



## Problem Determination Overview

Edition 7.3

11 January 2013





## Portrait Foundation Problem Determination Overview

©2013  
Copyright Portrait Software International Limited

All rights reserved. This document may contain confidential and proprietary information belonging to Portrait Software plc and/or its subsidiaries and associated companies.

Portrait Software, the Portrait Software logo, Portrait, Portrait Software's Portrait brand and Million Handshakes are the trademarks of Portrait Software International Limited and may not be used or exploited in any way without the prior express written authorization of Portrait Software International Limited.

### Acknowledgement of trademarks

Other product names, company names, marks, logos and symbols referenced herein may be the trademarks or registered trademarks of their registered owners.

### About Portrait Software

Portrait Software is now part of [Pitney Bowes Software Inc.](#)

Portrait Software enables organizations to engage with each of their customers as individuals, resulting in improved customer profitability, increased retention, reduced risk, and outstanding customer experiences. This is achieved through a suite of innovative, insight-driven applications which empower organizations to create enduring one-to-one relationships with their customers.

Portrait Software was acquired in July 2010 by Pitney Bowes to build on the broad range of capabilities at Pitney Bowes Software for helping organizations acquire, serve and grow their customer relationships more effectively. The Portrait Customer Interaction Suite combines world leading customer analytics, powerful inbound and outbound campaign management, and best-in-class business process integration to deliver real-time customer interactions that communicate precisely the right message through the right channel, at the right time.

Our 300 + customers include industry-leading organizations in customer-intensive sectors. They include 3, AAA, Bank of Tokyo Mitsubishi, Dell, Fiserv Bank Solutions, Lloyds Banking Group, Merrill Lynch, Nationwide Building Society, RACQ, RAC WA, Telenor, Tesco Bank, T-Mobile, Tryg and US Bank.

Pitney Bowes Software Inc. is a division of Pitney Bowes Inc. (NYSE: PBI).

For more information please visit: <http://www.pitneybowes.co.uk/software/>

#### UK

Portrait Software  
The Smith Centre  
The Fairmile  
Henley-on-Thames  
Oxfordshire, RG9 6AB, UK

Email: [support@portraitsoftware.com](mailto:support@portraitsoftware.com)  
Tel: +44 (0)1491 416778  
Fax: +44 (0)1491 416601

#### America

Portrait Software  
125 Summer Street  
16<sup>th</sup> Floor  
Boston, MA 02110  
USA

Email: [support@portraitsoftware.com](mailto:support@portraitsoftware.com)  
Tel: +1 617 457 5200  
Fax: +1 617 457 5299

#### Norway

Portrait Software  
Portrait Million Handshakes AS  
Maridalsveien. 87  
0461 Oslo  
Norway

Email: [support@portraitsoftware.com](mailto:support@portraitsoftware.com)  
Tel: +47 22 38 91 00  
Fax: +47 23 40 94 99

# About this document

## Purpose of document

This document provides a high-level guide to Portrait Foundation problem determination and an overview of the tools and techniques available for problem determination with Portrait Foundation implementations. The tools described are either part of the Portrait Foundation product, or the commercial products on which it is built or are freely available. The vast majority of implementation problem determination activity can be done with the standard Portrait Foundation tools (such as Model Diagnosis tool and Log Viewer). Some of the tools described here (such as Userdump and WinDbg) are used only in special circumstances.

It is beyond the scope of this document to provide a detailed step-by-step approach for resolving all problems or to describe in detail the features of the tools or to provide advice on general software development problems such as logic errors and memory leaks.

## Intended audience

This document is intended for anyone involved in problem determination with Portrait Foundation implementations.

## Related documents

None

## Software release

Portrait Foundation 4.3 or later.



# Contents

<b>1</b>	<b>High-level guide to problem determination</b>	<b>6</b>
1.1	Nature of the problem	6
1.2	Circumstances of the problem	7
1.3	Progressing the problem	7
<b>2</b>	<b>Problem determination tools and techniques</b>	<b>13</b>
2.1	Portrait Foundation tools	14
2.2	Microsoft tools	18
2.3	Other third-party tools	25
2.4	Other resources	27
<b>3</b>	<b>Other tools and techniques</b>	<b>28</b>
3.1	Code quality checking	28
3.2	Load testing	28

# 1 High-level guide to problem determination

With any integrated, rich-function software application, problem determination can be a complex activity that requires a blend of skills, experience and logical thinking. However, there are two main dimensions that are absolutely central:

- Nature of the problem – the way in which the problem is manifested
- Circumstances of the problem – the necessary conditions under which the problem is present and the extent to which the problem affects the functions of the product and the users of it

When you understand the nature and the circumstances of the problem, you can then choose appropriate tools and techniques to progress the problem.

## 1.1 Nature of the problem

Most problems will be manifested in one of the following ways:

Table 1 -Nature of the problem

Nature of the problem	Description	Typical symptoms
Crash	A program or service terminates unexpectedly	For an interactive program: the operating system displays an application error message box. For a service: users of the service experience error messages, timeouts
Hang	A program or service ceases responding to inputs	For an interactive program: the program ceases responding to mouse and keyboard input. In Windows Task Manager, the application may be described as 'Not Responding' For a service: users of the service experience timeouts or no response at all
Unexpected output data	The function appears to operate but the data produced does not match expectations (data is missing or incorrect)	A custom interaction, combo box, grid control, edit control or other form of output contains no data or unexpected data.
Unexpected user interface events	The function operates but generates an unexpected user interface event such as an error message	The user interface displays an unexpected message box, navigates to an unexpected URL or displays unexpected content in a custom or generated interaction.
Slow response	The function operates correctly but takes longer than expected to perform an action	The user interface displays correct results but takes longer than expected, for example substantially more than one second to display a simple page and/or substantially more than three seconds to display a simple page.
High memory usage	The function operates correctly but uses more memory than expected	If a system has to do a lot of paging (swapping virtual memory between disk and real memory), this can be manifested in slow response.
High resource usage/exhaustion	The system performs badly or begins to fail after time as a resource is depleted ( e.g. memory, handles )	The system is seen to increase its usage of a particular resource over time. This may be accompanied by a degradation in performance and/or failures
Instability	The system does not reliably perform tasks and may appear to become less stable with time.	Apparently random failures. Inconsistent behaviour.

Each of these requires different approaches so it is important that you determine which of these you are dealing with.

## 1.2 Circumstances of the problem

Identifying the nature of a problem is the easy part. Identifying the circumstances of the problem can be more difficult.

Consider the answer to some of these questions:

- Are a majority of product functions affected or only some?
- Are a majority of users affected or only some?
- Are a majority of machines affected or only some?
- Are you using a feature for the first time? If not, what has changed since it last worked?
- When is the problem manifested? (single or multiple users; high or low load conditions; start-up, shut-down or normal usage; time of day)
- Is the problem reproducible? What is the minimum set of user actions required to reproduce the problem?
- Does the same problem occur in other Portrait Foundation environments?
- What features are common to the problem case(s) that are not present in the success case(s)?
- Which tiers or components can you exclude from the problem? For example, you could exclude user interface components by running a model through the Model Diagnosis tool and you can test some VB scripts independently of the nodes and generated interactions that invoke them.

## 1.3 Progressing the problem

You will need a good idea of the nature and circumstances of the problem as described above before you can progress to a solution. Sometimes this will be an iterative process.

### 1.3.1 Fundamental environment and installation problems

It is regarded as good practice to view all errors, warnings, failures, and alerts as being significant in respect of the reliable running of a system. This is why pursuit of any particular problem should not be undertaken until a stable, error/warning/failure free environment has been established. Pursuit of resource leaks whilst models are failing and access violations are occurring is unlikely to be the best course of action.

If a system is not configured according to the recommendations set out in the Operations Guide then it should have its settings reviewed prior to any further deep investigation.

Security being locked down in more recent versions of the Windows operating system have introduced a new class of problem. Portrait support can provide a checklist of environmental considerations e.g. userids to user for services, and privileges that need to be granted, that need to be taken into account in setting up a system successfully.

If the problem affects the majority of product functions, or affects some so badly that other functions cannot be reached, or affects one Portrait Foundation environment but not another which is running the same Portrait Foundation software, then it is likely that there is a fundamental error in the environment or software installation.

Capture and analyse all Portrait Perfmon counters on all tiers. Failed models and Exceptions should not be present and if they are should be identified and addressed as either part of or a prerequisite for fixing the current problem.

Ensure MiniDump is configured properly and that there are no '\*.dmp' files. If any exist they should be analysed and the underlying problem fixed as either part of or a prerequisite for fixing the current problem.

Check the settings used when installing the product. Try uninstalling and re-installing the product. Check the install logs for errors.

Check that the database version and the version of the configuration deployed in it are consistent with the Portrait Foundation product version you are trying to use and that the versions of Portrait Foundation product software installed on the various tiers are compatible with each other.

Check that all third-party software required by your Portrait Foundation installation is installed correctly and that the software versions are supported for the version of Portrait Foundation you are using (see the Portrait Installation Guide for supported versions of third-party software).

Check that the database server has sufficient disk space for data and log files.

On all tiers, ensure Portrait Event Logging is enabled and review logs for additional information.

