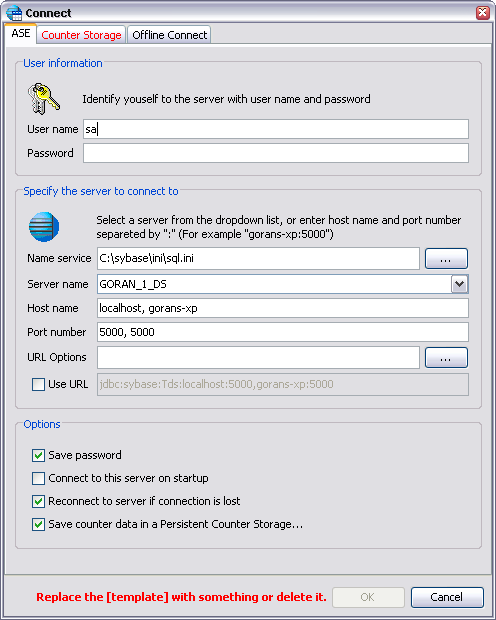
**☑** Connect to this server on startup  
Automatically connect to this server when AseMon is started.

**☑** Reconnect to server if connection is lost  
If connection to ASE server is lost due to whatever reason, try to reconnect to the server on text counter refresh.

**☑** Save counter data in a Persistent Counter Storage  
If you want to store performance counters in a database for later use.  
NOTE: if you check this box you need to specify details in the tab “Counter Storage”

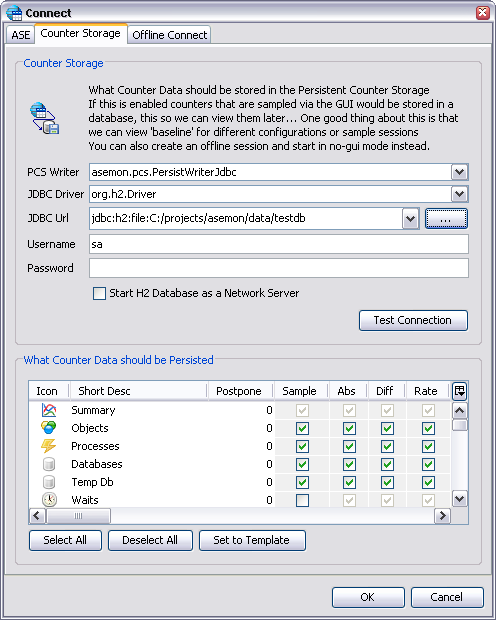
# Connecting to ASE and Storing Performance Counters in a database

This is done in the same dialog as for the ASE, just check an extra option and fill in some information.



On the “ASE” tab, use the option “Save counter data in a Persistent Counter Storage”, when choosing this option, you need to fill in some information in tab “Counter Storage”, which is marked red, because information is missing.

Complete the second tab “Counter Storage” with appropriate information.



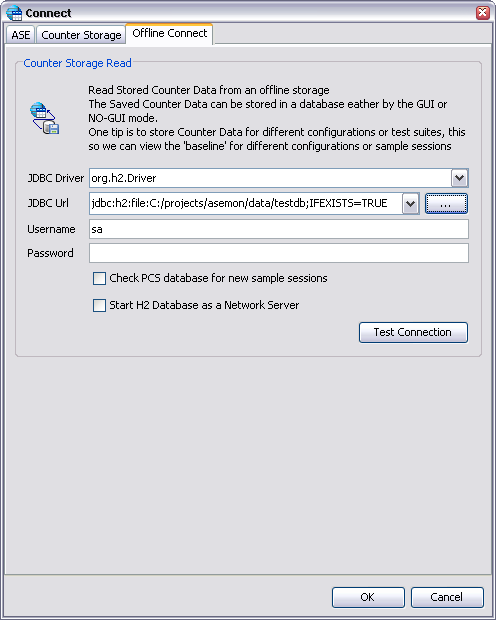
What Performance Counter Data should be stored in the database.  
In the GUI mode this can be changed during runtime.

URL examples for H2 can be found at:  
<http://www.h2database.com/html/features.html#database_url>

# Connecting to a offline storage.

To view Performance Counters that has been sampled and stored in a database.

URL examples for H2 can be found at:  
<http://www.h2database.com/html/features.html#database_url>

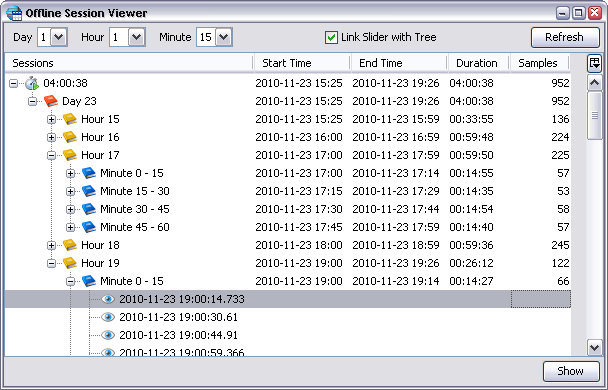


If more that one AseMon needs to connect to the database at the same time, or if you want to access the data from another tool like “DbVisualizer” or similar.

This option is not yet implemented!  
The basic idea of it is that you can watch data that is sampled by a “no gui” asemon and get notified that new samples are available.

TIP:   
Choose “jdbc:h2:file:[<path>]<dbname>;IFEXISTS=TRUE” in the “JDBC Url” ComboBox  
Then Press “…” button and locate the H2 database file, the “[<path>]<dbname>” will be replaced by the chosen filename.

