Version 2.4.7.0

- Fix: DataFileDelimiter Is Blank in Initialization Event Message When Tab Is Selected.

  Actually a tab character is written but appears as one or more spaces. The collection agent was changed to emit HTAB when the DataFileDelimiter is a tab.

- Enhancement: Added CPU Configuration Records.

  CPU Configuration Records relate logical processor to CPU in Hyper-Threaded environments and provide CPU configuration details, including Hyper-Threading utilization.

  For details please see “CPU Config Record Support FAQ Including HyperThreading.doc”.

- Fix: Multiple Stop Event Messages In The Event Log.

  This was due to failing to realize that the collection agent state was, already, trying to stop.

- Enhancement: Cycle Data Path Written To The Registry.

  At the start of each cycle, the Data Path is written into the Registry in a value named ActiveDataPath at the key HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\DMPerfss\Control.

- Fix: Performances Libraries That Are Incorrectly Installed And Excluded Fail To Show Up in the Exclusion Event Message.

  Performance SeNTry was changed so that the Exclusion List has precedence over Performance Library analysis. Formerly, only valid Performance Libraries were in the Exclusion List event message, although they were being excluded, anyway.

- Fix: Instance Names Contains Invalid Characters.

  The cause is unknown but instance name analysis was changed to expect more variations to the Microsoft Performance Counter Specification.

- Fix: Collector Stops with Win32 Exception When Processor Object Fails to Collect.

  An event message, similar to the following, may be observed in the event log.

  07/25/04-04:55:26 - Event ID: 2200, Category: Collection, Severity: Warning
  Object, Processor (L238, G238), failed to return data this interval

  This will trigger a Win32 exception similar to the following one. Examining the stack trace in the Win32 exception for one that is similar to the following indicates this problem.

  07/25/04-04:55:55 - Event ID: 97, Category: N/A, Severity: Error
  Win32 exception, 0xC0000005, encountered at location 0x0045614F
  Read attempt at location 0x0000000C
  Module List:
  <module list removed for brevity>
  Stack Trace (ThreadId 880):
  dmperfss.exe: PSSPerfObject::BufferInfo::FindPreviousObjectInstanceCounter + 84 bytes

E:\DmPerfss 2.4.6\Collector\PSSPerfObject.cpp, Line 2717 + 6 bytes
dmperfss.exe: PSSPerfObject::BufferInfo::InitForInstance + 57 bytes
E:\DmPerfss 2.4.6\Collector\PSSPerfObject.cpp, Line 2916 + 7 bytes
dmperfss.exe: PSSPerfLocalCounter::Evaluate + 106 bytes
E:\DmPerfss 2.4.6\Collector\PSSPerfLocalCounter.cpp, Line 518
dmperfss.exe: PSSWin2KSystemObject::PseudoCounter::Evaluate + 274 bytes
E:\DmPerfss 2.4.6\Collector\PSSWin2KSystemObject.cpp, Line 513 + 61 bytes
dmperfss.exe: PSSPerfLocalCounter::GetValue + 22 bytes
E:\DmPerfss 2.4.6\Collector\PSSPerfLocalCounter.h, Line 294
dmperfss.exe: PSSSmfOStream::FormatCounters + 151 bytes
E:\DmPerfss 2.4.6\Collector\PSSSmfOStream.cpp, Line 1567 + 8 bytes
dmperfss.exe: PSSSmfOStream::FormatObject + 142 bytes
E:\DmPerfss 2.4.6\Collector\PSSSmfOStream.cpp, Line 1618 + 58 bytes
dmperfss.exe: PSSPerfObject::PutIntervalRecord + 217 bytes
E:\DmPerfss 2.4.6\Collector\PSSPerfObject.cpp, Line 2154
dmperfss.exe: PSSReportable::PutIntervalRecord + 39 bytes
E:\DmPerfss 2.4.6\Collector\PSSReportable.cpp, Line 191
dmperfss.exe: PSSSmfOStream::WriteIntervalSection + 88 bytes
E:\DmPerfss 2.4.6\Collector\PSSSmfOStream.cpp, Line 1865 + 7 bytes
<rest of Stack Trace removed for brevity>

- Notes:
  - No longer runs on Windows NT 3.51 and prior.
  - Requires DbgHelp.dll version 5.12600.0.
  - Requires msvcp71.dll, msvcr71.dll.

Version 2.4.6.5

- Fix: The OnErrorCommand May Not Run When The Collector Is Stopped.

  The collector notifies the Service Control Manager (SCM) that it is has stopped before the OnErrorCommand. Because of timing the SCM may terminate the thread before launching the OnErrorCommand.

  This was fixed to notify the SCM as late as possible.

- Fix: The Collector Hangs Without Logging The Reason.

  On Windows 2000 and above, and event log message in the System Event Log may indicate a Performance SeNTry failure from the Service Control Manager event source such as the following:

  Event Type: Error
  Event Source: Service Control Manager
  Event Category: None
  Event ID: 7031
  Date: 4/14/2004
  Time: 4:47:43 PM
  User: N/A
  Computer: *<computer name>*
  Description:
  The Performance SeNTry service terminated unexpectedly. It has done this 1 time(s).
  The following corrective action will be taken in 0 milliseconds: No action.

This occurs when the collector abnormally terminates very early during initialization after the service control manager (SCM) has started the collector's main thread but before it tells the SMC that it is alive.

The termination message is not logged unless the collector suspends or stops, neither of which was occurred, so no message is present. Although the collector threads terminate, the main service thread does not because the collector has failed to notify the SCM that it stopped, leaving the process running with a single thread, which, typically cannot be killed by the task manager.

This issue was fixed by notifying the SCM that collection agent is stopping when it terminates for any reason.

- Fix: The Collector Terminates Without Logging The Reason.

  This occurs when an exception occurs in a thread created by a Performance Library Thread.

  The collection agent creates threads to trap exceptions from misbehaving Performance Libraries. However, a Performance Library could, in turn, create a thread. Exceptions were not being trapped in these Performance Library created threads causing the process to immediately abort.