Enable the Portrait Viewer destination and review Portrait LogViewer output. By default this will provide details of data, events, and errors. If this does not provide the necessary information; use the following log filter on the Log Viewer [ (\* ) @\* ) @\* ) : ! ( 1 , 2 , 3 , 4 , 5 , 12 ) ]

### 1.3.2 'Crash' problems

Table 2 - Progressing the problem: Crash

Typical causes	Coding error, leading to an unhandled exception
Tools and techniques	<p>Check for presence of files called "*.dmp" in the installed Portrait directory. Exceptions that occur in most Portrait threads are caught and these dump files are written by Portrait's MiniDump utilities. The operating system will also invoke MiniDump when an exception occurs that Portrait Foundation did not catch.</p> <p>Use WinDbg to analyse the process dump file. These files are small and only contain the callstack by default. It is possible to configure ( via the registry ) the MiniDump utilities to collect more information up to a full memory dump rather than just the stacktrace. MiniDump is documented in the document Debugging Tools</p>
Other points to consider	

### 1.3.3 'Hang' problems

Table 3 - Progressing the problem: Hang

Typical causes	Coding error, leading to a deadlock condition
Tools and techniques	<p>Take away all user load from all the servers and use Windows Performance Monitor to check the value of these counters on the appropriate servers: "Processor:% Processor Time" (this will be at or near-zero); "Portrait Process Engine: Thread Busy Count"/ "Active Server Pages: Requests Executing" / "ASP.NET Applications: Requests Executing" (one or more of these is likely to be non-zero if a deadlock has occurred whilst servicing a request).</p> <p>Use Userdump( userdump pid ), WinDBG ( .dump /ma c:\dump.dmp ) to produce a process dump file and use WinDbg to analyse the process dump file. Alternatively, use WinDbg to attach to the process and analyse it.</p>
Other points to consider	<p>Genuine deadlock conditions are highly unusual. The document Debugging Tools describes the basic steps required to identify a deadlock.</p> <p>Consider first whether you may actually have an 'Unexpected user interface event' type problem.</p>



### 1.3.4 Unexpected output data problems

Table 4 - Progressing the problem:  
Unexpected output data

<p>Typical causes</p>	<p>Incorrect inputs were supplied.</p> <p>A custom control has been incorrectly configured (for example, its XSLT transform is not correctly specified).</p> <p>A model contains a mapping of an inappropriate data item as the input to a model, node or data access transaction.</p> <p>A logical error in the definition of a model, generated interaction or custom interaction means that it does not take the expected path or does not process its input or output data correctly.</p> <p>A coding error in a node, data access transaction, database stored procedure or XSLT transform.</p>
<p>Tools and techniques</p>	<p>Check that any user inputs supplied were correct.</p> <p>Enable Portrait Logging and examine the logging output via Eventlog or via the LogViewer, possibly with the [ (*) : (*) : (*) : ! (1, 2, 3, 4, 5, 12) ] filter.</p> <p>Consider techniques that help you progress outwards from the database server to identify the tier at which the data becomes incorrect.</p> <p>Use SQL Profiler to identify the calls made against the database. Execute the calls separately in Query Analyzer to see what data they return.</p> <p>Use the Portrait Model Diagnosis tool to execute the model(s) and custom interaction(s) in question.</p> <p>Use the Portrait Configuration Suite to examine the definition of the custom interaction(s), custom control(s) or model(s) in question.</p> <p>Check the code of any project implementation node, data access transaction, database stored procedure, Web page or XSLT transform involved in the problem.</p>
<p>Other points to consider</p>	<p>Are the expectations valid? Discuss expectations with others and consider whether the results are reasonable.</p>

### 1.3.5 Unexpected user interface event problems

Table 5 - Progressing the problem:  
Unexpected user interface events

<p>Typical causes</p>	<p>Incorrect inputs were supplied.</p> <p>An implementation reference has been incorrectly configured (for example, its URL is not correctly specified).</p> <p>A model contains a mapping of an inappropriate data item as the input to a model, node or data access transaction.</p> <p>A logical error in the definition of a model or a custom interaction means that it does not take the expected path or does not process its input or output data correctly.</p> <p>A coding error in a node, data access transaction or database stored procedure.</p>
<p>Tools and techniques</p>	<p>Check that any user inputs supplied were correct.</p> <p>Enable Portrait Logging and examine the logging output via Eventlog or via the LogViewer, possibly with the [ (*) : (*) : (*) : ! (1, 2, 3, 4, 5, 12) ] filter.</p> <p>Use the Portrait Configuration Suite to examine the definition of the model(s) and custom interaction(s) in question.</p> <p>Use SQL Profiler to identify the calls made against the database. Execute the calls separately in Query Analyzer to see what data they would return.</p> <p>Use the Portrait Model Diagnosis tool to execute the model(s) and custom interaction(s) in question.</p> <p>Check the code of any project implementation script, node, data access transaction, database stored procedure or Web page involved in the problem.</p>
<p>Other points to consider</p>	<p>Are the expectations valid? Discuss expectations with others and consider whether the results are reasonable.</p>

### 1.3.6 Slow response problems

Table 6 - Progressing the problem:  
Slow response

<p>Typical causes</p>	<p>The resources required on the various tiers (Client, Web, CRM and Database), such as Memory, CPU and Disk I/O are insufficient for the demands being placed upon them.</p> <p>A coding error in an XSLT transform, node, data access transaction or database stored procedure means that excessive volumes of data are being processed or data is being processed inefficiently.</p> <p>A model contains a mapping of an inappropriate data item as the input to a model, node or data access transaction.</p> <p>A logical error in the definition of a model or a custom interaction means that it does not take the expected path or does not process its input or output data correctly.</p>
<p>Tools and techniques</p>	<p>Enable Portrait Logging and examine the logging output.</p> <p>Use Performance Monitor on each of the tiers to examine whether resource usage is higher than expected.</p> <p>Portrait Model and Node Counters may help analyse where bottlenecks are occurring.</p> <p>Portrait NWTest ( available on the offsite CD image ) application may highlight bandwidth issues between devices. This utility provides timing information for a sample DCOM network traffic pattern. Incorrectly configure network cards can cause delays that this utility will highlight.</p> <p>Consider techniques that help you progress outwards from the database server to identify the tier at which the response becomes slower than expected.</p> <p>Use SQL Profiler to identify the calls made against the database. Examine the data for CPU, Duration, Reads and Writes. Pay particular attention to any calls where the CPU exceeds 100ms and/or Duration exceeds 1000ms. Execute the calls separately in Query Analyzer to see what data they would return and what execution plan SQL Server will use.</p> <p>Check that the system environment matches the recommendations of the Portrait Installation Guide.</p> <p>Check that the settings in the Portrait Management Console match the recommendations for the environment.</p> <p>Enable Portrait Logging and examine the logging output via Eventlog or via the LogViewer, possibly with the [ (*) : (*) : (*) : ! ( 1 , 2 , 3 , 4 , 5 , 12 ) ] filter.</p> <p>Use the Portrait Configuration Suite to examine the definition of the model(s) and custom interaction(s) in question.</p> <p>Check the code of any project implementation script, node, data access transaction, database stored procedure or Web page involved in the problem.</p>
<p>Other points to consider</p>	<p>Are the expectations valid? Discuss expectations with others and consider whether the results are reasonable.</p>

### 1.3.7 High memory usage problems

Table 7 - Progressing the problem:  
High memory usage

<p>Typical causes</p>	<p>Many Portrait Foundation components are designed to use large amounts of memory in the form of caches in order to improve performance by cutting usage of other resources such as CPU and Disk I/O. Some caches do not become full immediately but continue to expand until they reach a maximum configured in the Portrait Management Console.</p> <p>A coding error in a node or data access transaction could explain why resources are consumed and never released (e.g. a memory leak).</p>
-----------------------	---

Tools and techniques	<p>Enable Portrait Logging and examine the logging output via eth Eventlog or via the LogViewer, possibly with the [ (*) : (*) : (*) : ! (1, 2, 3, 4, 5, 12) ] filter.</p> <p>Check that the settings in the Portrait Management Console match the recommendations for the environment. In particular, make sure that the Process Engine State Cache and Session Management Cache have not been set to 'Maximum'. Follow the recommended settings from the Portrait Operations guide. Prefetching of Model definitions or the caching of many model definitions can cause memory to be close to the 2GB virtual storage limit of Windows. The /3GB flag can be used to alleviate this constraint.</p> <p>Use Windows Task Manager to identify whether processes are consuming large amounts of memory.</p> <p>Use Performance Monitor to examine resource usage. Pay particular attention to the Private Bytes counter of the Process object. Note the changes to this counter over an extended period of use (many hours).</p> <p>Check any project implementation code involved in the problem.</p>
Other points to consider	<p>Are the expectations valid? Discuss expectations with others and consider whether the results are reasonable.</p> <p>Only if memory usage continues unchecked until all virtual memory is consumed will this lead to a failure.</p>