After a while, the “Offline Session View” Dialog will be visible



Move “slider” se below…  
to this “point in time” to view details…

Hour Range

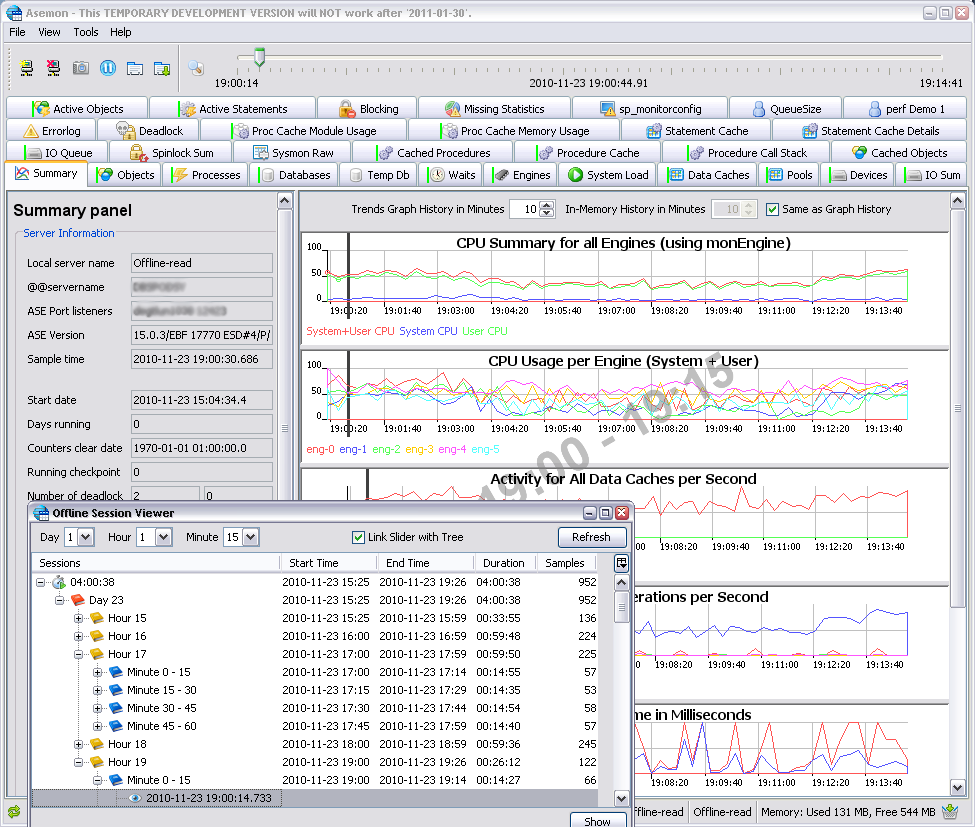
Range for the whole sample period

Minute Range

From this view select a specific “range” you want to load into AseMon, this is done by right click on the “range” and press “Show”. Then graphs for that specific range will be loaded in the Summary tab.

Current loaded “range” will be displayed as a Watermark over the Graphs, and also written on the left/right side of the slider.

Below the slider, current “point in time” are also displayed…



Set current point in time, to this time.

Number of samples within this range

Move slider to set current point in time

19:00 – 19:15   
Current range loaded

Timeline to show where we are viewing details

Double click in a graph to set current point in time

How to: Navigate between different Performance Counter Tabs:

* Simply click the Tab you are interested in, and the data will be loaded for that tab.  
  Tip: Tabs that has a “green stripe” attached to it **has** stored Performance Counters.

How to: Navigate within the selected “range”, or set “current point in time”

* Simply use the Slider at the top, on the right side of the toolbar.
* Or “double click” on any of the graphs, where you want to view details.
* Or press return/show on a specific sample in the “Offline Session View” dialog

The Graphs will have a vertical line, which indicates what counter details you are currently viewing.

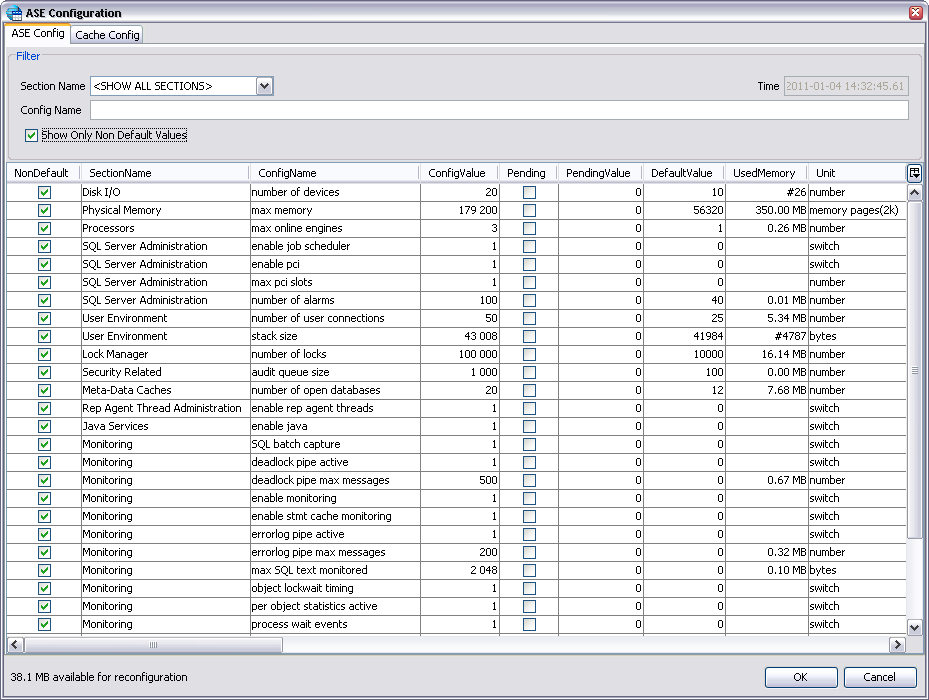
# ASE Configuration

This can be found under Menu -> View -> View ASE Configuration

When you have setup a Persistent Counter Storage, the configuration is automatically stored in the database for each sample session.  
The Configuration can be reached from the main menu, but it can also be reach by “right click” on a sample session in the Offline Session View dialog

## sp\_configure tab:

Another way to check how ASE is Configured.  
Note: More columns are available to the right…   
 Configurations that are pending on a ASE restart will be marked as red (not yet taken effect)