  This problem was fixed by trapping exceptions in Performance Library created threads.

- Fix: Performance Libraries Excluded by the Configuration File are Reported With an Warning Severity.

  The severity was changed to informational.

- Fix: Counter “Process:Ready Threads” Fails To Evaluate After Collection Period Start.

  A null field, for this counter, is written in the Process data record. Unless the "Thread State Ready" filter is active, no event message is logged.

  The following event message indicates the problem, when the "Thread State Ready" filter is active.

  05/06/04-21:03:14 - Event ID: 1700, Category: Filter, Severity: Warning
  In filter "Thread State Ready", the counter G230:(L5842, G905842) failed to evaluate.

Version 2.4.6.4

- Fix: Data for Print Queue (L1450) Objects Only Present In The First Cycle After Start.

  The following event messages indicate the problem.

  05/03/04-15:29:01 - Event ID: 1309, Category: PerfData, Severity: Warning
  Print Spooler (Global): no objects were found in the performance data buffer.

  05/03/04-15:30:02 - Event ID: 2205, Category: Collection, Severity: Warning
  The following objects were specified in the DCS but failed to return data.
  These objects will be skipped this cycle.
  Print Queue (L1450, G1450)

Version 2.4.6.3

- Fix: The Collector Fails During Resume After a Win32 Exception in the Main Thread.

  This was observed after a Win32 exception was reported in the PerfDisk.dll collection thread. The Win32 Exception in the Main Thread causes a corruption in the collection agent that persists through the rest of the lifespan of the collection agent.

  Performance Libraries are collected in individual threads to isolate problems. A problem is the main thread is more serious than in a collection thread because it may impact the entire service. Also, some enterprise operations may notice a stopped service but not a suspended one. Starting with Windows 2000, the OS automatically manages services that unexpectedly stop.

  To facilitate a higher probability of re-starting the collector successfully and, also, as an indicator of an operational problem to service management applications, the collection agent is now stopped, instead of suspended, when a Win32 Exception is encountered in the Main Thread.

Version 2.4.6.2

- Fix: Collector Stops Suddenly Without Indication

This occurs on Windows 2000 with object SMTP (G11900) in the DCS. No event messages are emitted.

- Fix: Installation Emits Vague Message When Insufficient Account Privileges Are Used

The error message was changed to contain the failure reason as well as the operational consequences.

Version 2.4.6.1

- Fix: Collector Stops with Win32 Exception When a Perflib Times Outs during a Close.

The collector stops with a Win32 exception when a close operation on a perflib fails to complete in the allotted time. This is followed by a message saying the thread associated with the perflib could not be destroyed. This is followed by a Win32 exception similar to the following.

This problem can be recognized by examining the stack in the Win32 exception.

10/11/03-03:00:16 - Event ID: 97, Category: N/A, Severity: Error
Win32 exception, 0xC0000005, encountered at location 0x0041C332
Read attempt at location 0x00000010
Module List:
<module list removed for brevity>
Stack Trace (ThreadId 3504):
DmPerfss.exe: PSSCollectionThread::OnFinalRelease + 36 bytes
    E:\DmPerfss 2.4.6\Collector\PSSCollectionThread.cpp, Line 436 + 11 bytes
DTSFnd.DLL: DTSShare::Release + 24 bytes
    E:\DmPerfss 2.4.6\Foundation\DTSShare.h, Line 237 + 7 bytes
DmPerfss.exe: PSSPerfDataSource::ThreadAutoPtr_t::Release + 36 bytes
    E:\DmPerfss 2.4.6\Collector\PSSPerfDataSource.cpp, Line 195 + 12 bytes
DmPerfss.exe: PSSPerfDataSource::Unrealize + 55 bytes