### 1.3.8 Resource depletion/exhaustion

Table 8 - Progressing the problem:  
Resource depletion/exhaustion

Typical causes	<p>Memory is not the only finite resource that can be consumed, or leaked, over time. Handles, or system PTEs are an example of a resource in this category. Leaks/depletion of these resources can manifest itself as failures or progressive instability or poor response.</p>
Tools and techniques	<p>The primary tool in determining the status of the system in this respect is Perfmon. All of the Portrait Foundation counters should be monitored as a matter of course all the time for a running system ( production or testing ). CPU, virtual bytes, context switches, handles, paging Windows Perfmon counters should similarly be enabled . Any counter where there is an upward trend or inexplicable peaks becomes a candidate for investigation.</p> <p>Enable Portrait Logging and examine the logging output via eth Eventlog or via the LogViewer, possibly with the [ (*) : (*) : (*) : ! (1, 2, 3, 4, 5, 12) ] filter.</p> <p>Check that the settings in the Portrait Management Console match the recommendations for the environment. In particular, make sure that the Process Engine State Cache and Session Management Cache have not been set to 'Maximum'. Follow the recommended settings from the Portrait Operations guide. Prefetching of Model definitions or the caching of many model definitions can cause memory to be close to the 2GB virtual storage limit of Windows. The /3GB flag can be used to alleviate this constraint. Note it is the Perfmon counter virtual bytes peak that is the indicator of how much of the 2GB virtual address range limit is in use.</p> <p>Check any project implementation code involved in the problem.</p>
Other points to consider	

### 1.3.9 Instability

Table 9 - Progressing the problem:  
Instability

Typical causes	<p>Typical causes are errors in code, configuration, or environment. Instability is an umbrella term describing a system that appears under certain circumstances to not behave in a consistent or reliable fashion. Instability can be caused by an intermittent error, corruption, uninitialised variables, poor error handling etc.</p>
----------------	--

Tools and techniques	Instability can be significantly reduced by addressing all exceptions, model failures, event log entries, and any other indicators of a system not performing optimally. If a system is not configured according to the guidelines detailed in the Operations Guide then instability may be a result. Review the configuration before proceeding. Portrait and Operating system Perfmon counters should be used to determine if there are any areas to investigate - failed models, exceptions, increasing resource usage.
Other points to consider	

## 2 Problem determination tools and techniques

The table lists tools and techniques available for problem determination and indicates their major uses in problem determination and the level of specialist knowledge required for their use.

Table 10 - Tools and techniques available for problem determination(continues)

Tool/Technique	Major uses in problem determination	Level of specialist knowledge required to operate	Level of specialist knowledge required to analyse results
Portrait Service Check	Check whether appropriate services are installed and running	None	Low
Windows Event Log	Check for significant System and Application events	Low	Medium
Windows Task Manager	Check basic health of a process (running, responding, CPU usage and so on)	None	Low
SysInternals DebugView	Check for error reports	Low	Medium
Portrait Logging	Obtain information about errors and normal function of Portrait Foundation components	Low	High
Portrait Log Viewer	Obtain information about errors and normal function of Portrait Foundation components	Low	Medium
Portrait Structured Exception Handler	Obtain information about structured exceptions detected by Portrait Foundation	None	High
Portrait Model Diagnosis tool	Execute models and examine the path taken and the data passed through models	None	Medium
SQL Profiler	Capture database events including dynamic SQL and stored procedure calls	Medium	Medium
Windows Performance Monitor	Check resource usage (CPU, Memory, Disk I/O and so on)	Low	Medium
Userdump	Extract process dump information from a blocked process	Medium	High
Process Dump applet	Extract process dump information when an exception occurs	Medium	High
ADPlus	Extract process dump information from a blocked process or when an exception occurs	Medium	High
WinDbg	Analyse process dump information	Medium	High
Portrait Known Issues list	Compare the nature and circumstances of the problem with the descriptions of known issues	None	None
Portrait Support	Access to specialist expertise when other problem determination tools have been exhausted	None	None
Portrait stored procedure p_amc_utl_sp_analysis	Analyse database events captured using SQL Profiler	Low	Low
SysInternals FileMon	Analyse access to files	Low	Medium
SysInternals RegMon	Analyse access to Windows registry	Low	Low

Table 10 - Tools and techniques available for problem determination(continues)

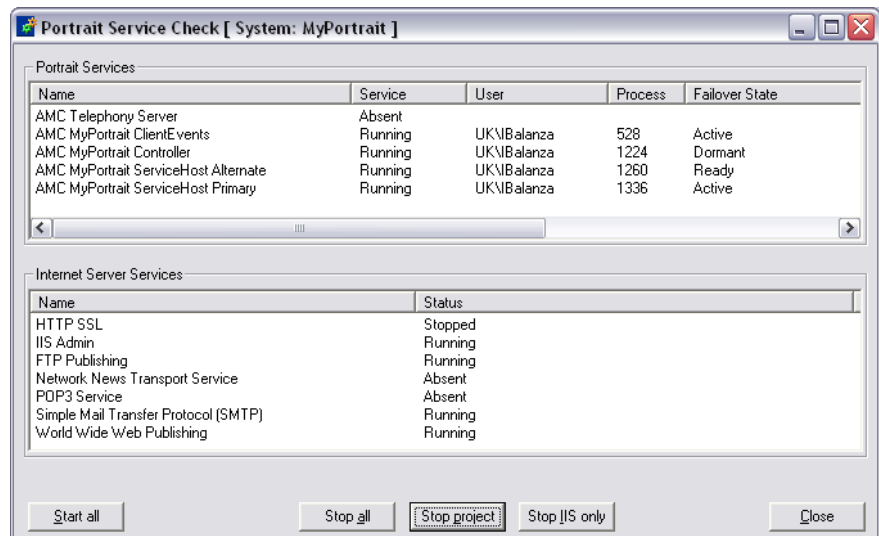
Tool/Technique	Major uses in problem determination	Level of specialist knowledge required to operate	Level of specialist knowledge required to analyse results
DumpBin	Discover detailed information about an executable component (EXE or DLL), such as the exports.	Low	Medium
OLEView	Discover information about registered COM components and interfaces	Low	Low
NWTest	Tests network speeds between machines	Low	Low
Model and Node Counters	Details runtime model and node usage	Medium	Medium
DebugDiag	Monitors for memory leaks	Medium	Medium
GloCode	Monitors for memory leaks/profiler	Medium	Medium

An overview of these tools and techniques is provided in the sections that follow.

## 2.1 Portrait Foundation tools

### 2.1.1 Service Check

Figure 1 - Portrait Service Check



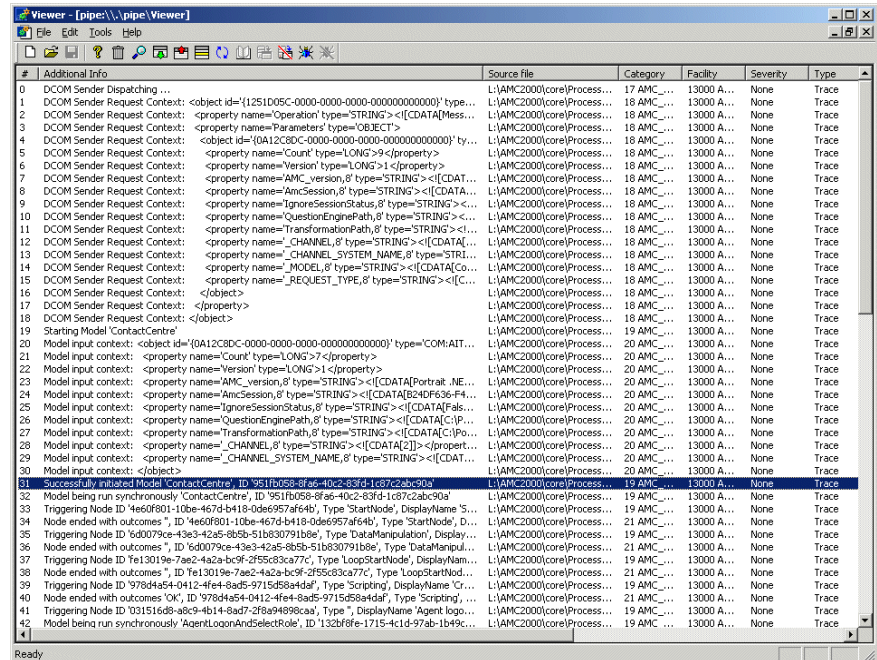
Portrait Service Check is an application which displays the status of Portrait Foundation services and, on a Web server, the services relevant to serving Web requests. It can also be used to start and stop these services. The status of each service is shown as 'Absent', 'Running', 'Stopped' or 'Stopping'.

If the circumstances of the problem suggest that a majority of system functions are affected, you can use Portrait Service Check to determine the status of the services. If a service is marked as 'Absent', when it should be present, there is a problem with the installation. If a service is marked as 'Stopped' when it should be running, it may have been stopped manually or there may be a problem that has caused it to stop.

**Use this when:** a problem is affecting a majority of system functions and you want to know what Portrait Foundation services are installed and running or you want to stop or start Portrait Foundation services.

