

Active Server Availability Feedback

James Hamilton

JamesRH@microsoft.com

Microsoft SQL Server

2002.06.12

SQL Server

Agenda

- ◆ **Availability**
 - **Software complexity**
 - **Availability study results**
- ◆ **System Failure Reporting (Watson)**
 - **Goals**
 - **System architecture**
 - **Operation & mechanisms**
 - **Querying failure data**
- ◆ **Data Collection Agent (DCA)**
 - **Goals**
 - **System architecture**
 - **What is tracked?**
 - **Progress & results**

S/W Complexity

- ◆ **Even server-side software is BIG:**
 - Windows2000: over 50 mloc
 - DB: 1.5+ mloc
 - SAP: 37 mloc (4,200 S/W engineers)
- ◆ **Tester to Developer ratios often above 1:1**
 - Quality per unit line only incrementally improving
 - Current massive testing investment not solving problem
- ◆ **New approach needed:**
 - Assume S/W failure inevitable
 - Redundant, self-healing systems right approach
 - We first need detailed understanding of what is causing both downtime

Availability Study Results

- ◆ **1985 Tandem study (Gray):**
 - Administration: 42% downtime
 - Software: 25% downtime
 - Hardware 18% downtime
- ◆ **1990 Tandem Study (Gray):**
 - Administration: 15%
 - Software 62%
 - Most studies have admin contribution much higher
- ◆ **Observations:**
 - H/W downtime contribution trending to zero
 - Software & admin costs dominate & growing
 - We're still looking at 10 to 15 year-old research

Agenda

- ◆ **Availability**
 - Software complexity
 - Availability study results
- ◆ **System Failure Reporting (Watson)**
 - Goals
 - System architecture
 - Operation & mechanisms
 - Querying failure data
- ◆ **Data Collection Agent (DCA)**
 - Goals
 - System architecture
 - What is tracked?
 - Progress & results