* Section Name:  
  Show only configuration for a specific configuration section
* Config Name  
  Filter on any configuration name
* Time  
  At what time this configuration was sampled

**☑** Show Only Non Default Values  
Only values that has been changed by someone is visible in the table.

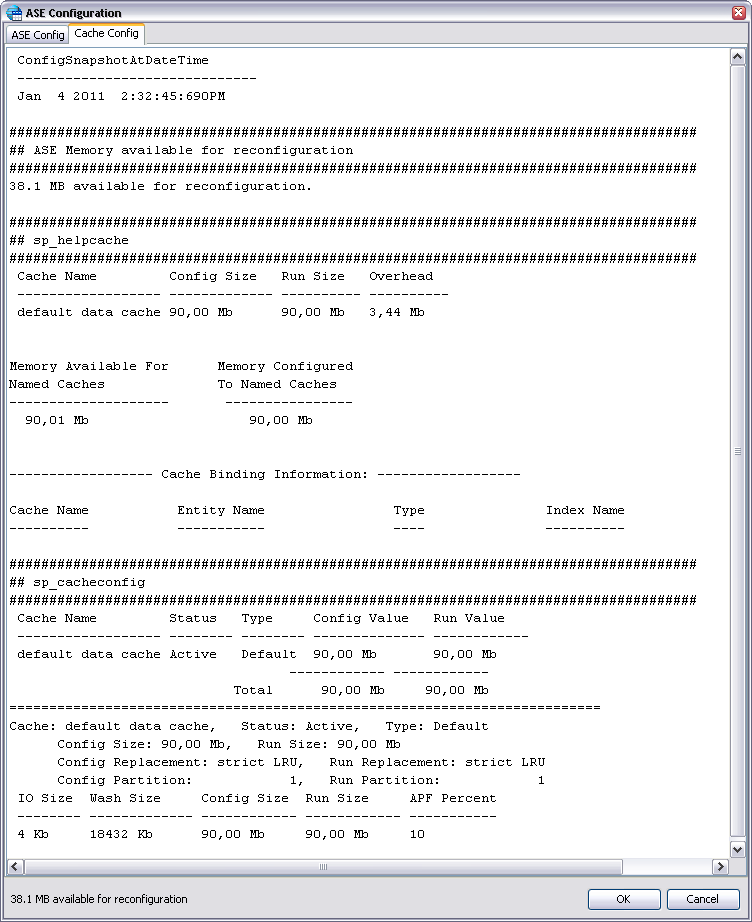
Left bottom, tells you how much memory that is available for reconfiguration.  
This is the same value as you would get from: sp\_configure ‘memory’

## sp\_cacheconfig tab:

In here I just execute: sp\_helpcache and sp\_cacheconfig

sp\_helpcache will display what objects that are bound to what caches

sp\_cacheconfig will display “cache partitions” and “pool configuration”



# A Example of how to find performance Issues

Install the “perfdemo” database

Overview of the “perfdemo” database

Overview of the “perfdemo” driver application

Goal: on my PC I was able to achieve approx 2500 “de-queue” per second.

# Overview of a Performance Counter Tab

### “right click” menu on data tables

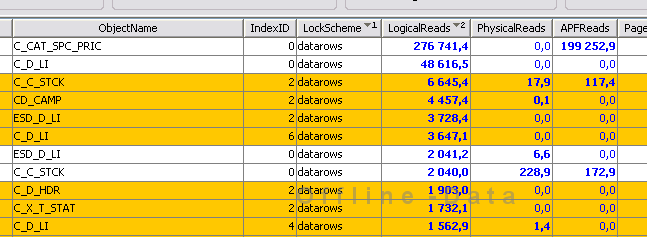
To do extra lookups in the database

Note: this is only available in the “online mode”

### Sorting on various columns

One of the basic features on all Performance Counter Collections is to Sort on various columns to find what objects are accessed most.

Primary Sort Order is here on “LockScheme”



Secondary Sort Order is on “Logical Reads”

For the moment I can have 3 levels of Sorting

Sorting will saved and restored when AseMon is restarted.  
Also the Column Order will be saved and restored, which means you can reorder the column (by drag and drop) to suit your readability.

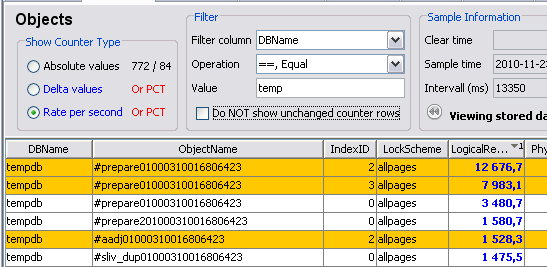
In “offline read mode” and “in-memory read mode” I will also try to restore Sort Order and Column Layout, this is excellent when you navigate between different samples.

### Filtering on various Columns and Contents

If you only interested in some specific Content in a Column, Filers is a very useful functionality.

1. Choose a specific Column to filter on
2. What type of filter do you want (= Equal, != Not Equal, > Greater Than, < Less Than)
3. Type what Content to filter on

Also a Checkbox for filter out “unmodified” counters is available.



In this example we are filtering on tables that are only in databases that is named something like “temp”

We are also sorting on LogicalReads  
Which gives us the most accesses tables in tempdb for this sample

We have 772 rows in the table  
Only 84 is visible (not filtered out)

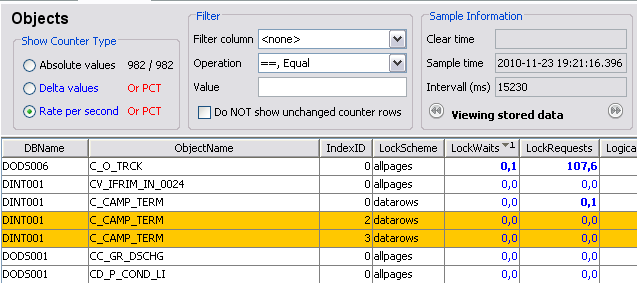
### Absolute, Delta & Rate Counters

In Counter Sets where you have specified that you want to do Delta and Rate calculation, you can switch between the different counter types.