E:\DmPerfss 2.4.6\Collector\PSSPerfDataSource.cpp, Line 378 + 16 bytes
DmPerfss.exe: PSSPerformanceLibrary::Unrealize + 264 bytes
E:\DmPerfss 2.4.6\Collector\PSSPerformanceLibrary.cpp, Line 1411 + 15 bytes
DmPerfss.exe: PSSPerformanceLibrary::OnFinalRelease + 22 bytes
E:\DmPerfss 2.4.6\Collector\PSSPerformanceLibrary.cpp, Line 2276
DTSFnd.DLL: DTSShare::Release + 24 bytes
E:\DmPerfss 2.4.6\Foundation\DTSShare.h, Line 237 + 7 bytes
DmPerfss.exe: DTSAutoShare<PSSPerfDataSource,PSSPerfDataSource *>::Release + 30 bytes
E:\DmPerfss 2.4.6\Collector\PSSPerfObjectBase.h, Line 473 + 6 bytes
DmPerfss.exe: PSSPerfObject::Unrealize + 26 bytes
E:\DmPerfss 2.4.6\Collector\PSSPerfObject.cpp, Line 1429
DmPerfss.exe: PSSPerfObject::~PSSPerfObject + 102 bytes
E:\DmPerfss 2.4.6\Collector\PSSPerfObject.cpp, Line 873
DmPerfss.exe: PSSPerfObject::`vector deleting destructor' + 21 bytes
..\Foundation\DTSIndexFwdIterator.h, Line 307 + 35 bytes
DmPerfss.exe: DTSConstPtrIterator<DTSList<PSSPerfObjectBase
*>,PSSPerfObjectBase>::Destroy + 25 bytes
..\Foundation\DTSPtrContainer.h, Line 208 + 25 bytes
DmPerfss.exe: PSSPtrList<PSSPerfObjectBase>::erase + 34 bytes
E:\DmPerfss 2.4.6\Collector\PSSList.h, Line 411
DmPerfss.exe: PSSDataSet::erase + 62 bytes
E:\DmPerfss 2.4.6\Collector\PSSDataSet.h, Line 418 + 24 bytes
DmPerfss.exe: PSSDataSet::clear + 35 bytes
E:\DmPerfss 2.4.6\Collector\PSSDataSet.cpp, Line 550
DmPerfss.exe: PSSPerformanceSeNTry::StopCollection + 241 bytes
E:\DmPerfss 2.4.6\Collector\PSSPerformanceSeNTry.cpp, Line 2128
DmPerfss.exe: PSSPerformanceSeNTry::DoEventAction + 1061 bytes
E:\DmPerfss 2.4.6\Collector\PSSPerformanceSeNTry.cpp, Line 2460 + 10 bytes
DmPerfss.exe: PSSCollectionCycle::OnTransition + 414 bytes
E:\DmPerfss 2.4.6\Collector\PSSCollectionCycle.cpp, Line 3721
DmPerfss.exe: DTSObjectCallback<PSSCollectionCycle,DTSXTimer<DTSLocalTime>
*,int>::Invoke + 25 bytes
..\Foundation\DTSCallback.h, Line 310 + 10 bytes
DmPerfss.exe: DTSXTimer<DTSLocalTime>::Invoke + 11 bytes
..\Foundation\DTSTimer.h, Line 231 + 11 bytes
DTSFnd.DLL: DTSDispatcher::ProcessExpiredTimers + 167 bytes
E:\DmPerfss 2.4.6\Foundation\DTSDispatcher.cpp, Line 341 + 5 bytes
DTSFnd.DLL: DTSDispatcher::DispatchEvent + 102 bytes
E:\DmPerfss 2.4.6\Foundation\DTSDispatcher.cpp, Line 464 + 7 bytes
DmPerfss.exe: PSSDispatcher::DispatchEvent + 74 bytes
E:\DmPerfss 2.4.6\Collector\PSSDispatcher.cpp, Line 90 + 8 bytes
DTSFnd.DLL: DTSDispatcher::Run + 36 bytes
E:\DmPerfss 2.4.6\Foundation\DTSDispatcher.cpp, Line 482
DmPerfss.exe: PSSPerformanceSeNTry::DispatchEvents + 68 bytes
E:\DmPerfss 2.4.6\Collector\PSSPerformanceSeNTry.cpp, Line 2262 + 21 bytes
DmPerfss.exe: PSSPerformanceSeNTry::Run + 122 bytes
E:\DmPerfss 2.4.6\Collector\PSSPerformanceSeNTry.cpp, Line 1650 + 7 bytes
DmPerfss.exe: DataServiceMain + 70 bytes
E:\DmPerfss 2.4.6\Collector\dmperfss.c, Line 905
ADVAPI32.dll: CryptVerifySignatureA + 162 bytes
The system error was "The instruction at "0x (0xC0000005)"

Version 2.4.6

- Fix: Synthetic Physical and Logical Disk Counters May Not Report Zero.

  The following synthetic counters appear in the data records of Logical and Physical Disk Objects.

  - % Disk Busy
  - Avg Disk Service Time
  - Avg Disk Queue Time

  The disk performance library sometimes returns a value of more than 100% for % Disk idle Time that is used to compute the synthetic counters. This was causing the computation a very small number instead of zero, which is correct.

  The formula was changed to account for % Disk Idle Time > 100% and now correctly writes zero.

- Fix: Duplicate Network Instance Names Only Produce Valid Data for First Instance

  Duplicate Network Interface Instances are discriminated on output by appending #<n>, where <n> is a number from 1 to the number of duplicates. When evaluating counters, the difference between the current and previous data is calculated. For each duplicate current instance, only the first previous instance is used in the difference. This results in only correct counter data for the first duplicate instance.

- Fix: Lotus Notes Fails To Collect

  The Lotus Notes Object was broken in a prior release and has been fixed.

- Fix: Non-Displayable Characters Appear In Instance Names In SMF File

  Non-Displayable characters in instance names are converted to blanks when they are written to the SMF file.

- Fix: Memory Leak When Win32 Exception Stack Trace Event Is Logged On Windows Server 2003.

  The following is an example Stack Trace.

  01/07/03-09:36:20 - Event ID: 97, Category: N/A, Severity: Error
  Win32 exception, 0xC0000005, encountered at location <*address*>
  Read attempt at location <*address*>
  Module List:
  [Module List Removed]
  Stack Trace (ThreadId <*id*>):
  MSVCRT.dll: memcmp + <*n*> bytes
  DMPerfss.exe: PSSDataSet::insert + <*n*> bytes
  E:\DmPerfss 2.4.3.3\Collector\PSSDataSet.cpp, Line 1065 + <*n*>bytes
  DMPerfss.exe: PSSFileDCSArchive::ReadDataSet + 507 bytes
  E:\DmPerfss 2.4.3.3\Collector\PSSFileDCSArchive.cpp, Line 1017 + <*n*> bytes
  [rest of list removed]

- Fix: Account With No Password Is Not Used.

  The collector did not remember the account parameter without a password.

- Enhancement: The Log File Is Not Age Cycled When The Collector Is Suspended

  If the collector is suspended, the log file is cycled according to size but not age, to permit review of the log should a large amount of time pass while suspended. Formerly, the logs were cycled, leading to a possible loss of information.

- Enhancement: Reporting Account Authentication Failure Message On Installation

  The collector will issue a message to the screen if it installed with an account that fails to authenticate. This will not affect the installation. However, the collector will default to the service's account (LocalSystem or SYSTEM) during operation.

- Enhancement: Exclude List and Configuration File.

  An, optional, configuration file can be used to indicate performance libraries to exclude or include. The file is called DmPerfss.cfg and must reside in the same directory as DmPerfss.exe. This file is shipped with a default setting to exclude the following performance libraries, which have exhibited problems on Windows 2003 Server. The exclusion is effective on all versions of Windows.

  - netfxperf.dll, version 1.0.3215.11 (.NET CLR Data, NET CLR Networking objects)
  - mscoree.dll, version 1.0.3215.11 (.NETFramework objects)
  - NTFRSPRF.dll, version 5.1.3590.0 (FileReplicaConn, FileReplicaSet objects)
  - wmiaprpl.dll, version 5.1.3590.0 (WmiApRpl objects)

- NOT Fixed: Pattern Objects Used In A Filter Produces NO Data Records.