## 2.1.2 Portrait Logging

Figure 2 - Portrait Logging, viewed with Portrait Log Viewer



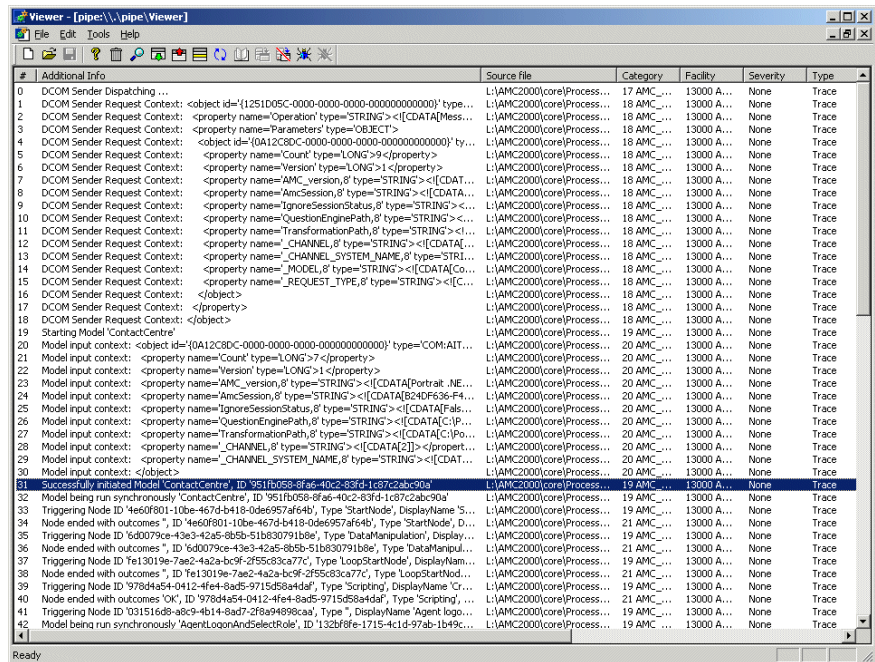
Portrait Foundation components are implemented with calls to the Portrait Logging mechanism, designed to produce output when an error or significant event occurs. When the Portrait Logging system is disabled, these calls have no effect. However, when the Portrait Logging system is enabled, the data from these calls are distributed to a number of logging destinations, which are responsible for producing formatted output. Portrait Foundation is equipped with several logging destinations, including File (output is sent to a text file, which can be viewed with a text editor), Database (output is sent to a table in the database, which can be viewed with a tool such as Query Analyzer), EventLog (output is sent to the Windows Event Log) and Viewer (output is sent to the Portrait Log Viewer).

Portrait Logging components are installed by the Portrait Foundation software install process. Portrait Logging is enabled, disabled and configured through the Portrait Management Console. There is an online application help file which explains how to use the Portrait Management Console.

**Use this when:** you want to obtain detailed information on the error and significant events occurring in a general problem scenario.

## 2.1.3 Log Viewer

Figure 3 - Portrait Log Viewer



From Portrait Foundation Release 2.4 onwards, the product includes a Portrait Log Viewer. This tool allows the user to view Portrait Logging messages in a formatted list control that supports filtering, highlighting, searching and saving.

**Use this when:** you want to check for errors and debug output in Portrait Foundation versions from Release 2.4 onwards.

## 2.1.4 MiniDump

Some major Portrait Foundation components intercept exceptions in most threads. These components then create a small memory dump in a file called **process\_timestamp.dmp** in the installed Portrait directory. These dumps can be analysed with WinDBG or Visual Studio.

This function is permanently enabled and becomes active only when an exception occurs.

MiniDump is also configured to be invoked whenever an exception is raised in a Portrait Foundation executable that is not caught within the application. ( this is in the style of Dr Watson ).

If further information other than the call stack is required MiniDump can be configured to capture more information all the way up to a full memory dump.

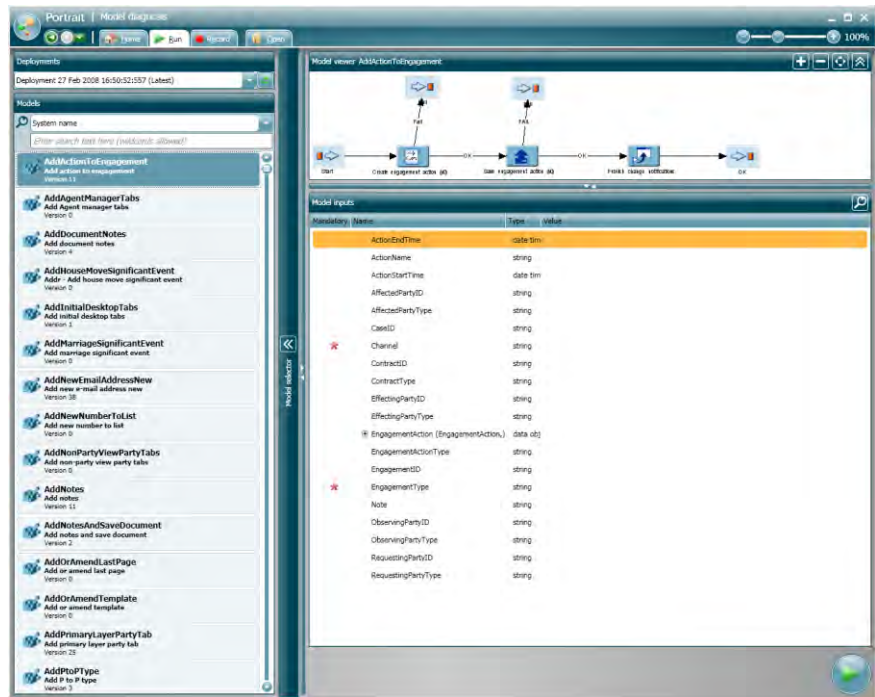
MiniDump is documented in the document Debugging Tools.

**Use this when:** you want to obtain detailed information on the location in the Portrait Foundation code at which an exception was thrown.



## 2.1.5 Model Diagnosis

Figure 4 - Portrait Model Diagnosis tool



The Portrait Model Diagnosis tool lets you run models and analyse their results. You can run models locally (using a CRM server on the same workstation as Model Diagnosis tool is installed) or remotely (on another CRM server). You can also analyse models that have run completely independently of the Model Diagnosis tool provided that the model's results have been logged to the database. When you run a model, the Model Diagnosis tool provides detailed information on all nodes executed in the model and the data they use. During its analysis the Model Diagnosis tool reads data that the model has written to a particular log destination during its execution, including information on all nodes called and the data they use.

Portrait Model Diagnosis tool is installed by the Portrait Foundation software install process. You can execute it by selecting it from the Portrait Test Tools item on the Start menu. There is an online application help file which explains how to use the Portrait Model Diagnosis tool.

**Use this when:** you want to obtain detailed information on the execution of a particular model or when you want to exclude the Web user interface.

## 2.1.6 Portrait stored procedure p\_amsctl\_sp\_analysis

Figure 5 - Portrait stored procedure p\_amsctl\_sp\_analysis

StoredProcOrSQL	NumCalls	TotalTime	MaxTime	AvgTime	TotalCPU	MaxCPU	AvgCPU	TotalReads	MinReads	MaxReads
1 p_amsctl_sp_analysis	78	944	80	12	0	0	0	1421	15	110
2 p_amsctl_sp_get_object_descrsn	3	766	670	255	0	0	0	156	20	78
3 p_amsctl_sp_get_model_config	3	529	296	176	0	0	0	1053	100	700
4 p_amsctl_sp_get_party_known_as	2	486	470	243	0	0	0	876	432	444
5 UPDATE amsctl_sp_party_type_data WITH...	1	440	440	440	0	0	0	812	812	812
6 p_amsctl_sp_move_party_search	2	423	423	211	0	0	0	379	117	262
7 p_amsctl_sp_get_input_output	10	263	263	26	0	0	0	493	7	595
8 p_amsctl_sp_get_operation	1	203	203	203	0	0	0	140	140	140
9 p_amsctl_sp_get_posid	1	203	203	203	0	0	0	312	312	312
10 p_amsctl_sp_get_node	3	169	76	56	0	0	0	96	32	32
11 p_amsctl_sp_get_party_encl_detail	1	156	156	156	0	0	0	1493	1493	1493
12 IF @@TRANCOUNT > 0 COMMIT TRAN ...	3	156	110	52	0	0	0	42	0	21
13 p_amsctl_sp_get_script_defn	7	146	63	20	0	0	0	175	6	135
14 p_amsctl_sp_get_list	8	125	93	15	0	0	0	302	9	221
15 p_amsctl_sp_get_task_history	1	123	123	123	0	0	0	242	242	242
16 SELECT TOP 771 cba_stab.contract_id...	3	120	60	40	0	0	0	1090	0	545
17 p_amsctl_sp_get_build	1	93	93	93	0	0	0	430	430	430
18 p_amsctl_sp_get_top_lvl_modi_loc	6	76	46	12	0	0	0	84	14	14
19 p_amsctl_sp_get_output_map	1	63	63	63	0	0	0	111	111	111
20 p_amsctl_sp_add_party_to_action	3	63	63	21	0	0	0	96	32	32
21 p_amsctl_sp_get_input_map	1	46	46	46	0	0	0	119	119	119

From Portrait Foundation Release 2.1 onwards, the Portrait Foundation database contains a stored procedure p\_amsctl\_sp\_analysis which can be used to analyse a trace table captured by SQL Profiler. The stored procedure is designed to analyse trace tables containing only these database events: **RPC:Completed** and **SQL:Batch Completed**. The stored procedure produces a result set that contains a line for each distinguishable stored procedure or dynamic SQL statement, along with the number of times it was executed and the minimum, maximum, total and average of CPU, Duration, Reads and Writes.