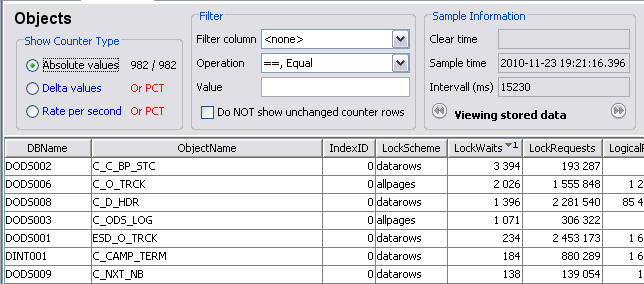
* Absolute Values  
  This is the raw counter values from the select statement that retrieves Performance Counter Data. Most MDA tables is constantly incremented.
* Delta Values  
  This is the difference between previous Counter Sample and the Latest Counter Sample  
  So if the Absolute Counter has been incremented from 100 to 210, this value would be 110.
* Rate per Second  
  This is changes per seconds  
  Formula: Delta Values / Time between the two samples  
  Looking at this value we do not have to consider if the sample interval is 5, 10 or 60 seconds, this value is reflecting the change rate instead of just the differences between two samples.

In most cases the most usable Counter Type is “Rate”, but that is just a reflection of the last sample.  
If you want to see most accessed table since ASE was rebooted you might want to look at the Absolute values instead.

Here is a example of the above:



The first screen capture is of **Rate Counters**, where we have some Lock Contention on the Object “C\_O\_TRCK”



While the next Screen Capture is of **Absolute Counters**, where we see that the table with most Lock Contention since ASE started to sample data is “C\_C\_BP\_STC”

This means that we can look at short term changes **and** also long term behavior in the same view.

### Dock and Undock of Counter Sets from the main window

If you have a big screen or multiple screen you may want to view multiple Tabs at the same time, this is done by “right click” on the tab and choose “Un Dock, show content in a window” or simply double click on the tab.

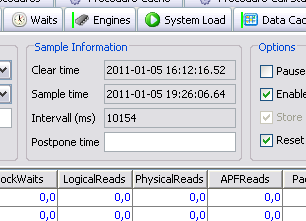
To bring it back to the main window, simply close the window or “right click” on the tab again and choose “Dock, bring back the window into the tab”

### Sample Information Panel

In this panel you will have timing information about current sample.

* Clear Time  
  This is when ASE counters was cleared (for example sp\_sysmon does this in older releases)  
  When this is done you would expect next sample to a bit inaccurate.
* Sample Time  
  When this Counter Set was actually sampled
* Interval  
  Time between previous sample and the current sample in milliseconds.
* The last field “Postpone time” is used differently depending if we are in Online, In Memory or Offline mode. See below descriptions for more information about this field.

#### Sample Information when in Online Mode



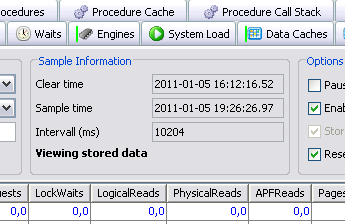
Here we can say:   
Only sample data every 10 minutes or whatever time span you think is suitable.

In the “online” mode you can specify how **often** a specific Performance Counter Set should be sampled, this is specified in seconds. But if you specify ‘10m’ it will calculate it into 600…

If this field is left blank, the counter set will be refreshed on every regular refresh.

Use this field for heavy Performance Counter Sets, that you do not want to sample that often.

#### Sample Information when in Online (in memory) Mode



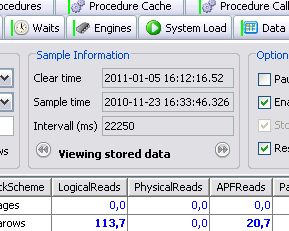
This just tells you that you are looking at historical data.

This while we are still collecting data in the online collector

If you are viewing the in-memory storage, which is a “short history” so we can go back and look at old samples while we still continue to sample data.

This is indicated that the Slider to the right of the toolbar is visible, and that the “View stored data” label is visible where the normal “postpone time” field is showed.

#### Sample Information when in Offline Mode

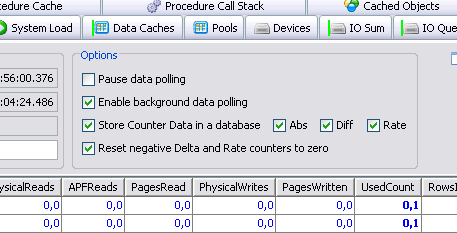


In the offline mode, you can fast forward to samples that actually consists of data.

When you are in the “offline” mode some Performance Counter Tabs doesn’t always has data, therefore you may want to fast forward to next/previous sample that has actually sampled any data.

This will be very useful in Performance Counter Tabs like ‘Active Statements’, ‘Active Objects’, ‘Procedure Call Stack’ that did not have any **active** work while the collector sampled data. (or took a snapshot of current work). Also it would be useful in ‘Spinlock Sum’ or ‘Cached Objects’ which you probably do not sample as often as the rest of the Counter Sets.

### Options on all Counter Tabs



* Pause data polling  
  Do not sample data…
* Enable background data polling  
  Even if this tab is the **active** tab, continue to sample data
* Store Counter Data in a database
  + Abs = Store Absolute Values
  + Diff = Store the Difference calculated values between two samples
  + Rate = Store the Changes Per Second calculated values between two samples
* Reset negative Delta and Rate counters to zero  
  - If a absolute counter is wrapped (goes above max value), then the difference calculation will be negative, so if it’s a negative number, simply set it to zero instead.  
  - In some cases you expect a negative counter value, and then you do not want to set to zero. One example could for instance be: if it’s a row count on a table, and you want to know if the table size decreases…

### Local Options

Some Tabs has local options.