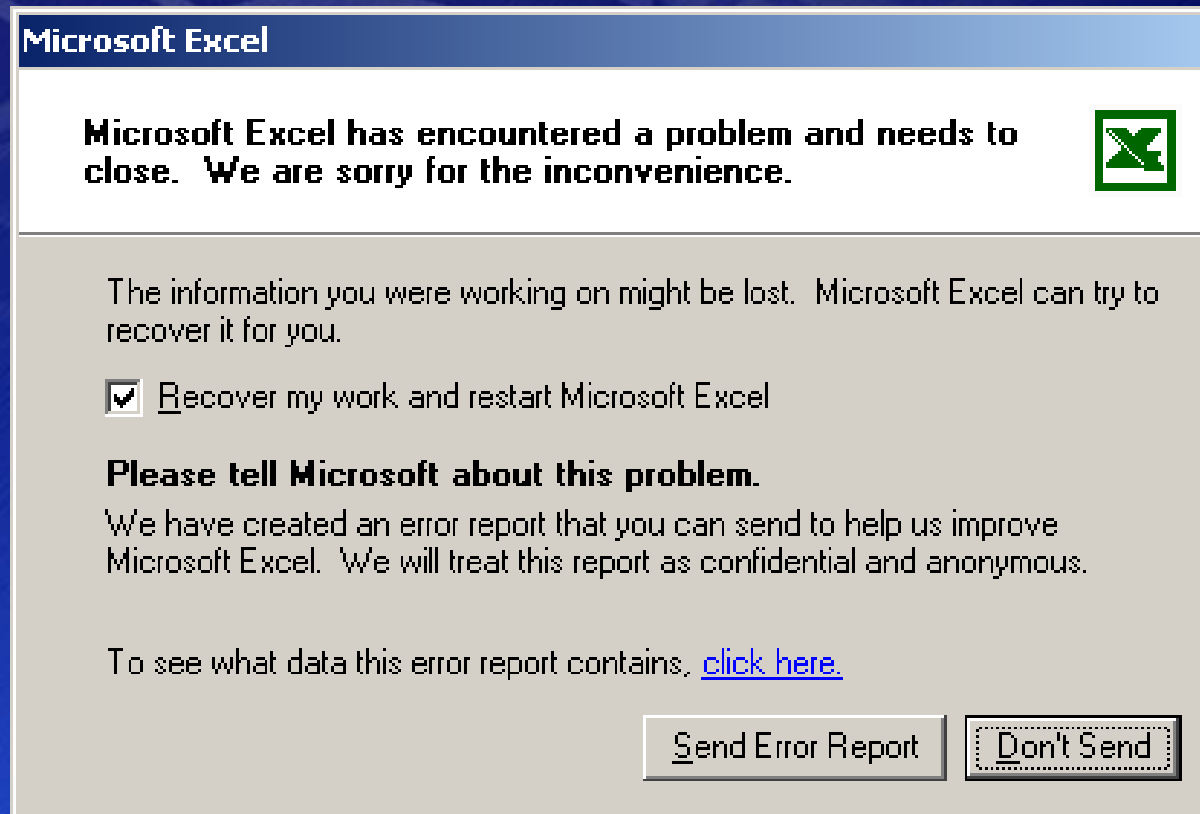
Watson Goals

- ◆ **Instrument SQL Server:**
 - Track failures during customer usage
 - Report failure & debug data to dev team
 - Goal is to fix big ticket issues proactively
- ◆ **Instrumented components:**
 - Setup
 - Core SQL Server engine
 - Replication
 - OLAP Engine
 - Management tools
- ◆ **Also in use by:**
 - Office (Watson technology owner)
 - Windows XP