  Pattern objects are those that have patterns (wildcards: '*', '?') in the OMI. Currently only two are defined:

  - SNA Adapter SnaDlc*
  - MSSQL$*:xxx

  This problem has NOT been fixed. It exists only in version 2.4.6 and later. It was exposed when the problem "Collection Agent Incorrectly Reports Extensible Counters And/Or Objects Are Not In The Dcs When Used In A Filter" was fixed because pattern objects are extensible. It will be fixed in a later version.

- Fix: Collection Agent Incorrectly Reports Extensible Counters And/Or Objects Are Not In The Dcs When Used In A Filter.

  The following are examples of event messages that indicate this bug:

  Event Type: Warning
  Event Source: DMPerfss
  Event Category: DCS
  Event ID: 609

Date: 5/27/2003
Time: 2:13:52 PM
User: NT AUTHORITY\SYSTEM
Computer: BORIS
Description:
In filter "<filter name>", object, <object id>, is not defined in the DCS.
The objects referenced in a filter must be defined in the DCS. This filter will be skipped in this cycle.

Event Type: Warning
Event Source: DMPerfss
Event Category: Filter
Event ID: 611
Date: 5/25/2003
Time: 5:49:41 PM
User: BORIS\dsteier
Computer: BORIS
Description:
In filter "<filter name>", counter <counter id>, in object <object id>, is not defined in the DCS.
The counters referenced, in a filter, must be defined in the DCS. This filter will be skipped in this cycle.

- Fix: IPC does not return Command and Class when NAKing version number.

  This was causing DmCmd to display GetMachineEnvironment, instead of the sent command, for any NAKed command. This has been fixed.

- Enhancement: When a Reporting Account is used, the account profile is loaded, unconditionally. This presented a problem for customers that don't enable profiles due to security concerns on the server.

  Profile Loading is now optional and parameterized. Formerly, it was required.

  Loading of the account profile is controlled by the "-profile" installation parameter. The "-profile" parameter is optional. If not specified, it defaults to "Required" which means a successful profile load must occur in order to use the account. If used, it must be specified with the "-account" parameter.

  Note that Profile Loading is a separate step from Account logon. It is possible to use a logged on account without a profile, however, there may be side effects (discussed later).

  Example

  C:\NTSMF> dmperfss -install -account pstest -password pstest -profile required

  Performance SeNTry version 2.4.6: successfully installed!

  C:\NTSMF>

  The option is described as follows:

  -profile <*LoadingOption*>

<*LoadingOption*>	Description
Required	If the profile fails to load, the account will not be used. An event message is issued for failures.
Optional	If the profile fails to load, the account will still be used. An event message is issued for failures.
None	The profile is never loaded. No event messages are issued.

Note that Profile Loading is available using IPC (version 6), also.

With optional profile loading, if the profile fails to load, the account will be used but the profile will not. It will default to the Local System profile (Default User). This will result in a reporting account without standard initialization (environment variables, mapped network drives, current user registry access, etc.) and it may not have sufficient permission to access some resources owned by Default User. However, profile loading does not affect network access, in general, although network drives will have to be manually mapped.

- Fix: Account specified with no domain uses the local machine domain when made using IPC (DmCmd, etc.).

  Accounts specified without a domain are local machine accounts (.\Account). When specified using IPC, the machine's domain is used instead. This has been corrected.

- Fix: Module Object is not written to data file.

  The Module Object was broken in version 2.4.5. It has been fixed.

- Fix: Single character filename discriminators are written in the wrong location.

  During hierarchical file movement, a filename is discriminated with a single character if the name already exists in the destination directory. The original algorithm was the character should precede the ".SMF". The bug is that it is sometimes is off by one position to the left. E.G.

  MACHINE.200005150300.SUM.SMF
  MACHINE.200005150300.SUAM.SMF

  Should be written as

  MACHINE.200005150300.SUMA.SMF

  The algorithm was changed to always write the single character after the date/time field of the file name. This allows manipulation of the filename by extension (*.SUM.SMF) using scripting, etc. E.G.

  MACHINE.200005150300.SUM.SMF

  will be written as

  MACHINE.200005150300A.SUM.SMF

- Fix: Triplet is missing from Registry Info (Type 10) records.

- Enhancement: IPC version 6 implemented.

  The IPC version was changed to reflect the profile changes.

Version 2.4.5

- Enhancement: Disk Configuration Records Are Written to the Data File.

  Configuration Record types 11 (ATA Disks) and 12 (SCSI Disks) are written to the data file when the Disk Configuration Records parameter is enabled. Note that this parameter also controls Fault Tolerant Disk Configuration (Type 9) records.

  SCSI Record Format

  ```
  0,12,[ ntsmf header...],[ntsmf triplet...],
  [WinNT disk#],
  SCSI,
  SCSI Version #,
  [vendor],                 7.4.2 standard inquiry data
  [product],                7.4.2 standard inquiry data
  [revision],               7.4.2 standard inquiry data
  [serialnumber],           8.6.6 Unit serial number page
  [# identifiers N],        8.6.4 Device Identification Page
  [id #1 codeset],[id #1 assoc],[id #1 type],[id #1 ident],
  [id #2 codeset],[id #2 assoc],[id #2 type],[id #2 ident],
    .
    .
    .
  [id #N codeset],[id #N assoc],[id #N type],[id #N ident],
  ```

  ATA Record Format

  ```
  0,11,[ ntsmf header...],[ntsmf triplet...],
  [WinNT disk#],
  IDE,
  ATA version Number,
  [modelnumber],    8.15.21 Words (46:27): Model number
  [serialnumber],   8.15.17 Words (19:10): Serial number
  [revision],       8.15.20 Word (26:23): Firmware revision
  ```

- Fix: The Following System Object Counters are missing on Windows NT:

  - % Total Processor Time
  - % Total User Time
  - % Total Privileged Time

  Although not reported, the following counters may also be missing:

  - "Total Interrupts/sec"
  - % Total DPC Time
  - % Total Interrupt Time

The collection agent was not detecting the correct version of Windows and was using Windows 2000 Synthetic System Counters that do not operate on Windows NT. This defect has been corrected.