In Process, you can choose:

* Do **not** collect data for ASE System Processes

In Active Statements, you can choose to collect extra information:

* Get SQL text from the monProcessSQLText
* Do: dbcc sqltext(spid)
* Do: sp\_showplan spid
* DO: dbcc stacktrace(spid)
* Get Procedure Call Stack from monProcessProcedures

# Performance Counter

## System Performance Counter Sets

### Summary

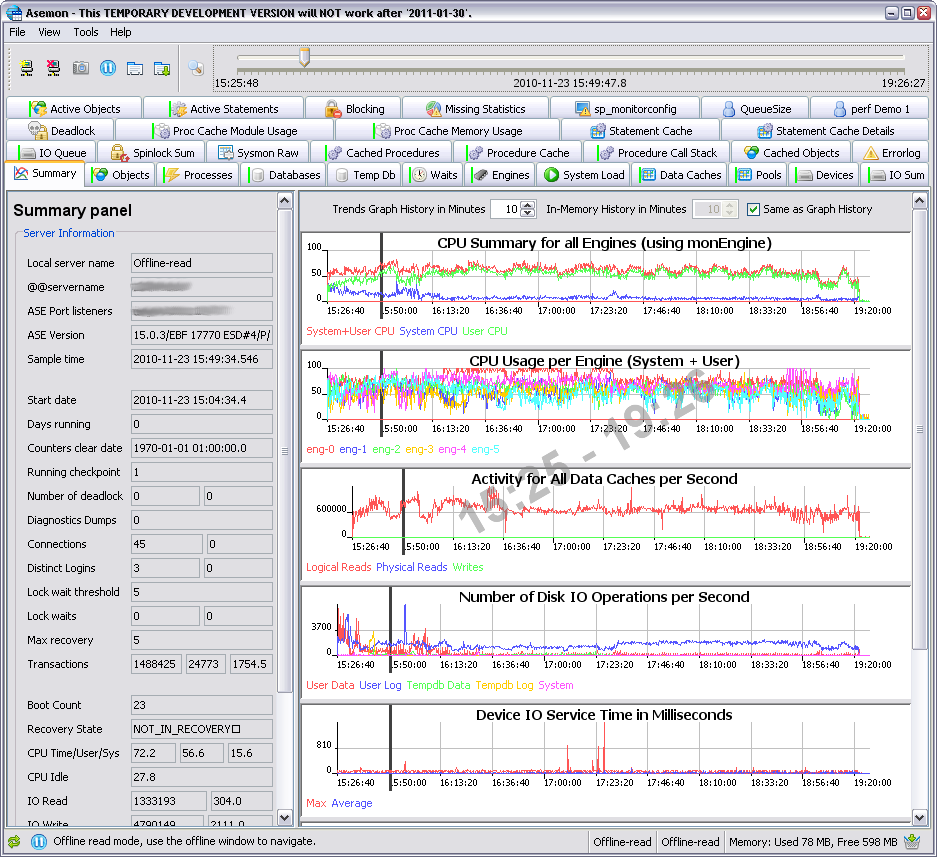
This shows some basic information in the left panel, and summary graphs in the right panel

|  |  |
| --- | --- |
| Needs ASE Version | Any version |
| Accessing tables | monState and global variables |
| Needs ASE Roles | mon\_role |
| Needs ASE Config | none |

Graphs attached to this Counter Set:

* CPU usage, based on Global Variables: @@cpu\_busy, @@cpu\_io
* Number of Transactions per Second (only from ASE 15.0.3 ESD#3)
* Number of connected users to ASE
* Disk read/write, based on Global Variables: @@total\_read, @@total\_write
* Network Packets, based on Global Variables: @@pack\_received, @@pack\_sent, @@packet\_errors

The below Screen Shot is from an “offline” database, just as an example of how the summary panel looks like.



In the case where there are multiple fields, it means that we are showing Abs, Diff & Rate values.  
Use tooltip to find out what it is

This panel contains information fetched from monState and global variables

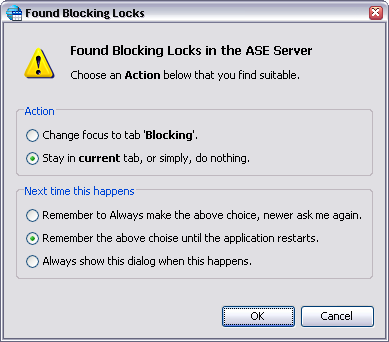
Summary Graphs from various Performance Counter Sets

See later in this document what Counter sets that has graphs attached to them

#### For the Summary Tab, there are some special functionality

#### Blocking locks functionality

If the field “Lock Waits” is indicating that there are blocking locks that has been blocking for more than 5 seconds (or above the field “Lock wait threshold”), the following dialog will be displayed

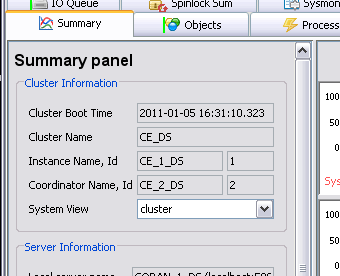


If you choose “Always make the above choice”  
Then this question will newer be asked again, so if you want to change the behavior, you need to remove the entries “ChangeToJTabDialog.Blocking.\*” from the file ~/.asemon/asemon.save.properties

Open the tab “Blocking”, this to view what objects that are currently blocking each other

#### ASE Cluster Edition specifics

When you are connected to a ASE Cluster Edition an extra Section is displaying information about what Cluster Instance you are connected to and if we should grab Performance Counters from the whole cluster or just the current Cluster Instance.