**Use this when:** you want to identify which stored procedures or dynamic SQL statements are contributing most to the CPU, Duration, Reads and Writes in a problem scenario. To use the stored procedure, you must first have used SQL Profiler to capture and save a trace table containing **RPC:Completed** and **SQL:Batch Completed** database events.

## 2.2 Microsoft tools

### 2.2.1 Windows Event Log

Figure 6 - Windows Event Log

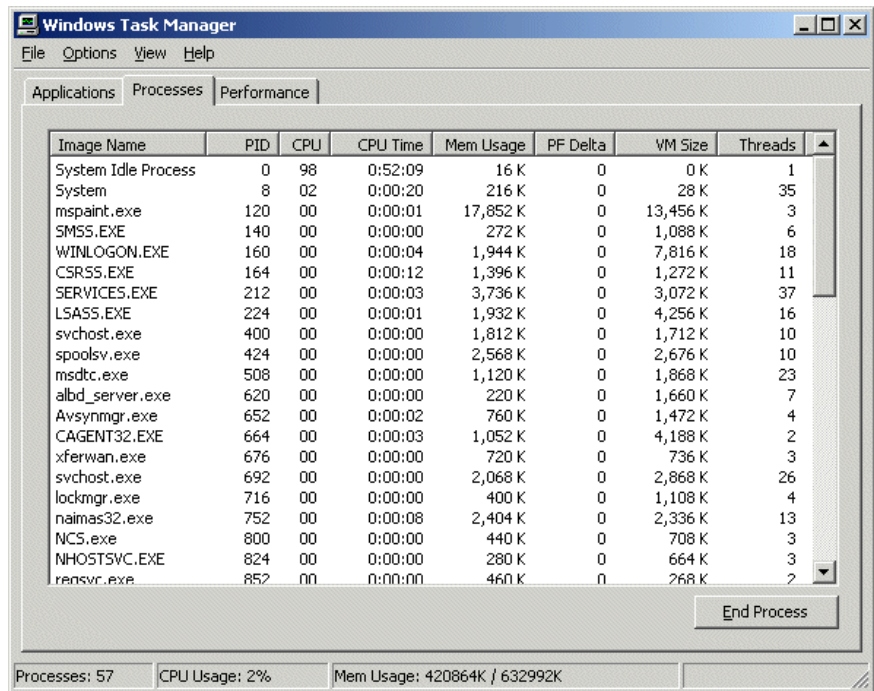
Type	Date	Time	Source	Category	Event
Information	21/05/2003	14:11:04	ServiceHost	None	0
Information	21/05/2003	14:11:03	ServiceHost	None	0
Information	21/05/2003	14:10:47	AmcCollector	None	0
Information	21/05/2003	14:10:46	AmcCollector	None	0
Information	21/05/2003	14:03:17	AmcWebChannelService	(1)	1007
Information	21/05/2003	14:03:17	AmcWebChannelService	(1)	4
Information	21/05/2003	14:03:15	AmcWebChannelService	(1)	5
Information	21/05/2003	14:01:31	AmcCollector	None	0
Information	21/05/2003	14:01:30	AmcCollector	None	0
Information	21/05/2003	13:46:09	ServiceHost	None	0
Information	21/05/2003	13:46:08	AmcWebChannelService	(1)	1007
Information	21/05/2003	13:46:08	AmcWebChannelService	(1)	4
Information	21/05/2003	13:44:51	Portrait Install	None	0
Information	21/05/2003	13:41:13	AltUserValidate	None	0
Information	21/05/2003	13:40:50	AmcCollector	None	0
Information	21/05/2003	13:39:34	AltUserValidate	(1)	4
Information	21/05/2003	13:39:04	MsiInstaller	None	11707
Information	21/05/2003	13:37:17	McLogEvent	None	5000
Information	21/05/2003	13:37:14	ClearCase	None	1024

The Windows Event Log is part of the Windows operating system. Some applications (including Portrait Foundation) write to the Application section of the event log when certain application errors or significant events occur. Windows operating system writes to the System section of the event log when a system error or significant event occurs.

**Use this when:** you want to know what Application and System events have been logged.

## 2.2.2 Windows Task Manager

Figure 7 -Windows Task Manager

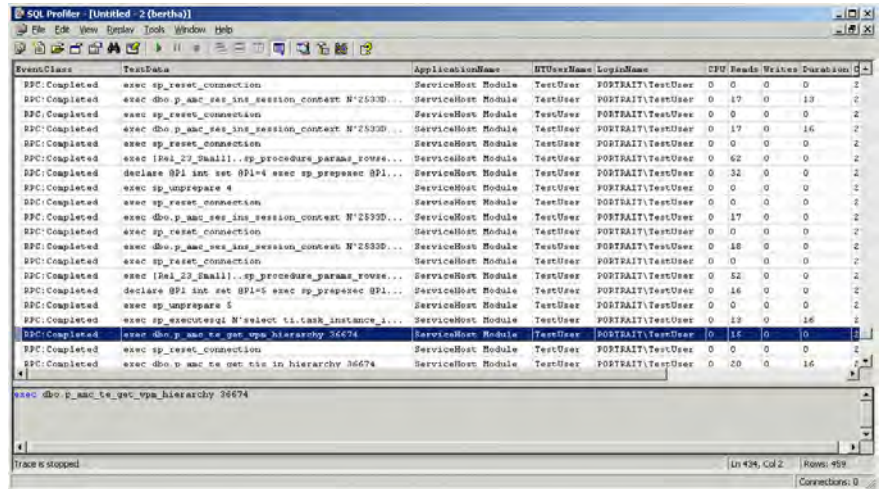


The Windows Task Manager is part of the Windows operating system. It provides a high-level view of the status of a system and some detailed information on processes running on that system (including CPU usage, memory usage and page faults).

**Use this when:** you want to confirm whether a process is running and responding and see basic information about that process. For more detailed information, use Windows Performance Monitor.

### 2.2.3 SQL Profiler

Figure 8 -SQL Profiler



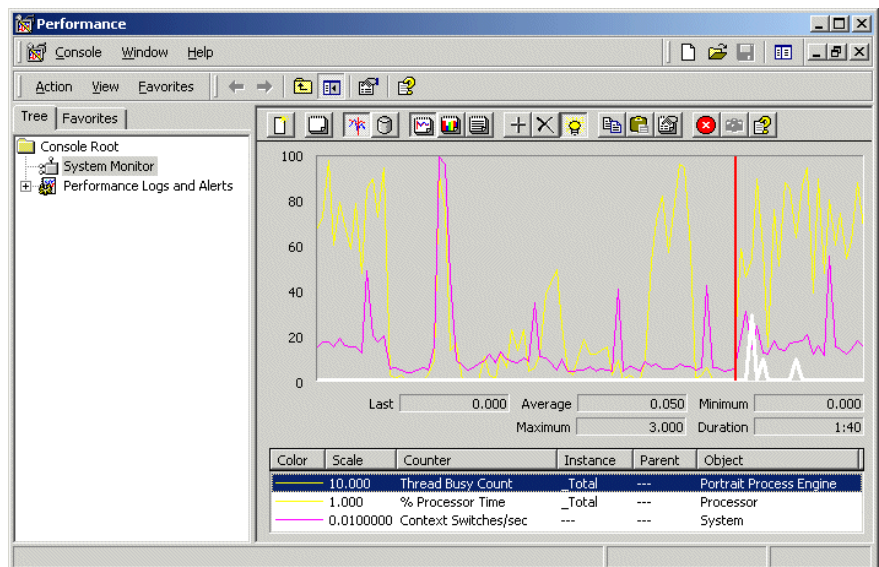
SQL Profiler is an application that allows you to capture events (including dynamic SQL and stored procedure calls) occurring on a database server.

SQL Profiler is part of the install set for SQL Server.

**Use this when:** you want to obtain detailed information on the database access occurring in a problem scenario.

### 2.2.4 Windows Performance Monitor

Figure 9 -Windows Performance Monitor



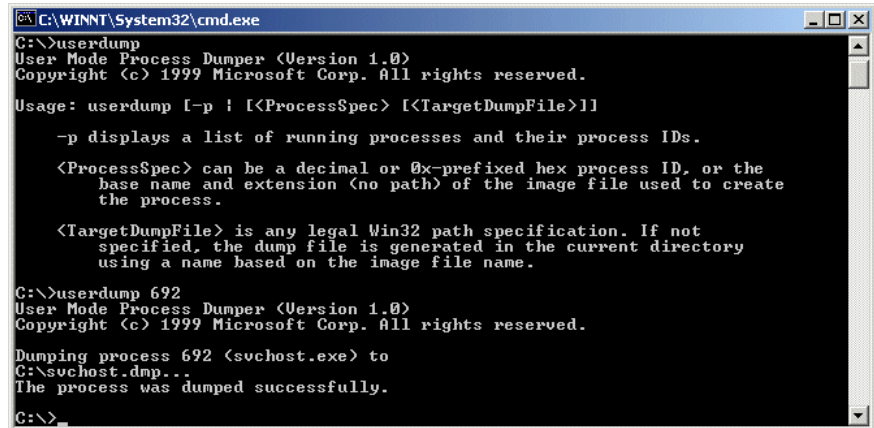
The Windows Performance Monitor is part of the Windows operating system. It provides access to very detailed information on the status of a system, in the form of objects with counters whose values can be displayed graphically and/or captured to a file for later analysis.