- Change: Network Interface Instances with Identical Names are Discriminated

  A '#n', where n is a number starting with 1, is appended to all identically named Network Interface instances.

- Fix: Collector manages data & log files from other machines when a common network share is used for the data directory.

  The collector was changed to only look for file names that start with the local machine name when doing data file management.

- Fix: Win32 Exception in PSSDataSet::insert().

  This bug was discovered in v2.4.3 so it also exists in 2.4.4.

  This bug has the following properties:

  - One or more objects must fail to return data across one or more cycles.
  - Duplicate objects may occur in event id 2200, increasing by one every cycle.
  - Eventually, a fatal Win32 Exception is generated where the stack top contains memcmp() followed by PSSDataSet::insert().

  Single Object Cases

  ```
  01/04/03-00:00:01 - Event ID: 2200, Category: Collection, Severity:
  Warning
  The following objects were specified in the DCS but failed to return
  data.
  These objects will be skipped this cycle.
  LogicalDisk (L236, G236)
  ```

  - OR -

  ```
  05/05/03-00:00:00 - Event ID: 1309, Category: PerfData, Severity:
  Warning
  Print Spooler (Global): no objects were found in the performance
  data buffer
  ```

  Duplicate Object Case

  01/07/03-09:36:01 - Event ID: 2200, Category: Collection, Severity: Info
  The following objects failed to return data.
  The cycle will be delayed for up to 3 minutes in an effort to obtain a complete data set.
  HTTP Service (L0, G4054)
  HTTP Service (L0, G4054)
  HTTP Service (L0, G4054)

  Stack Trace Example

  01/07/03-09:36:20 - Event ID: 97, Category: N/A, Severity: Error

Win32 exception, 0xC0000005, encountered at location <address>
Read attempt at location <address>
Module List:
[Module List Removed]
Stack Trace (ThreadId <id>):
MSVCRT.dll: memcmp + <n> bytes
DMPerfss.exe: PSSDataSet::insert + <n> bytes
E:\DmPerfss 2.4.3.3\Collector\PSSDataSet.cpp, Line 1065 + <n>bytes
DMPerfss.exe: PSSFileDCSArchive::ReadDataSet + 507 bytes
E:\DmPerfss 2.4.3.3\Collector\PSSFileDCSArchive.cpp, Line 1017 + <n> bytes
[rest of list removed]

- Fix: Synthetic Disk Counters missing every other interval when base counters are reported.

- Change: Service Installation: Account Affects Only Reporting

  The account parameter formerly affected the entire service. It was changed to only affect the reporting of data and the management of the output directory (compression, file movement, etc.) including any processes launched.

  The service is now installed under LocalSystem, by default. The user may still change the logon account of the service, using the Service Control Manager. However, use of an account other than LocalSystem is not advised as it may affect the collection of certain performance objects that require or assume system level privileges.

- Change: The Service Ignores the Performance Library Disabled Flag.

- Change: ProcessWorkingSetThreshold filter raised to 5Meg for default.

- Change: Process and Thread objects are collected serially and before any other objects, which are collected asynchronously.

- Fix: "#J' In The Counter Field In Data Record.

  This behavior is caused by a collection timeout which is logged as an event message.

  The collector sequentially collects objects from a single perflib. A timeout of one of these objects, in the sequence, causes the rest of the sequence to re-use prior interval's data. This results in division by zero for rate counters because the denominators are equal. A '#J' is written in the counter data field when a division by zero occurs. Also, raw counts are duplicated, but this may be hard to distinguish from the case where there was valid, but identical, activity in the current and previous interval.

  The fix causes no data records to be emitted in the data file, for objects that fail to collect for a timeout or any other reason.

- Enhancement: IPC version 5 implemented.

  IPC was changed to honor account/password changes.

Version 2.4.4

- Enhancement: Added Synthetic Counters To the Logical and Physical Disk Objects.

The following counters appear in the data records of Logical and Physical Disk Objects. They are not selectable by the user.

- % Disk Busy
- Avg Disk Service Time
- Avg Disk Queue Time

- Bug Fix: Missing Reason For Process/Thread Priority/Working Set Change Failure.

  The collector was missing the crucial information from event messages for behavior stated above.

- Enhancement: Added Module Object.

  Support for a new synthetic object, called a Module Object, was added. This object will report Module names, using full pathnames, as instance names. The counter IdProcess is used to correlate module instances to the parent process.

- Bug Fix: SMF Files Are Not Compressed If The Collector Abnormally Terminates.

  The collector was changed to compress all SMF files. Formerly, it would only compress the most recent which caused a problem if the collector was abnormally terminated because a non-compressed file would be missing at the end of the next cycle.

Version 2.4.3

- Bug Fix: New SMF File Corrupted When Cycle Is Re-Started After a Write Error On The SMF File.

  At the beginning, the new file will contain one or more records from the previous file.

- Bug Fix: Message Reporting Level Not Honored Unless Info & Warning Are Both Selected.

- Bug Fix: Lotus Notes Objects Names Have <no name> When Lotus Notes Is Not Installed.

- Bug Fix: Lotus Notes Objects Missing From "Not Installed On Machine" Log Message.

- Bug Fix: The Collector Suspends when No DCS Filename Separator Is Used.

  The following message is logged:

  01/23/02-13:46:34 - Event ID: 301, Category: File System, Severity: Error
  Could not open the file, "<*path*>.SMF".
  The system error was "The process cannot access the file because it is being used by another process (32)"

  Additionally, the log file is created with, and the Cycle End/Error Command macros resolve to, a filename of only the machine name and no extension.

- Bug Fix: The collector sometimes stops with a Context Dump when there is a collection or performance library communication failure.

  The collector stops with an error similar to the following context dump:

01/15/02-12:02:10 - Event ID: 97, Category: N/A, Severity: Error
Win32 exception, 0xC0000005, encountered at location 0x<*number*>
Read attempt at location 0x<*number*>
Module List:
<*module list*>
Stack Trace (ThreadId <*number*>):
<*stack trace*>

- Bug Fix: Logical and physical disk are not collected when Disk Limit in Megabytes (not percent) is present in the DCS.