Should we collect Performance Counter from the whole Cluster or just this Cluster Instance  
The default is collect from the whole cluster

Name of the whole ASE Cluster

What Cluster Instance are we currently connected to

### Objects

|  |  |
| --- | --- |
| Needs ASE Version | Any version |
| Accessing tables | monOpenObjectActivity |
| Needs ASE Roles | mon\_role |
| Needs ASE Config | enable monitoring=1, object lockwait timing=1, per object statistics active=1 |

Graphs attached to this Counter Set:

* none

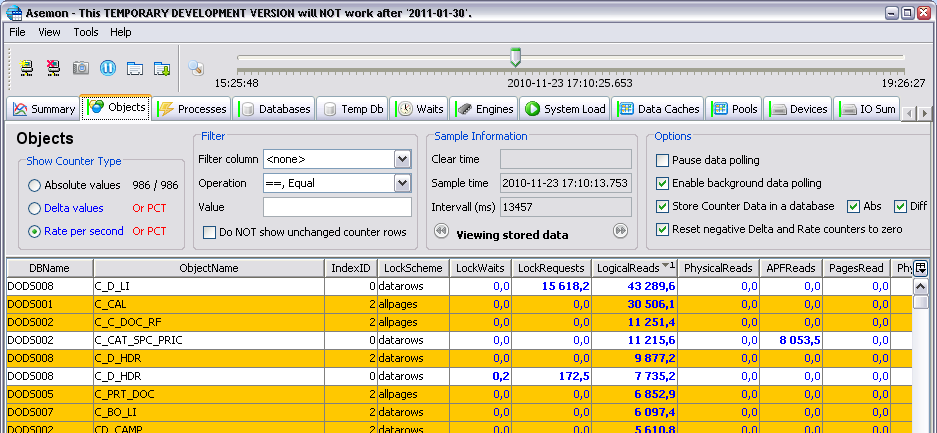
#### Usage

Main usage for this tab is

* Finding most accessed tables
  + Could be in-memory table scans
  + Table access that needs to do physical disk IO
* Finding tables that are has Locking Contentions
* Finding Insert / Update / Delete activity

#### Screen Capture

This example is sorting on most LogicalReads



#### Color Coding

* Non colored rows are data pages
* Yellow rows are index pages

### Processes

|  |  |
| --- | --- |
| Needs ASE Version | Any version |
| Accessing tables | monProcessActivity, monProcess, sysprocesses, monProcessNetIO |
| Needs ASE Roles | mon\_role |
| Needs ASE Config | enable monitoring=1, object lockwait timing=1, wait event timing=1 |

Graphs attached to this Counter Set:

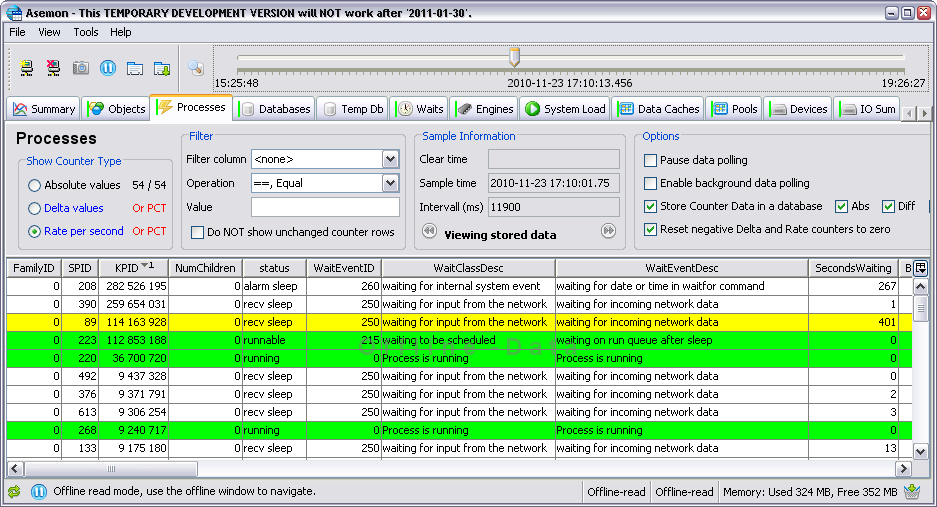
* Checkpoint and Housekeeper Writes Per Second

#### Usage

Main usage for this tab is

* Finding specific SPID’s / Users that are heavy users of the system.
* Finding which SPID’s are waiting for various events or recourses
* A Tip could be to filter on rows where WaitEventID != 250 (waiting for client to send data)

#### Screen Capture



#### Color Coding

* Green rows are SPID’s that are in status RUNNING or RUNNABLE  
  Either they are executing on an Engine (running), or we are waiting to be scheduled (runnable)
* Yellow rows are ASE System SPID’s
* Pink rows are SPID’s that are blocked by a lock (blocked by some other SPID that holds the lock)
* Red rows are SPID’s that are blocking **other** SPID’s from executing (root cause of a block)

### Databases

|  |  |
| --- | --- |
| Needs ASE Version | Any version |
| Accessing tables | monOpenDatabases |
| Needs ASE Roles | mon\_role |
| Needs ASE Config | enable monitoring=1 |

Graphs attached to this Counter Set:

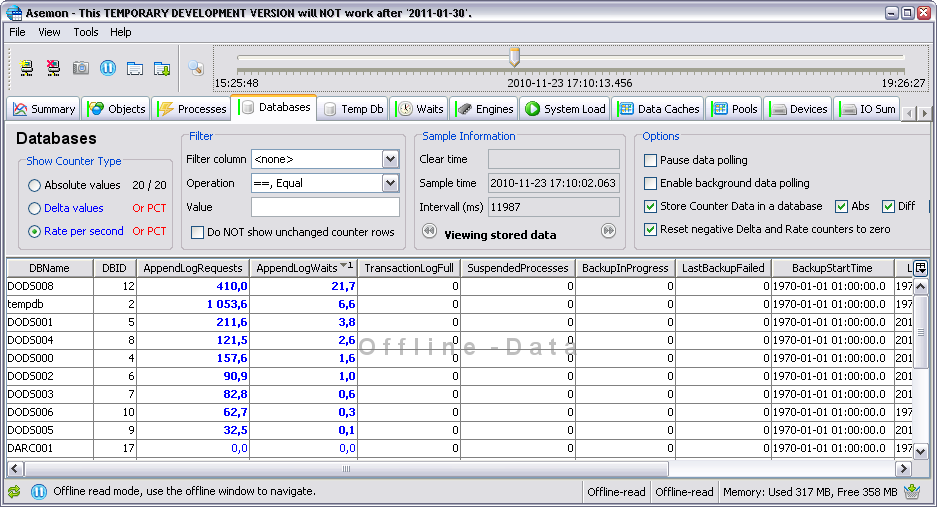
* none

#### Usage

Main usage for this tab is

* Finding out how many log request is done to a specific database
* Finding out Contention on the Log Semaphore
* Finding out if a database has a full transaction log and how many SPID’s are waiting for that