From Portrait Foundation Release 2.3 onwards, Portrait Foundation publishes its own counters which can be viewed in Windows Performance Monitor. These can give useful information about the performance and capacity of a Portrait Foundation system, for example the number of models started or completed per second and the number of Portrait Process Engine threads busy or waiting.

**Use this when:** you want to capture detailed information on performance-related counters (such as CPU and memory usage) over an extended period of time.

## 2.2.5 Userdump

Figure 10 -Userdump



```

C:\WINNT\System32\cmd.exe
C:\>userdump
User Mode Process Dumper (Version 1.0)
Copyright (c) 1999 Microsoft Corp. All rights reserved.

Usage: userdump [-p !] [<ProcessSpec> [<TargetDumpFile>]]

-p displays a list of running processes and their process IDs.

<ProcessSpec> can be a decimal or 0x-prefixed hex process ID, or the
base name and extension (no path) of the image file used to create
the process.

<TargetDumpFile> is any legal Win32 path specification. If not
specified, the dump file is generated in the current directory
using a name based on the image file name.

C:\>userdump 692
User Mode Process Dumper (Version 1.0)
Copyright (c) 1999 Microsoft Corp. All rights reserved.

Dumping process 692 (svchost.exe) to
C:\svchost.dmp...
The process was dumped successfully.

C:\>

```

Userdump is a command-line program that allows you to produce a process dump file from a process that is currently experiencing a 'hang' problem. This is of most use if the process appears to have become blocked by some form of deadlock problem. If the nature of the problem is not 'hang', but 'crash', you should use the Process Dump applet described in section 2.2.6.

Userdump is part of the OEM Support Tools package, available for download from <http://www.microsoft.com>.

The Microsoft Knowledge Base article *How to Use the Userdump.exe Tool to Create a Dump File* (Q242215) describes how to download and use Userdump.

**Use this when:** a process appears to have become blocked and you want to produce a process dump file in order to determine what is happening inside the process. Use WinDbg to examine the process dump file. It is often helpful to collect two memory dumps for any given process. This will allow the person doing analysis of the dumps to determine if work was being performed or not.

## 2.2.6 NWTest

This utility is on the offsite CD image. It needs installing an registering as a service on both of the machines that are to have their connection speed testing.

To test the connection speed start the service on one machine ( *target\_machine* ) and issue the following command on the other machine

```
NWTest target_machine
```

**Use this when:** a health check is being done or connection speed may be impacting response/throughput.

## 2.2.7 MiniDump

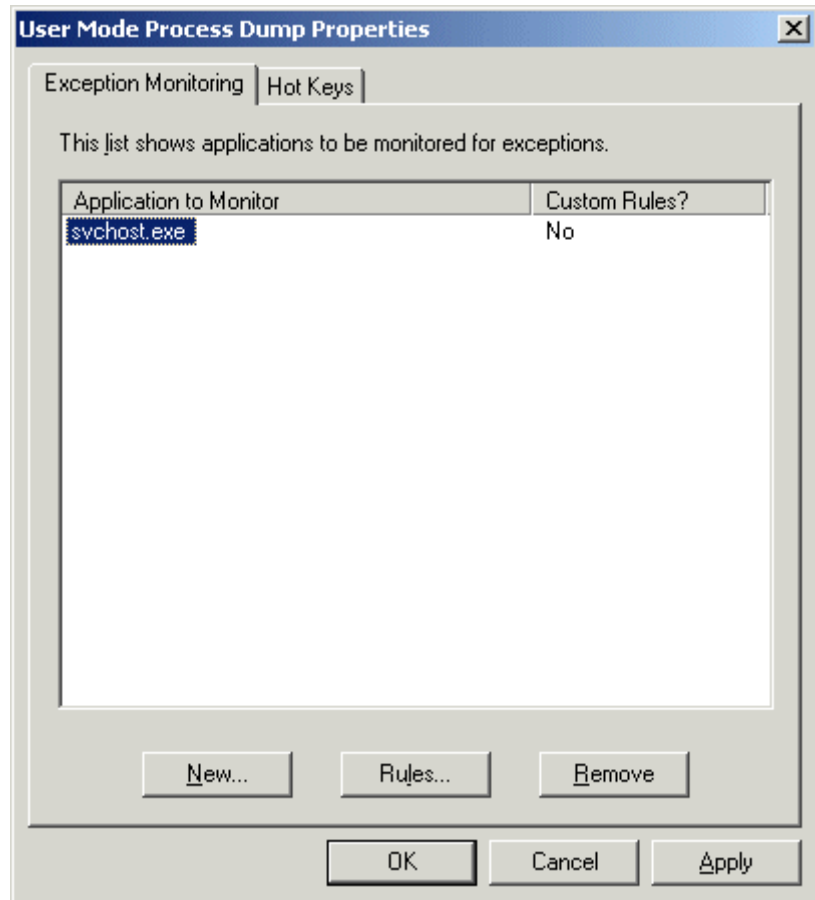
MiniDump is a set of utilities that replace Process Dump ( documented later ). Process Dump is not supported on Windows 2003 Server.

Portrait Foundation installation will define MiniDump as the Post Mortem Debugger and define the necessary registry keys to get Memory dumps created if named Portrait Foundation processes get exceptions ( whether handled by the executable or not ).

**Use this when:** This feature should be enabled permanently and each '.dmp' file should be investigated thoroughly Use WinDbg to examine the process dump file.

## 2.2.8 Process Dump applet

Figure 11 -Process Dump applet



Process Dump is a control-panel applet that allows you to specify that a process dump file should be produced by an application when it experiences certain circumstances, such as an unhandled exception.

Process Dump is part of the OEM Support Tools package, available for download from <http://www.microsoft.com>.

The Microsoft Knowledge Base article *How to Use the Userdump.exe Tool to Create a Dump File (Q242215)* describes how to download and use Process Dump.

Portrait Foundation installation will configure Process Dump and define the necessary registry keys to get Memory dumps created if named Portrait Foundation processes get exceptions ( whether handled by the executable or not ).

**Use this when:** This feature should be enabled permanently and each '.dmp' file should be investigated thoroughly Use WinDbg to examine the process dump file.

## 2.2.9 ADPlus

Figure 12 -ADPlus

```

C:\WINNT\System32\cmd.exe
C:\Program Files\Debugging Tools for Windows>adplus
Autodump+ 5.03 Usage Information
Switches: '-hang', '-quick', '-crash', '-iis', '-p <PID>', '-pn <Process Name>'
          '-pageheap', '-quiet', '-o <output directory>', '-notify <target>'
Required: '<-hang>', or '<-quick>', or '<-crash>' AND '<-iis>' or '<-p>' or '<-pn>'
Optional: '-pageheap', '-quiet', '-o <outputdir>', '-notify <computer>'
Examples: 'ADPlus -hang -iis',          Produces memory dumps of IIS and all
                                                MIS/COM+ packages currently running.
          'ADPlus -crash -p 1896', Attaches the debugger to process with PID
                                                1896, and monitors it for 1st and 2nd
                                                chance access violations (crashes).
          'ADPlus -quick -pn mmc.exe', Attaches the debugger to all instances
                                                of MMC.EXE and dumps debug information
                                                about these processes to a text file.
          'ADPlus -?' or 'ADPlus -help': Displays detailed help.
-----
For more information on using Autodump+, please refer to the following KB:
http://support.microsoft.com/support/kb/articles/q286/3/50.asp
C:\Program Files\Debugging Tools for Windows>
    
```

ADPlus is a command-line application which is capable of performing the functions of both Userdump and the Process Dump control panel applet.

ADPlus is part of the Debugging Tools for Windows package, available for download from <http://www.microsoft.com>.

The Microsoft Knowledge Base article *HOW TO: Use ADPlus to Troubleshoot "Hangs" and "Crashes"* (Q286350) describes how to download and use ADPlus.