  This problem is only seen on Windows 2000 versions and above.

Version 2.4.2.10+ (beta)

- Bug Fix: Certain Performance Libraries failed to load.

  Performance Libraries were failing to load when their components (other dlls) were not found because they were not installed in the standard places the OS expects. The collection agent was changed to cause the Operating System to also look in the Performance Library installation directory.

- Bug Fix: Compression of large SMF files cause collection agent to hang.

- Bug Fix: Collection Agent Hangs when a component of a Performance Library is not found.

  When a performance Library was loaded, an Operating System alert was displayed that required a response from the user if a component of the performance library was not found. This alert has been disabled.

- Enhancement: disabled Performance Libraries are ignored.

- Enhancement: Selected DCS Objects Are Required To Supply Data at Cycle Start.

  At cycle start, certain DCS objects are now required to be ready supply data. Performance SeNTry will not start the cycle if these Performance Libraries are not ready to supply data. The objects are Cache, System, Process, Memory, Thread, and Processor. The collection agent will wait for these objects to become ready to supply data according to the CycleDataReadyDelay parameter.

- Enhancement: Added support for the 'CycleDataReadyDelay' parameter.

  The 'CycleDataReadyDelay parameter indicates the number of seconds to wait, at cycle start, for perflib dlls to initialize before the cycle starts. This is helpful on large systems where the collection service is ready to collect data before the performance libraries are ready to provide data. The default is 180 seconds. Setting the value to zero instructs the service not to wait for slow perflibs to return data.

Version 2.4.2.9 (beta)

- Bug Fix: not supplying an argument to -f, -account, or -password causes dmperfss to crash.

- Bug Fix: ReportRegistryInfo displays non displayable character between registry values when the DCS is a File DCS.

- Bug Fix: ReportRegistryInfo only writes last registry value to the SMF file

- Bug Fix: Activity filters are not removed when Object collection fails (its not found or no has data).

Version 2.4.2.8 (beta)

- Bug Fix: Disklimit retry cause crash

- Enhancement: Sample Base counter algorithm changed.

Version 2.4.2.7 (beta)

- Bug fix: Occurrences of extra commas in the data records.

  Extra commas in the data records were occurring on non-US English versions of Microsoft Windows. These numbers were written using locale sensitive functions, which use commas instead of decimal points for certain locales. The counter formatting functions were changed to always write with decimal points regardless of the locale.

- Enhancement: ' Base' added to base counter names.

- Enhancement: Debug Symbols are part of the distribution.

  Debug symbols were added to the distribution as an optional component. If located in the same directory as the collection agent, they will be install during "dmperfss -install". They do not affect the normal operation of the collection agent. However, they aid in problem tracking should a fatal operation system problem occur.

- Debug: The default debugger (usually Dr. Watson) is invoked when a fatal system error is encountered.

  Normally, the collection agent logs and ignores fatal errors in performance libraries and stops gracefully on fatal errors, otherwise. In this beta, the debugger is invoked to produce a Dr. Watson log to help with problem determination.

- Enhancement: Added Pseudo Process Counter: Threads Ready or Waiting.

  A new NTSMF counter that indicates the number of Threads in ready or wait states has added to the Process record. There is no longer any need to write Thread object records.

- Enhancement: SQL Server 2000.

  The Microsoft naming convention that included the dependency on the server process name (MSSQL$<server process>:<object name>) was changed to allow the selection of an SQL Server 2000 performance object in an installation independent way. The installation dependant information (object name /server process relationship) was translated in the SMF parent/instance formatting.

- Enhancement: Separate defaults for Windows 2000.

The collection agent internal default DCS was separated into a Windows 2000 version and a Windows NT version. In addition to the NT objects, the Windows 2000 default DCS includes Thread, Network Interface, IP, and TCP.

Also, on Windows 2000, the interval period was decreased to 60 seconds. It remains 5 minutes on NT.

- Bug Fix: To maintain full compatibility with NT 4.0, the System Object % DPC Time and % Interrupt Time counters were added for backwards compatibility.

- Enhancement: Added filters to default DCS.

- Bug Fix: Configuration record reports the wrong memory size for machines with more than 4GB of memory.

  Memory size field in header fixed for Win 2000 systems with 4GB or more of physical memory. No fix is possible for Windows NT4 systems with 4GB (or more) of physical memory.

- Enhancement: Installation for running under a specific account.

  The new installation parameters "-account" and "-password" were added to install the collection agent so it may be run using a specified account. This method is an alternative to using the Services Control Panel to change the account.

  Each parameter takes and argument specifying account and password. The account may be prefixed with a domain and a reverse slash, if a domain account is used, otherwise a local machine account is assumed. The password is optional, if one isn't required. LocalSystem is assumed if the account parameter is not present.

  Please note that using an account, other than LocalSystem, requires special permissions for Performance SeNTry to run correctly. Please refer to the documentation for details.

- Bug Fix: @LastDataFile points to file that doesn't exist.

  The Cycle End Command (CEC) macro, @LastDataFile, used the pathname of the last data file BEFORE it was moved by Hierarchical File Management (HFM). The CEC is run AFTER HFM, causing @LastDataFile to reflect an incorrect path during the CEC. Also, during HFM, the file name may be slightly modified if an identical name exists in the destination directory (see version 2.4.0 Bug Fix: Duplicate Data file names interferes with Hierarchical Data File Management (DFM)) causing the @LastDataFile to point to the name of a file that does not actually exist. @LastDataFile was changed to point to the last data file AFTER it was moved, reflecting both the file name and location changes of HFM.
  Notes:
  1. When HFM is not used is used, @LastDataFile will always contain the pathname of the data file in the output directory.
  2. If the last data file is deleted, not moved, @LastDataFile will reflect the pathname of the deleted file.
  3. If @LastDataFile used in an On Error Command (OEC) when Performance SeNTry fails to start a new cycle, @LastDataFile will reflect the data file from the previous cycle, or blank if Performance SeNTry is starting.