#### Screen Capture



#### Color Coding

* None

### Temp Db

|  |  |
| --- | --- |
| Needs ASE Version | 15.5 (15.0.2 if Cluster Edition) |
| Accessing tables | monTempdbActivity |
| Needs ASE Roles | mon\_role |
| Needs ASE Config | enable monitoring=1, object lockwait timing=1, per object statistics active=1 |

Graphs attached to this Counter Set:

* none

#### Usage

Main usage for this tab is

* xxx

#### Screen Capture

#### Color Coding

* None

### Waits

|  |  |
| --- | --- |
| Needs ASE Version | Any version |
| Accessing tables | monSysWaits, monWaitEventInfo, monWaitClassInfo |
| Needs ASE Roles | mon\_role |
| Needs ASE Config | enable monitoring=1, wait event timing=1 |

Graphs attached to this Counter Set:

* none

#### Usage

Main usage for this tab is

* xxx

#### Screen Capture

#### Color Coding

* None

### Engines

|  |  |
| --- | --- |
| Needs ASE Version | Any version |
| Accessing tables | monEngine |
| Needs ASE Roles | mon\_role |
| Needs ASE Config | enable monitoring=1 |

Graphs attached to this Counter Set:

* CPU Summary for all Engines (using monEngine)
* CPU Usage per Engine (System + User)  
  (If this is Cluster Edition, the graph will contain all engines on all cluster instances)

#### Usage

Main usage for this tab is

* xxx

#### Screen Capture

#### Color Coding

* None

### System Load

|  |  |
| --- | --- |
| Needs ASE Version | 15.5 (15.0.2 if Cluster Edition) |
| Accessing tables | monSysLoad |
| Needs ASE Roles | mon\_role |
| Needs ASE Config | none |

Graphs attached to this Counter Set:

* none

#### Usage

Main usage for this tab is

* xxx

#### Screen Capture

#### Color Coding

* None

### Data Caches

|  |  |
| --- | --- |
| Needs ASE Version | Any version |
| Accessing tables | monDataCache |
| Needs ASE Roles | mon\_role |
| Needs ASE Config | enable monitoring=1 |

Graphs attached to this Counter Set:

* Activity for All Data Caches per Second

#### Usage

Main usage for this tab is

* xxx

#### Screen Capture

#### Color Coding

* None

### Pools

|  |  |
| --- | --- |
| Needs ASE Version | Any version |
| Accessing tables | monCachePool |
| Needs ASE Roles | mon\_role |
| Needs ASE Config | enable monitoring=1 |

Graphs attached to this Counter Set:

* none

#### Usage

Main usage for this tab is

* xxx

#### Screen Capture

#### Color Coding

* None

### Devices

|  |  |
| --- | --- |
| Needs ASE Version | Any version |
| Accessing tables | monDeviceIO |
| Needs ASE Roles | mon\_role |
| Needs ASE Config | enable monitoring=1 |

Graphs attached to this Counter Set:

* none

#### Usage

Main usage for this tab is

* xxx

#### Screen Capture

#### Color Coding

* None

### IO Sum

|  |  |
| --- | --- |
| Needs ASE Version | Any version |
| Accessing tables | monIOQueue |
| Needs ASE Roles | mon\_role |
| Needs ASE Config | enable monitoring=1 |

Graphs attached to this Counter Set:

* Number of Disk IO Operations per Second

#### Usage

Main usage for this tab is

* xxx

#### Screen Capture

#### Color Coding

* None

### IO Queue

|  |  |
| --- | --- |
| Needs ASE Version | Any version |
| Accessing tables | monIOQueue |
| Needs ASE Roles | mon\_role |
| Needs ASE Config | enable monitoring=1 |

Graphs attached to this Counter Set:

* Device IO Service Time in Milliseconds

#### Usage

Main usage for this tab is

* xxx

#### Screen Capture

#### Color Coding

* None

### Spinlock Sum

|  |  |
| --- | --- |
| Needs ASE Version | Any version |
| Accessing tables | sysmonitors |
| Needs ASE Roles | sa\_role |
| Needs ASE Config | none |

Graphs attached to this Counter Set:

* none

#### Usage

Main usage for this tab is

* xxx

#### Screen Capture

#### Color Coding

* None

### Sysmon Raw

|  |  |
| --- | --- |
| Needs ASE Version | Any version |
| Accessing tables | sysmonitors |
| Needs ASE Roles | sa\_role |
| Needs ASE Config | none |

The Counter Set depends on:

* Spinlock Sum

Graphs attached to this Counter Set:

* none

#### Usage

Main usage for this tab is

* xxx

#### Screen Capture

#### Color Coding

* None

### Cached Procedures

|  |  |
| --- | --- |
| Needs ASE Version | Any version |
| Accessing tables | monCachedProcedures |
| Needs ASE Roles | mon\_role |
| Needs ASE Config | per object statistics active=1 |

Graphs attached to this Counter Set:

* none

#### Usage

Main usage for this tab is

* xxx

#### Screen Capture

#### Color Coding

* None

### Procedure Cache

|  |  |
| --- | --- |
| Needs ASE Version | Any version |
| Accessing tables | monProcedureCache |
| Needs ASE Roles | mon\_role |
| Needs ASE Config | enable monitoring=1 |

Graphs attached to this Counter Set:

* Number of Procedure Requests per Second (procs,triggers,views)

#### Usage

Main usage for this tab is

* xxx

#### Screen Capture

#### Color Coding

* None

### Procedure Call Stack

|  |  |
| --- | --- |
| Needs ASE Version | Any version |
| Accessing tables | monProcessProcedures |
| Needs ASE Roles | mon\_role |
| Needs ASE Config | none |