 - Internet Explorer
 - MSN Explorer
 - Visual Studio 7
 - ...

What data do we collect?

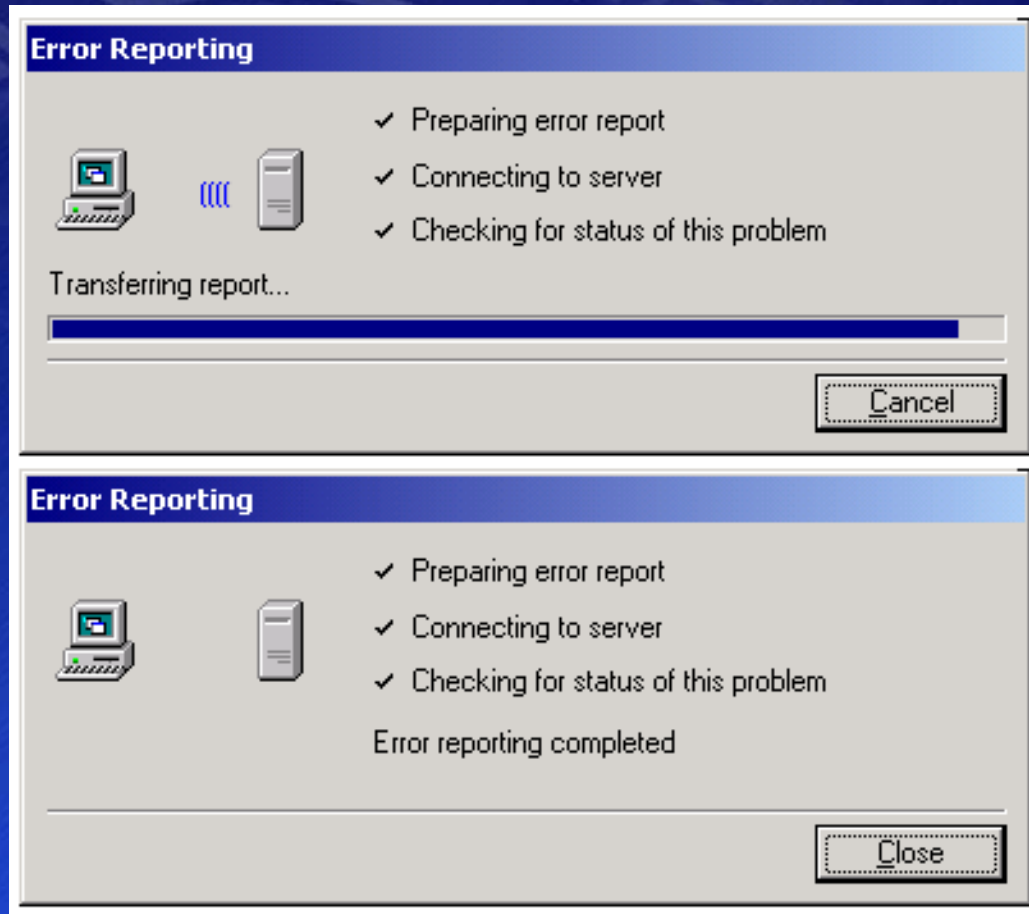
- ◆ **For crashes: Minidumps**
 - Stack, System Info, Modules-loaded, Type of Exception, Global/Local variables
 - 0-150k each
- ◆ **For setup errors:**
 - Darwin Log
 - setup.exe log
- ◆ **2nd Level if needed by bug-fixing team:**
 - Regkeys, heap, files, file versions, WQL queries