## 2.2.10 WinDbg

Figure 13 -WinDbg

```

Dump C:\svchost.dmp - WinDbg6.0.0017.0
File Edit View Debug Window Help
Symbol search path is: srv*c:\websymbols*http://msdl.microsoft.com/download/symbols;c:\program files\ait\portrait\symbols
Loading Dump File [C:\svchost.dmp]
User Dump File: Only application data is available
Windows 2000 Version 2195 UP Free x86 compatible
Product: WinNT
System Uptime: not available
Process Uptime: not available
Symbol search path is: srv*c:\websymbols*http://msdl.microsoft.com/download/symbols;c:\program files\ait\portrait\symbols
Executable search path is:
eax=00000000 ebx=00000000 ecx=0007aa9c edx=00000000 esi=00000000 edi=0000006c
eip=77f89e7 esp=0006fc38 ebp=0005fca8 iopl=0         nv up ei pl zr na po nc
cs=001b  e8=0023  ds=0023  es=0023  fs=0038  gs=0000             efl=00000246
ntdll!ZwReadFile+
77f89e7 c22400          ret     0x24
0:000> ~*kb
0 Id: 2b4 2b0 Suspend: 0 Teb: 7ffde000 Unfrozen
ChildEBP RetAddr  Args to Child
0006fc34 77e96156 0000006c 00000000 00000000 ntdll!ZwReadFile+0xb
0006fca8 77d8752 0000006c 0006fd80 00000216 KERNEL32!ReadFile+0x181
0006fd04 77d47fd0 0000006c 0006fd80 00000216 ADVAPI32!ScSetPipeInput+0x20
0006fd50 77d47f5f 0000006c 0006fd80 00000216 ADVAPI32!ScDispatcherLoop+0x4a
0006ffb0 0100113d 00074dd0 77fa7ef2 77e8b65c ADVAPI32!StartServiceCtrlDispatcherW+0xf6
0006ffc0 77ea847c 77fa7ef2 77e8b65c 77fd0000 svchost!mainCRTStartup+0x81
0006fff0 00000000 010010b8 00000000 000000c8 KERNEL32!BaseProcessStart+0x3d
1 Id: 2b4 2bc Suspend: 0 Teb: 7ffdd000 Unfrozen
ChildEBP RetAddr  Args to Child
0043ff0c 77ea9b35 00000098 00000000 00000000 ntdll!NtWaitForSingleObject+0xb
0043ff34 77e8b32b 00000098 ffffffff 00000000 KERNEL32!WaitForSingleObjectEx+0x71
0043ff44 762918ca 00000098 ffffffff 00000000 KERNEL32!WaitForSingleObjectEx+0xf
0043ff54 762917d6 0007be5c 00000000 01003000 esi!RunService+0xc2
0:000>
Ln 0, Col 0 Sys 0:C:\svch Proc 000:2b4 Thrd 000:2b0 [ASM] [SRV] [CAPS] [NUM]
    
```

WinDbg is a powerful symbolic debugger that can be used to examine a process dump file or to attach to a running process. To use WinDbg requires that you have access to the symbol (.pdb) files (containing information about variables, functions and their source line numbers) for the components which comprise the process. To use the source-level view of WinDbg requires that you also have access to the source code for the components being debugged. Symbol files for all Portrait Foundation product components are shipped with the product, but source code is not. Source-level debugging is therefore limited to project-specific components where both symbol and source information is available.

Portrait have developed a body of WinDBG extensions that can be used to help analysis of general or Portrait Foundation specific parts of a memory dump. For

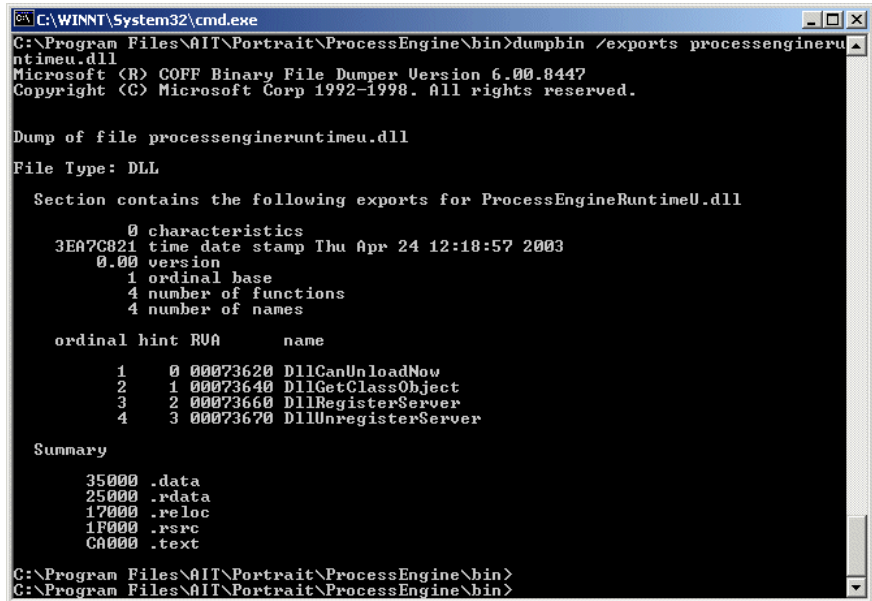
example there is an extension that will extract the contents of the Waparound log destination and create a .plf file for viewing in the Portrait Log Viewer.

WinDbg is part of the Debugging Tools for Windows package, available for download from <http://www.microsoft.com>.

**Use this when:** you have obtained a process dump file that you wish to analyse, particularly for unhandled exceptions or deadlock problems, or you want to study the workings of a running process.

### 2.2.11 DumpBin

Figure 14 -DumpBin



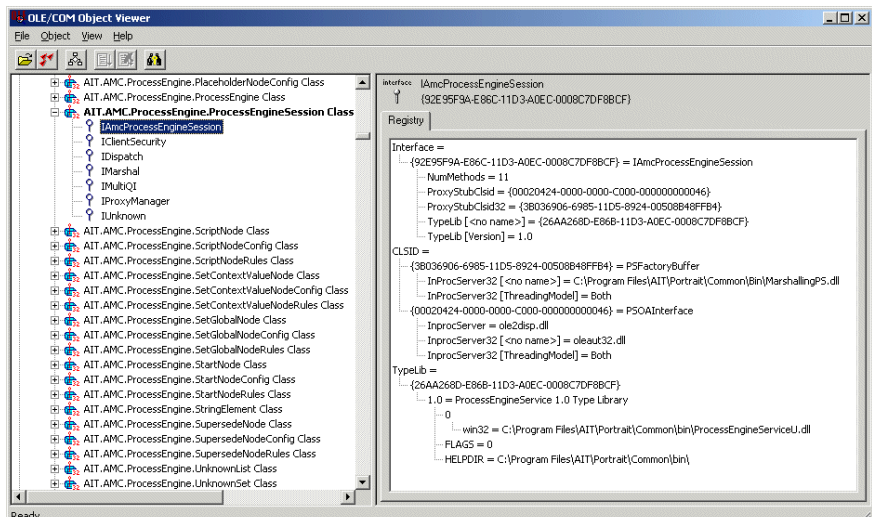
DumpBin is a command-line application which displays information about 32-bit Common Object File Format (COFF) binary files. You can use DumpBin to examine COFF object files, standard libraries of COFF objects, executable files, and dynamic-link libraries (DLLs). Running DumpBin with no command-line options causes it to display a usage statement that summarizes its options.

DumpBin is included with the Visual Studio package.

**Use this when:** you want to obtain detailed information about a binary file.

### 2.2.12 OLEView

Figure 15 -OLEView





OLEView is an application which allows you to view information about registered COM components and interfaces. OLEView uses information in the registry to display lists of all registered interfaces, all registered components and all the registered interfaces that the components support. Double-clicking a component name in the display causes OLEView to do a CoCreateInstance on the component. OLEView can also be used to read information from a type-library file (TLB, DLL or EXE). OLEView can be very useful in determining the cause of problems that relate to registration of COM components and interfaces.

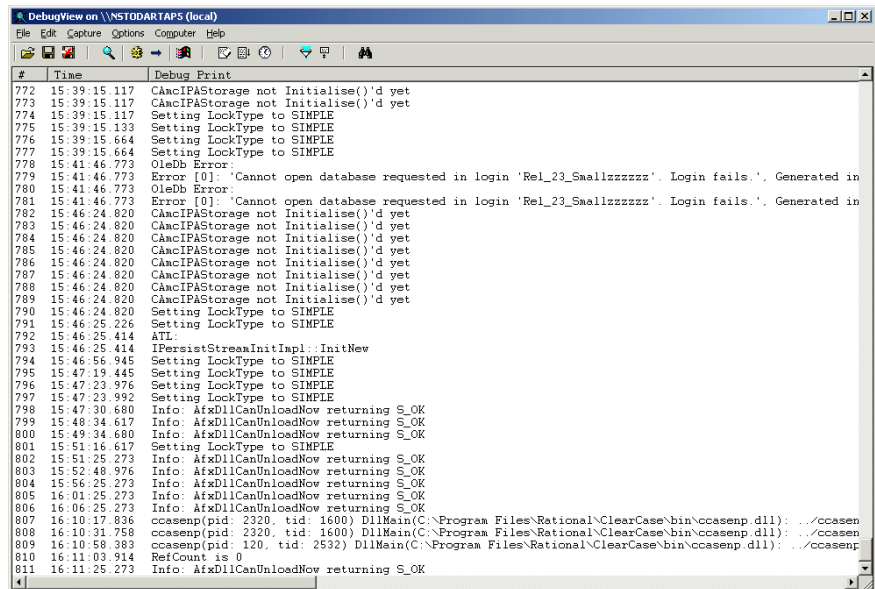
OLEView is included with the Visual Studio package.