Graphs attached to this Counter Set:

* none

#### Usage

Main usage for this tab is

* xxx

#### Screen Capture

#### Color Coding

* None

### Cached Objects

|  |  |
| --- | --- |
| Needs ASE Version | Any version |
| Accessing tables | monCachedObject |
| Needs ASE Roles | mon\_role |
| Needs ASE Config | none |

Graphs attached to this Counter Set:

* none

#### Usage

Main usage for this tab is

* xxx

#### Screen Capture

#### Color Coding

* None

### Errorlog

|  |  |
| --- | --- |
| Needs ASE Version | Any version |
| Accessing tables | monErrorLog |
| Needs ASE Roles | mon\_role |
| Needs ASE Config | enable monitoring=1, errorlog pipe active=1, errorlog pipe max=200 |

Graphs attached to this Counter Set:

* none

Note: This counters can’t be stored to a Persistent Counter Store

#### Usage

Main usage for this tab is

* xxx

#### Screen Capture

#### Color Coding

* None

### Deadlock

|  |  |
| --- | --- |
| Needs ASE Version | Any version |
| Accessing tables | monDeadLock |
| Needs ASE Roles | mon\_role |
| Needs ASE Config | enable monitoring=1, deadlock pipe active=1, deadlock pipe max=500 |

Graphs attached to this Counter Set:

* none

Note: This counters can’t be stored to a Persistent Counter Store

#### Usage

Main usage for this tab is

* xxx

#### Screen Capture

#### Color Coding

* None

### Proc Cache Module Usage

|  |  |
| --- | --- |
| Needs ASE Version | 15.0.1 |
| Accessing tables | monProcedureCacheModuleUsage |
| Needs ASE Roles | mon\_role |
| Needs ASE Config | none |

Graphs attached to this Counter Set:

* none

#### Usage

Main usage for this tab is

* xxx

#### Screen Capture

#### Color Coding

* None

### Proc Cache Memory Usage

|  |  |
| --- | --- |
| Needs ASE Version | 15.0.1 |
| Accessing tables | monProcedureCacheMemoryUsage |
| Needs ASE Roles | mon\_role |
| Needs ASE Config | none |

Graphs attached to this Counter Set:

* none

#### Usage

Main usage for this tab is

* xxx

#### Screen Capture

#### Color Coding

* None

### Statement Cache

|  |  |
| --- | --- |
| Needs ASE Version | 15.0.2 |
| Accessing tables | monStatementCache |
| Needs ASE Roles | mon\_role |
| Needs ASE Config | enable monitoring=1, enable stmt cache monitoring=1, statement cache size" |

Graphs attached to this Counter Set:

* none

#### Usage

Main usage for this tab is

* xxx

#### Screen Capture

#### Color Coding

* None

### Statement Cache Details

|  |  |
| --- | --- |
| Needs ASE Version | 15.0.2 |
| Accessing tables | monCachedStatement |
| Needs ASE Roles | mon\_role |
| Needs ASE Config | enable monitoring=1, enable stmt cache monitoring=1, statement cache size |

Graphs attached to this Counter Set:

* none

#### Usage

Main usage for this tab is

* xxx

#### Screen Capture

#### Color Coding

* None

### Active Objects

|  |  |
| --- | --- |
| Needs ASE Version | Any version |
| Accessing tables | monProcessObject |
| Needs ASE Roles | mon\_role |
| Needs ASE Config | enable monitoring=1, per object statistics active=1 |

Graphs attached to this Counter Set:

* none

#### Usage

Main usage for this tab is

* xxx

#### Screen Capture

#### Color Coding

* None

### Active Statements

|  |  |
| --- | --- |
| Needs ASE Version | Any version |
| Accessing tables | monProcessStatement, monProcess, sysprocesses |
| Needs ASE Roles | mon\_role |
| Needs ASE Config | enable monitoring=1, statement statistics active=1, wait event timing=1 |

Graphs attached to this Counter Set:

* none

#### Usage

Main usage for this tab is

* xxx

#### Screen Capture

#### Color Coding

* None

### Blocking

|  |  |
| --- | --- |
| Needs ASE Version | 15.0.2 ESD#2 |
| Accessing tables | monLocks |
| Needs ASE Roles | mon\_role |
| Needs ASE Config | enable monitoring=1, wait event timing=1 |

Graphs attached to this Counter Set:

* none

#### Usage

Main usage for this tab is

* xxx

#### Screen Capture

#### Color Coding

* None

### Missing Statistics

|  |  |
| --- | --- |
| Needs ASE Version | 15.0.3 ESD#1 |
| Accessing tables | sysstatistics in each database |
| Needs ASE Roles | Runtime = None, During install = sa\_role |
| Needs ASE Config | capture missing statistics |

Graphs attached to this Counter Set:

* none

#### Usage

Main usage for this tab is

* xxx

#### Screen Capture

#### Color Coding

* None

### sp\_monitorconfig

|  |  |
| --- | --- |
| Needs ASE Version | Any version |
| Accessing tables | sp\_monitorconfig system procedure |
| Needs ASE Roles | none |
| Needs ASE Config | none |

Graphs attached to this Counter Set:

* none

Note: default postpone time is set to 600 (sample this every 10 minutes)

#### Usage

Main usage for this tab is

* xxx

#### Screen Capture

#### Color Coding

* None

## User Defined Counter Sets

If the existing System Counter Sets doesn’t meet your needs, you can simply create your own Counter Collectors, which I call “User Defined Counters”. The easiest way to do this is to use the wizard “Create ‘User Defined Counter’ wizard”, which can be found under the Tools menu. Or you can add entries manually to the configuration file.

## Using the ‘User Defined Counter’ Wizard

## Adding entries manually to the configuration file

# Reading ‘offline data’ using the GUI

If you have a database with captured monitoring data, you can connect to that “offline database” and read the content using the ordinary AseMon GUI.