Watson user experience:



- Server side is registry key driven rather than UI
- Default is “don’t send”

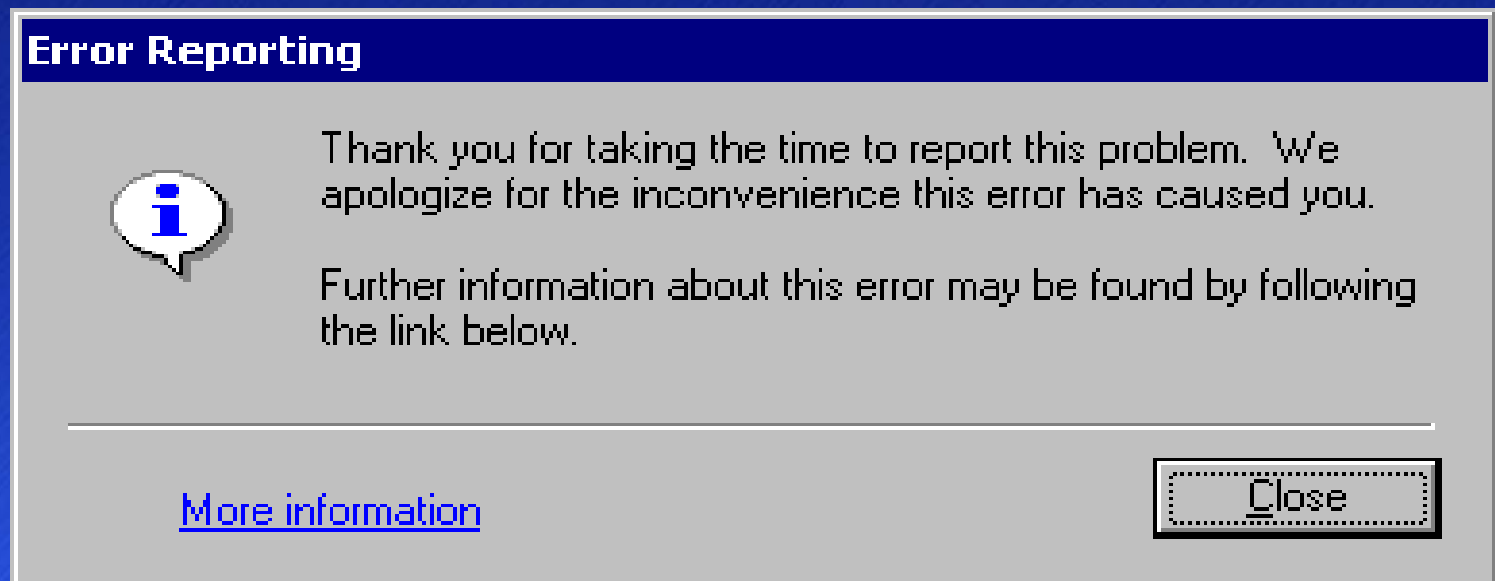
Crash Reporting UI



- Server side upload events written to event log rather than UI

information back to users

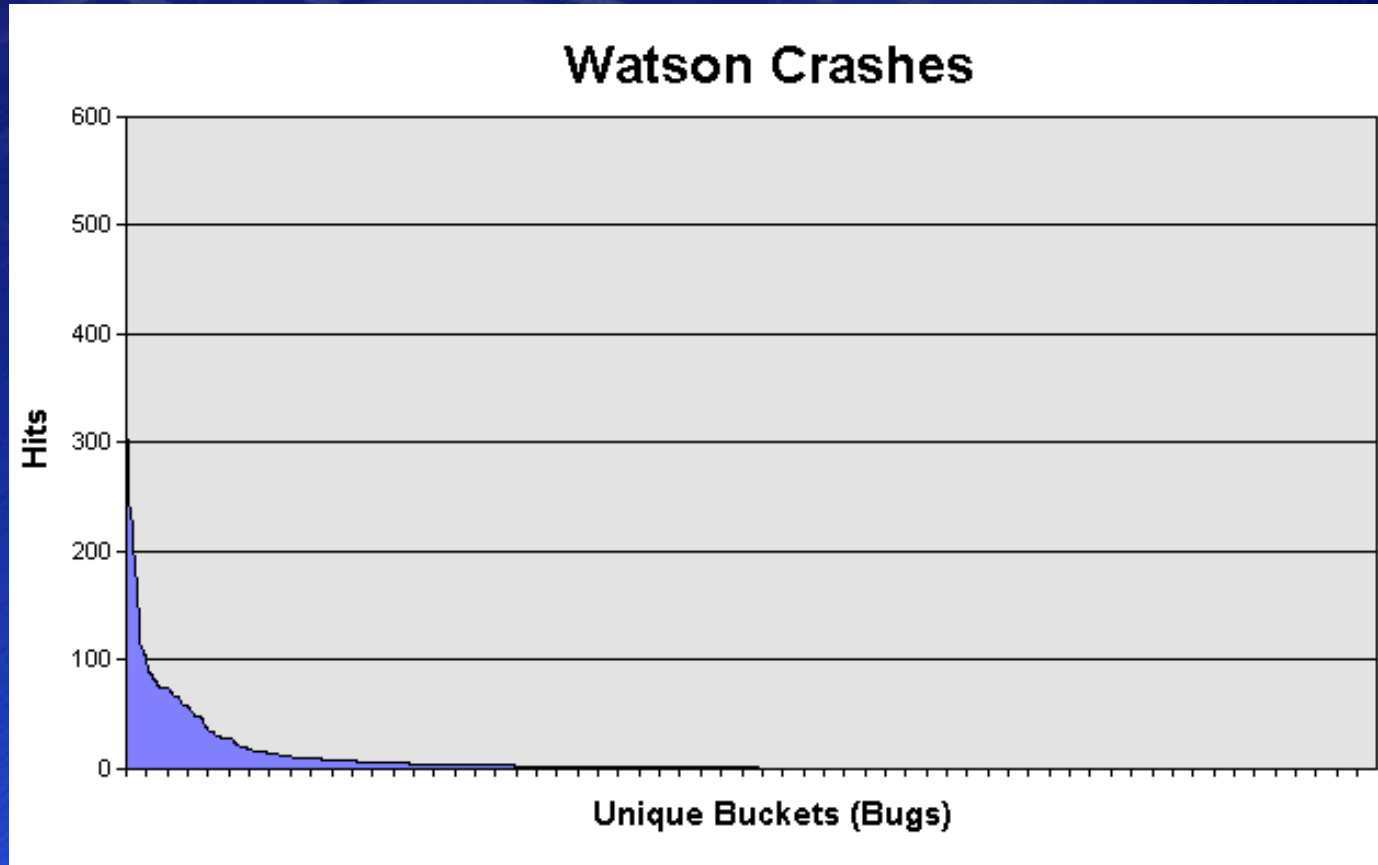
- ◆ ‘More information’ hyperlink on Watson’s Thank You dialog can be set to problem-specific URL