- Bug Fix: Cycle End Command (CEC) fails when arguments are quoted.

The Cycle End Command (CEC) fails, when the CEC is non-executable, e.g. .cmd, .bat, .vbs, etc. or cmd.exe.
Performance SeNTry has been fixed to correctly run non-executable commands without having to specify the associated executable. Formerly, non executables were run using "cmd.exe /c" or another executable, such as wscript.exe, as the CEC.

A Note About cmd.exe In Cycle End Commands

Sometimes cmd.exe is used to execute a Cycle End Command (CEC) with a CEC macro that contains a path, such as:

```
cmd.exe /c @StartupDir\CEC.cmd @LastDataDir
```

This runs correctly when @StartupDir and @LastDataDir do not contain spaces in the pathname, as is in the default installation of NTSMF. However, when NTSMF is installed in a location where the path contains spaces, such as "c:\program files", we previously recommended that all arguments, as well as any user supplied arguments containing spaces, be quoted:

```
cmd.exe /c "@StartupDir\CEC.cmd" "@LastDataDir" "arg with spaces"
```

However, this is illegal syntax for cmd.exe. cmd.exe expects multiple non-quoted arguments or a single quoted argument. This single argument, may contain quotes. The correct syntax is to surround all quoted arguments with quotes:

```
cmd.exe /c ""@StartupDir\CEC.cmd" "@LastDataDir" "arg with spaces""
```

Additionally, if the command contains a pathname with spaces, it should be quoted, also:

```
"c:\program files\SuperScripto.exe" "@StartupDir\CEC.ss" "@LastDataDir" "arg with spaces"
"c:\Windows 2000\cmd.exe" /c ""@StartupDir\CEC.cmd" "@LastDataDir" "arg with spaces""
"@StartupDir\CEC.cmd" "@LastDataDir" "arg with spaces"
```

- Bug Fix: SMF compression is attempted when Performance SeNTry fails to initially start.

  Performance SeNTry incorrectly attempts to compress a non-existent SMF file because a cycle was never started due to a startup failure. This only happens if compression is enabled in the DCS. The following error message may appear in the event log:

  ```
  03/22/01-01:07:45 - Event ID: 1402, Category: Cycle End, Severity: Warning

  The following compression command returned an exit code of 4. This might mean that it did not complete successfully or it may have partially completed. See the following for details.

  "E:\NTSMF241\pkzip25.exe" -.dd .ZIP

  PKZIP: (E4) Unknown or illegal option - .dd
  ```

- Bug Fix: Incorrect time change notification in event log.

Performance SeNTry incorrectly issues a "time change" notification when a resume or stop is issued when the Collection Agent has been suspended for more than 1 hour. The following is an example.

> 03/23/01-08:56:58 - Event ID: 1203, Category: Dispatcher, Severity: Warning
> The system time has moved forward by 1 hours and 33 minutes

- Bug Fix: Discovery Record/Data Record mismatch on Windows 2000.

  On Windows 2000, discovery record 5, for Logical Disk, is missing one or more counter names, however, data is appearing in the corresponding field in the data record. The same error may occur in other Discovery Record types. This was due to a misinterpretation of the counter type of the extra large Logical Disk base counters on Windows 2000. This has been corrected so that the base counters display the counter name of the numerator, which is the usual case for other objects.

- Bug Fix: Discovery Records indicate uninstanced objects while data records show up as instanced.

  If, during discovery, zero instances are returned, the discovery record will indicate an uninstanced object. During collection, should intervals appear, the data record will be written as an instanced object. The software was corrected to always indicate an instanced object, even if no instances are present.

  This was first reported with v2.4.1 and affected the MS Exchange, Logical disk and Job Detail in Windows 2000 systems.

- Bug Fix: Synthesized Windows 2000 System counters, "% Total Processor Time", "% Total User Time", and "% Total Privileged Time" show double the expected value.

  This was due to incorporating the _Total instance into the sum of the instances used to calculate the totals. The summing was replaced by using the _Total instance values directly.

Version 2.4.1

- Enhancement: Improved immunity to misbehaving performance libraries.

  The collector uses operating system threads to collect performance data making it much more immune to misbehaving performance libraries.

Version 2.4.0

- Bug Fix: fixed instance names that are incorrectly, ANSI encoded, in the performance data buffer. This was causing the UNICODE to ANSI conversion to convert a string that was already in ANSI resulting in strange characters written into the data file.

- Enhancement: changed the following in the internal, default DCS to more closely match SeNTry Admin's defaults:

  - Enable writing type 6 discovery records.
  - Enable data file sharing.
  - Start Markers secs = 0.

- Disk Limit = 100 MB.

Note that the internal, default DCS differs from SeNTry Admin's "Default DCS" in two ways:
- There are no defined filters.
- The Collection Interval is 5 minutes instead of one minute to account for the extra data generated by having no filters.

- Enhancement: dmperfss -suspend. The "-suspend" parameter (used in conjunction with the "-install" parameter) puts the service is in Manual Cycle Start Mode. Please see the description of the Manual Cycle Start Mode enhancement for details.

- Enhancement: Manual Cycle Start Mode. The service is put in Manual Cycle Start Mode when it is installed with the new "-suspend" parameter or manually suspended. In this mode, it starts in the suspended state, instead of automatically collecting, and must be manually resumed using DMcmd, SeNTry Admin, or a net continue command. This mode is remembered when the service stops in this state and it will start in Manual Cycle Start Mode when re-started.

- Bug Fix: Duplicate Data file names interferes with Hierarchical Data File Management (DFM).

  The problem is observed when the service is started or cycled twice in the same minute and a duplicate data (SMF) file is created. When Hierarchical DFM is used with a non-zero Current Retention, the duplicate is over-written, which is not a problem. When the Current Retention is zero, the duplicate is moved from the Current directory at Cycle End. At the end of the next cycle, when DFM is performed again, the file in Current will fail to move because of the duplicate.

  The fix is when a duplicate is detected, the filename of the newer file is changed slightly by inserting a single letter after the date & time.