**Use this when:** you want to obtain information about registered COM components and interfaces.

## 2.3 Other third-party tools

### 2.3.1 SysInternals DebugView

Figure 16 - SysInternals DebugView



DebugView is an application that intercepts calls made to DbgPrint by device drivers and OutputDebugString made by Win32 programs. It allows for viewing and recording of debug session output on your local machine or across the Internet without an active debugger.

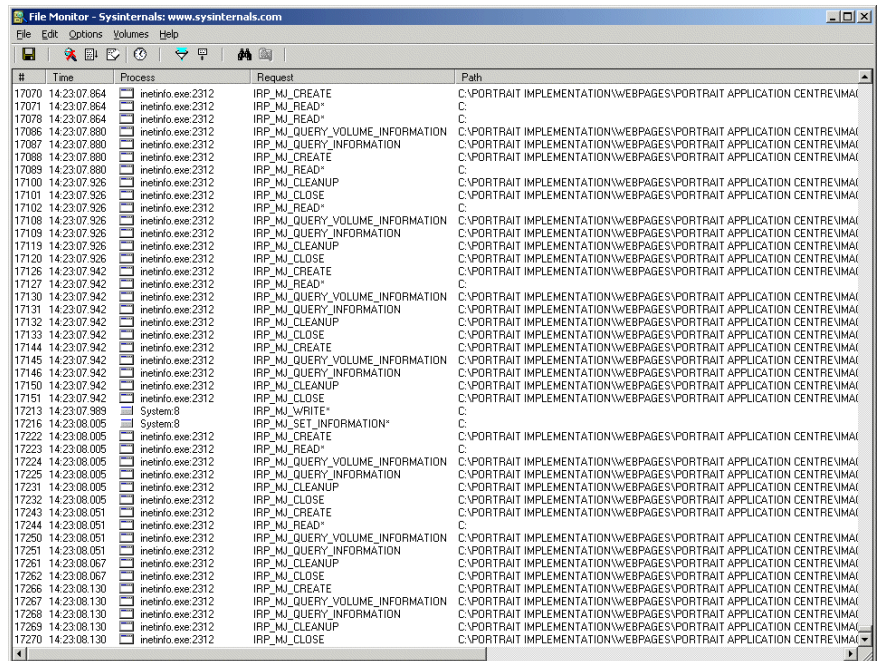
DebugView is freely available for download from <http://technet.microsoft.com/en-gb/sysinternals/default.aspx>.

Some components, for example, the ASP.NET runtime exception handler, produce debug output when a serious error occurs.

From Portrait Foundation Release 2.4 onwards, the product includes a Portrait Log Viewer, which can intercept and display the type of data shown by DebugView. The Portrait Log Viewer allows the user to view messages in a formatted list control that supports filtering, highlighting, searching and saving. There is therefore no longer any need to use DebugView.

### 2.3.2 SysInternals File Monitor

Figure 17 - SysInternals File Monitor



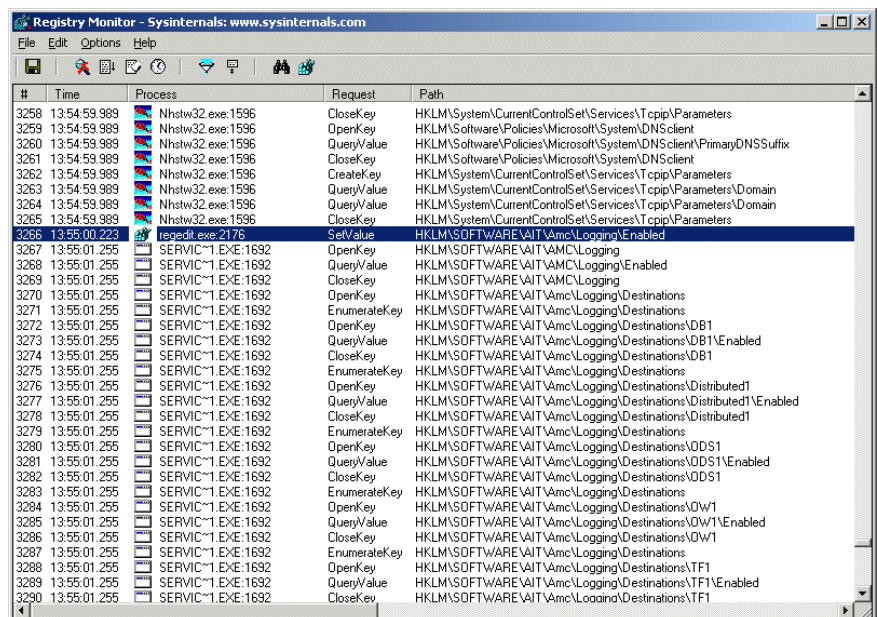
File Monitor is an application that monitors and displays file system activity on a system in real-time.

File Monitor is freely available for download from <http://technet.microsoft.com/en-gb/sysinternals/default.aspx>

**Use this when:** you want to obtain detailed information on access to files, for example because you suspect that excessive file access is contributing to a performance problem.

### 2.3.3 SysInternals Registry Monitor

Figure 18 - SysInternals Registry Monitor



Registry Monitor is an application that monitors and displays registry activity on a system in real-time.

Registry Monitor is freely available for download from <http://technet.microsoft.com/en-gb/sysinternals/default.aspx>.

**Use this when:** you want to obtain detailed information on access to the registry, for example because you suspect that excessive registry access is contributing to a performance problem.

## 2.4 Other resources

### 2.4.1 Portrait Known Issues list

For each Portrait Foundation release, a list of known issues is published, some of which will have workarounds. Compare the nature and circumstances of the problem with the descriptions of known issues.

**Use this when:** you have a problem affecting a minority of Portrait Foundation functions and you have confirmed that the Portrait Foundation environment and installation is correct and that you are using the function correctly.

### 2.4.2 Portrait Product Support

Consult the Portrait Product Support documentation for details of how to use the service. In order for Portrait Product Support to provide assistance, you will need to provide the nature and circumstances of the problem and full details of the product components in use and the environment in which they are installed. You may be asked to supply Portrait log files, process dump files, registry export files or other information to assist with the problem analysis.

**Use this when:** you have exhausted the tools and techniques available to you.

### 2.4.3 Microsoft Developer Network (MSDN) and Microsoft Knowledge Base

MSDN is an extremely valuable source of information on Microsoft products and also on general debugging techniques.

The MSDN article *Troubleshooting Common Problems with Applications: Debugging in the Real World* gives further ideas on questions to consider and debugging approaches to take.

The MSDN article *Debugging Distributed Web Applications* provides a good starting point for those getting started with advanced debugging tools like WinDbg and Userdump.

The Microsoft Knowledge Base article *Microsoft Debugging Tools Knowledge Base Articles* (<http://www.microsoft.com/whdc/ddk/debugging/dbg-kb.msp>) provides a collection of Microsoft Product Support links that are relevant for debugging.

## 3 Other tools and techniques

Some of the software development tools and techniques used by the Portrait Foundation product development and support teams and recommended for Portrait Foundation implementations are listed below. These tools are better suited to guaranteeing the quality and robustness of software during the development and testing process rather than problem determination after it goes into production.

### 3.1 Code quality checking

Manual code reviews are extremely useful in identifying code quality problems. In addition to manual methods, several tools have been used with great success to detect certain classes of quality problem.

#### 3.1.1 GloCode

We have a license for this product. It is a very basic memory leak / profiling tool. It is very much a lightweight version of BoundsChecker.

#### 3.1.2 DebugDiag

This tool ( and its predecessor LeakDiag ) are memory leak detection tools. Both tools can be found on the Portrait Foundation offsite CD image.

#### 3.1.3 NuMega BoundsChecker

BoundsChecker is capable of identifying a wide range of errors, including memory leaks and memory usage problems. It is easy to use and well suited to use during the development cycle.

#### 3.1.4 Gflags, UMDH and DHCMP

These tools from Microsoft are designed to aid in the detection of memory leaks. Using these tools can reveal the call stack for any heap memory allocation that has been made but not freed.

These tools are part of the Debugging Tools for Windows package, available for download from <http://www.microsoft.com>.

#### 3.1.5 PageHeap

Microsoft PageHeap is designed to detect errors in heap memory usage, such as buffer overruns. Once enabled, it throws an exception immediately that a heap memory error occurs. This makes it much easier to identify the cause of problems that might otherwise be manifested in very obscure ways.

PageHeap is part of the Windows Application Compatibility Toolkit 3.0, available for download from <http://www.microsoft.com>.

Its use is documented in the document Debugging Tools.

### 3.2 Load testing

It is advised that a project should use load testing tools to create repeatable tests for volume and throughput testing to identify any issues only apparent under load, especially in a 24x7 high volume environment.

### 3.2.1 Portrait Scenario Director

This tool is available in versions of Portrait Foundation from 2.4 onwards. Portrait Scenario Director is an application that allows a user to record and replay Portrait Foundation implementation scenarios, simulating one or more concurrent users exercising normal business activity.

### 3.2.2 Mercury Interactive LoadRunner

LoadRunner has been used extensively in the Portrait Foundation product performance lab to exercise parts of the product under high load conditions.