Key Concept: Bucketing

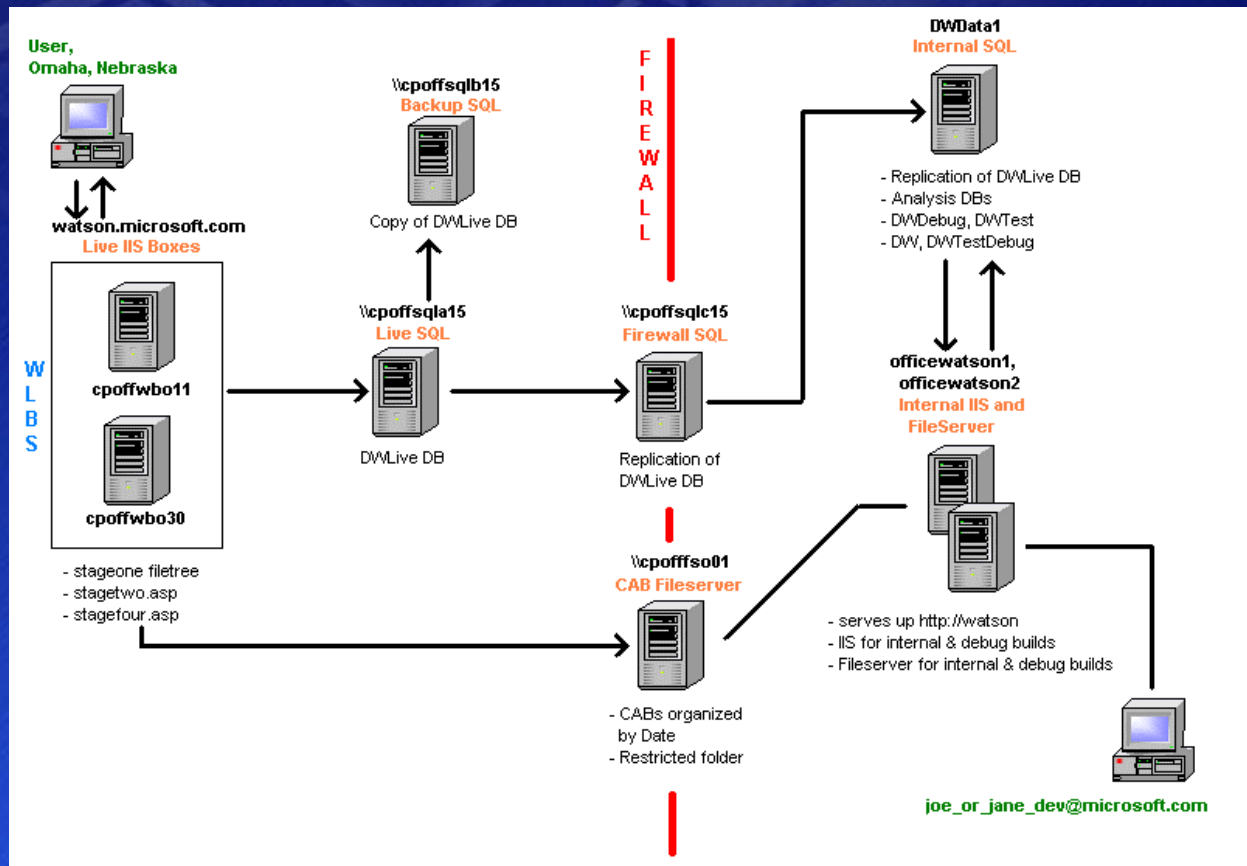
- ◆ **Categorize & group failures by certain 'bucketing parameters':**
 - **Crash: AppName, AppVersion, ModuleName, ModuleVersion, Offset into module...**
 - **SQL uses stack signatures rather than failing address as buckets**
 - **Setup Failures: ProdCode, ProdVer, Action, ErrNum, Err0, Err1, Err2**
- ◆ **Why bucketize?**
 - **Ability to limit data gathering**
 - **Per bucket hit counting**
 - **Per bucket server response**
 - **Custom data gathering**

The payoff of bucketing



- Small number of S/W failures dominate customer experienced failures

Watson's Server Farm



Watson Bug Report Query

WATSON Refresh 5

Database Table Dumps

Raw Crash Table: <choose>	Crash 'bucket' table.
Grouped Crash Table: <choose>	Table grouped by szSymbol.
Grouped Crash Chart: <choose>	Chart grouped by szSymbol.
Raid Reports: <detailed> <summary>	RAID bugs linked to crash buckets.
Setup Failures: <choose>	Setup failures.
AssertTags: <choose>	Assert tags and RAID bugs.
Miscellaneous: <choose>	Other database tables.
Office Watsonized Alerts: <choose>	Office Watsonized Alerts
Office Ship Asserts: <choose>	Office Ship Asserts
Build a Query: Construct Query	Query for buckets meeting your criteria.