  Examples:
  MOOSE.200005082241.SMF is changed to MOOSE.200005082241A.SMF
  MOOSE.200005082241.SMF.ZIP is changed to MOOSE.200005082241A.SMF.ZIP

  Note that this change affects all files moved by DFM, not just SMF files.

- Bug *Unfix*: In version 2.3, several programming attempts were made to account for resource leakage (Process: Virtual Bytes, Handle Count) in Performance SeNTry when the Network Segment is in the DCS. The problem is in the Performance Library supplied my Microsoft and was found on Windows NT 4.0 over a range of service packs. This problem has not been verified on Windows 2000.

  The fixes were removed in version 2.4 because they were causing other problems. Please be aware that, in version 2.4, collecting Network Segment will result in resource leakage. This leakage may be minor and tolerable for short periods of time (one week, possible one month). If this is so and Network Segment data is required, the service may be restarted (NOT resumed) intermittently to reclaim the lost resources.

- Enhancement: The service will automatically restart, after a delay period, when a disk full error is encountered. On disk full error, the service will log a message, suspend itself and check every 5 minutes or interval duration, which ever is greater, for the disk full condition to disappear, at which time it will restart.

A suspend command, while the service is suspended waiting for the disk full condition to disappear, will disable the automatic restart. This event is not logged.

- Enhancement: The Performance Object Collection Buffer is re-used instead of being re-allocated, saving time and memory fragmentation.

- Enhancement: The buffer used to format messages is pre-allocated increasing the probability of logging messages during low memory conditions.

- Enhancement: Additional system error messages (such as access violation) are now logged.

- Enhancement: Event messages have the message code printed after the message text.

- Enhancement: The Initialization Message in not redundantly logged. Formerly, it was logged at the start of every cycle. Now, it is logged at service start, when the DCS is changed, and at the start of every new log file.

- Enhancement: One or more process names can be specified on the "Process Busy Threshold Filter". These processes so named are always written into the data file without regard to the filter threshold.

- Bug Fix: Fatal errors are trapped during discovery. Formerly, the service crashed if a Performance Library misbehaved. *All* objects supported by a Performance Library that issues a fatal error will be skipped and a message logged.

- Enhancement: The "On Error Command" was added. This command will execute, in similar fashion to the Cycle End Command, when the service stops or suspends with an error.

- Enhancement: Performance Counters less than 0.0001 are written with 6 decimal places.

- Enhancement: The following new Filters were added to support Windows 2000 and Terminal Server:
  - Suppress Idle User - suppresses the Idle instance of the User object
  - User Busy Threshold - filters the User object in a fashion similar to the Process object
  - Session Activity - filters Session objects with zero activity (all zero counters)

- New Feature: Collection Period. Allows the service to run only during the time period specified. The following is information on the dynamics of this new feature.

  - The service will suspend, waiting for a Collection Period, if the current time is outside of the Collection Period. Note that this type of suspension is different from suspending on error or by request. In this type of suspension, a DCS is active and will honor all parameters as they become active (such as Cycle End Command, see below). By contrast, a Cycle End Command is never scheduled when the service is suspended on error or by request.
    Note, however, that a scheduled Cycle End Command (one with a delay) could be run in a "suspended on error" mode if it was scheduled in a cycle and the next cycle failed to start, due to an error, causing the service to suspend.
  - The following actions occur, in the order specified, during Cycle Hour Processing:
    - Collection Ends, if is it running.
    - Data File Maintenance
    - Cycle End Command is scheduled

- Cycle Duration > Collection Period is ignored. Cycle Duration <= Collection Period is allowed, but new cycles start on the hour.
- Cycle Start is dictated by the start time of the Collection Period, not Cycle Hour. However, Cycle Hour may occur within or outside of a Collection Period. If it occurs within a Collection Period, regular Cycle Hour Processing is done (see above). If it occurs outside of a Collection Period, only Data File Maintenance and Cycle End command are done.

  Also note that Cycle Hour Processing is **NOT** done if the service is suspended with an error or by request. It is done only if it is suspended waiting for the Collection Period to start.
- Cycle End Processing includes Cycle Hour Processing, if the Cycle End coincides with Cycle Hour. Cycle End Processing also includes Data File Maintenance.
- Cycle Hour Processing occurs at Cycle Hour. If the Cycle Hour occurs outside of the Collection period, the only the Cycle End Command.
- When the Cycle Hour occurs at Collection Period Start, the Cycle hour will occur before the actual start of the Collection Period.
- When the Collection Period is specified, any NextDCS definition is ignored.

- New Feature: Next DCS. Allows another DCS to start at the time specified (creates a DCS chain). The service starts a DCS chain specifying an active DCS using DMcmd or SeNTry Admin). Each DCS in turn points to another in the chain. The following is information on the dynamics of this new feature.

  - Cycle Hour Processing will ONLY occur at Cycle Hour if the DCS specifying Cycle Hour is running. However, you are not precluded from specifying different Cycle Hour/Cycle End parameters in each DCS, in the chain.

    If a single Cycle Hour Processing per day is desired, then the following parameters should be identical in all the chained DCSs.
    - Cycle End Command
    - Cycle Hour
    - CycleEndCommandInhibit

  - There is no limit to the number of DCSs in the chain.
  - Since NextDCS specifies a file DCS, registry DCSs are not valid for specifying the NextDCS parameter.
  - No validation is done to ensure that a NextDCS exists. If it doesn't, the chain is broken and the currently running DCS will be re-run, instead.
  - No validation is done to ensure that a circular chain exists (either there is no NextDCS parameter or all of the DCSs in the chain are present).
  - No validation is done to ensure that the active DCS is the one supposed to run at that time. At DCS activation, the NextDCS Start time is always assumed to be in the future. Eventually, over a 24-hour period, the correct DCS sequence will re-establish itself.
  - NextDCS processing will only occur if the service is collecting. If re-started or resumed the active DCS at the time of stop or suspension will be the one used. The user is responsible for running the correct DCS when the service initially starts or resumes from suspension by activating the desired DCS using DMcmd or SeNTry Admin.