Database Drilldown Queries

Enter a BUCKET: Live Crash View Bucket

Enter a BUCKET: Live Crash View Calltree

Enter a SETUP FAILURE: Live View Setup

Enter an ASSERTTAG: OfficeNet View Assert

Enter ANY of the following to see matching BUCKET(s):

Application Name (szAppName):	sqlservr.exe
Application Version (szAppVer):	
Module Name (szModName):	
Module Version (szModVer):	
Offset into Module (offset):	
Function name fragment (szSymbol):	

Show Bucket(s) Use internal data

Administration [links](#)

Done Local intranet

Watson Tracking Data

D.W. Bucket Table - Microsoft Internet Explorer

Address [http://watson/genlist.aspx?Table=bucketall&Database=5&W=\(szAppName=sqlservr.exe\)](http://watson/genlist.aspx?Table=bucketall&Database=5&W=(szAppName=sqlservr.exe))

szAppName	szAppVer	szModName	szModVer	offset	cHits	cCabs	iRaidBug	szSymbol
39722	sqlservr.exe	9.0.354	sqlservr.exe	2000.90.354	B7E65D3F	3		
39715	sqlservr.exe	9.0.354	rsvpsp.dll	5.2.3621.0	730F03A8	1	1	RSVPSP.DLL!GetHandleAndDeleteAcceptEx
39673	sqlservr.exe	9.0.353	sqlservr.exe	2000.90.353.0	EE774AB80	10	10	SQLSERVER.EXE!HoBtSchemaFactory::PersistVisibleMetadata
39675	sqlservr.exe	9.0.353	sqlservr.exe	2000.90.353.0	EE771885	10	10	SQLSERVER.EXE!HoBtSchemaFactory::PersistVisibleMetadata
39618	sqlservr.exe	9.0.353	ntdll.dll	5.0.2195.2779	4ACA04D4	7	7	NTDLL.DLL!NtGetContextThread
39667	sqlservr.exe	9.0.353	ntdll.dll	5.0.2195.2779	D5BED539	4	4	NTDLL.DLL!NtGetContextThread
39685	sqlservr.exe	9.0.353	sqlservr.exe	2000.90.353.0	4D48E6BE	4	4	SQLSERVER.EXE!CBulkText::ReadTextCol
39668	sqlservr.exe	9.0.353	ntdll.dll	5.0.2195.4929	DD5D4AC3	3	3	NTDLL.DLL!0x0000843f
39666	sqlservr.exe	9.0.353	sqlservr.exe	2000.90.353.0	F01E7298	3	2	SQLSERVER.EXE!CTraceController::Shutdown
39654	sqlservr.exe	9.0.353	sqlservr.exe	2000.90.353	83BDFCE4	2		
39665	sqlservr.exe	9.0.353	sqlservr.exe	2000.90.353.0	F010BCC2	2	2	SQLSERVER.EXE!CTraceController::Shutdown
39670	sqlservr.exe	9.0.353	sqlservr.exe	2000.90.353.0	4D480962	2	2	SQLSERVER.EXE!CBulkText::ReadTextCol
39725	sqlservr.exe	9.0.353	sqlservr.exe	2000.90.353	A4286C68	2		
39657	sqlservr.exe	9.0.353	ntdll.dll	5.0.2195.4929	19954A28	1	1	NTDLL.DLL!0x0000843f
39659	sqlservr.exe	9.0.353	ntdll.dll	5.0.2195.2779	CEA4F9F4	1	1	NTDLL.DLL!NtGetContextThread
39660	sqlservr.exe	9.0.353	ntdll.dll	5.0.2195.2779	13731196	1	1	NTDLL.DLL!NtGetContextThread
39669	sqlservr.exe	9.0.353	ntdll.dll	5.0.2195.4929	259E126A	1	1	NTDLL.DLL!0x0000843f
39674	sqlservr.exe	9.0.353	ntdll.dll	5.0.2195.2779	C462B035	1	1	NTDLL.DLL!NtGetContextThread
39677	sqlservr.exe	9.0.353	rsvpsp.dll	5.2.3621.0	730B03A8	1	1	RSVPSP.DLL!GetHandleAndDeleteAcceptEx
39619	sqlservr.exe	9.0.353	sqlservr.exe	2000.90.353.0	A17D5C13	1	1	SQLSERVER.EXE!0x0006aaade
39664	sqlservr.exe	9.0.353	sqlservr.exe	2000.90.353.0	F01E05FF	1	1	SQLSERVER.EXE!CTraceController::Shutdown
39671	sqlservr.exe	9.0.353	sqlservr.exe	2000.90.353.0	A8907FC3	1	1	SQLSERVER.EXE!CQueryResourceGrantManager::ReturnGrant
39672	sqlservr.exe	9.0.353	sqlservr.exe	2000.90.353.0	BF22DEA7	1	1	SQLSERVER.EXE!CMEDAccess::GetDatabaseFromId
39676	sqlservr.exe	9.0.353	sqlservr.exe	2000.90.353.0	1A4C0358	1	1	SQLSERVER.EXE!HoBtSchemaFactory::PersistVisibleMetadata
39701	sqlservr.exe	9.0.353	sqlservr.exe	2000.90.353.0	4D4528E4	1	1	SQLSERVER.EXE!CBulkText::ReadTextCol
39705	sqlservr.exe	9.0.353	sqlservr.exe	2000.90.353.0	125D9E88	1	1	SQLSERVER.EXE!CBulkText::ReadTextCol
39562	sqlservr.exe	9.0.352	ntdll.dll	5.0.2195.2779	139D59BD	20	20	NTDLL.DLL!NtGetContextThread
39551	sqlservr.exe	9.0.352	ntdll.dll	5.0.2195.2779	1C2732F1	7	7	NTDLL.DLL!NtGetContextThread
39620	sqlservr.exe	9.0.352	ntdll.dll	5.0.2195.2779	BF5BF282	7	7	NTDLL.DLL!NtGetContextThread
39555	sqlservr.exe	9.0.352	sqlservr.exe	2000.90.352.0	9BF3E9F1	6	6	SQLSERVER.EXE!CSbXmitState::CacheXmitQueueMetadata
39629	sqlservr.exe	9.0.352	sqlservr.exe	2000.90.352.0	689D0707	5	5	SQLSERVER.EXE!CCursorWorktable::GetTEMPDBXdes
39559	sqlservr.exe	9.0.352	sqlservr.exe	2000.90.352.0	711FB752	4	4	SQLSERVER.EXE!HoBtSchemaFactory::GetHoBtSchemaAccess
39587	sqlservr.exe	9.0.352	rsvpsp.dll	5.2.3621.0	730B03A8	3	3	RSVPSP.DLL!GetHandleAndDeleteAcceptEx
39721	sqlservr.exe	9.0.352	sqlservr.exe	2000.90.352.0	B6FBDBD6	3	3	SQLSERVER.EXE!CSbXmitState::CacheXmitQueueMetadata
39556	sqlservr.exe	9.0.352	ssnetlib.dll	2000.90.352.0	FC58AC51	3	3	SSNETLIB.DLL!AcceptFunc
39554	sqlservr.exe	9.0.352	ntdll.dll	5.0.2195.2779	3D8260FF	2	2	NTDLL.DLL!NtGetContextThread
39597	sqlservr.exe	9.0.352	sqlservr.exe	2000.90.352.0	6230E838	1	1	SQLSERVER.EXE!0x010990ea
39598	sqlservr.exe	9.0.352	sqlservr.exe	2000.90.352.0	FDB0F56E	1	1	SQLSERVER.EXE!0x010990ea
39599	sqlservr.exe	9.0.352	sqlservr.exe	2000.90.352.0	7A372A81	1	1	SQLSERVER.EXE!0x012b1483
39600	sqlservr.exe	9.0.352	sqlservr.exe	2000.90.352.0	6C308945	1	1	SQLSERVER.EXE!0x012b1483
39621	sqlservr.exe	9.0.352	ntdll.dll	5.0.2195.2779	F60E3CD1	1	1	NTDLL.DLL!NtGetContextThread
39558	sqlservr.exe	9.0.352	rsvpsp.dll	5.2.3621.0	2525B635	1	1	RSVPSP.DLL!HashAcceptEx
39645	sqlservr.exe	9.0.352	rsvpsp.dll	5.2.3621.0	E5D5E081	1	1	RSVPSP.DLL!HashAcceptEx
39573	sqlservr.exe	9.0.352	rsvpsp.dll	5.1.3604.0	7309034B	1	1	RSVPSP.DLL!0x0001034b
39552	sqlservr.exe	9.0.352	sqlservr.exe	2000.90.352.0	33906AD3	1	1	SQLSERVER.EXE!0x0006aaade

Watson Drill Down

Bucket 39673 - Microsoft Internet Explorer

Address: http://watson/ViewBucket.aspx?Database=58iBucket=39673

Watson (Internal) Bucket '39673'

[Home] [Debug Tips]

Callstack Tree
 Raid Bugs
 Survey Responses
 Hits By Day
 DataWanted Table
 Cab Table
 Update Bucket Values
 [Config](#)

DW.NET Bucket Table

iBucket	szAppName	szAppVer	szModName	szModVer	offset	cHits	cDataWanted	szResponse
39673	sqlservr.exe	9.0.353	sqlservr.exe	2000.90.353	EE77ABB0	10	0	1
Filename	Version	szSymbol	FrameLevel	cCabs	cHits (estimated)	iRaidBug		
sqlservr.exe	2000.90.353.0	SQLSERVER.EXE!HoBtSchemaFactory::PersistVisibleMetadata	0	10	10			

szStageOne: \StageOne\sqlservr_exe\9_0_353\sqlservr_exe\2000_90_353\EE77ABB0.htm

Callstack Tree

Outline level: 0 All

- (0) SQLSERVER.EXE ! HoBtSchemaFactory::PersistVisibleMetadata + 0x1ef (10 cabs)
 - (1) SQLSERVER.EXE ! HoBtSchemaFactory::CreateHoBtSchema + 0x2eb
 - (2) SQLSERVER.EXE ! SECreateRowset + 0x745
 - (3) SQLSERVER.EXE ! CIndexDDL::CreateRowsets + 0x87c
 - (4) SQLSERVER.EXE ! CIndexDDL::ConstructIndices + 0x372
 - (5) SQLSERVER.EXE ! CIndexDDL::Create + 0x135
 - (6) SQLSERVER.EXE ! CIndexDef::CreateIndex + 0xf1c
 - (7) SQLSERVER.EXE ! CStmtCreateIndex::XretExecute + 0x9fd
 - (8) SQLSERVER.EXE ! CExecuteStatement::XretExecute + 0x17
 - (9) SQLSERVER.EXE ! CSqlExecContext::ExecuteStmts + 0x5eb
 - (10) SQLSERVER.EXE ! CSqlExecContext::FExecute + 0x538
 - (11) SQLSERVER.EXE ! CSQLSource::Execute + 0x621
 - (12) SQLSERVER.EXE ! CLanguageExecEnv::EcrExecuteCommandSpecific + 0x10e
 - (13) SQLSERVER.EXE ! CBatchExecEnv::EbelExecuteCommand + 0x12f
 - (14) SQLSERVER.EXE ! CBatchExecEnv::EcrProcessCommand + 0x7df
 - (15) SQLSERVER.EXE ! process_request + 0x6f
 - (16) SQLSERVER.EXE ! process_request_helper + 0x15c
 - (17) SQLSERVER.EXE ! process_commands + 0x114
 - (18) SQLSERVER.EXE ! SOS_Task::Param::Execute + 0x75
 - (19) SQLSERVER.EXE ! SOS_Scheduler::RunTask + 0x12d
 - (20) SQLSERVER.EXE ! SOS_Scheduler::ProcessTasks + 0x120

Link to Bug: Database: Office10 szSymbol: <Same bug for every szSymbol>

Cab Table

| | Filename | DateCreated | Size | OSInfo | Exception |
|---|-----------------------|---------------------|-------|-----------------------------|-----------------------------|
| 1 | 6_5_2002\04925021.cab | 6/5/2002 9:41:09 PM | 31485 | Windows XP RTM (build 2600) | c0000005 (access violation) |
| 2 | 6_5_2002\84727643.cab | 6/5/2002 9:41:12 PM | 32791 | Windows XP RTM (build 2600) | c0000005 (access violation) |
| 3 | 6_5_2002\45831499.cab | 6/5/2002 9:41:15 PM | 32471 | Windows XP RTM (build 2600) | c0000005 (access violation) |
| 4 | 6_5_2002\84098768.cab | 6/5/2002 9:41:18 PM | 32995 | Windows XP RTM (build 2600) | c0000005 (access violation) |

Local intranet

Agenda

- ◆ **Availability**
 - Software complexity
 - Availability study results
- ◆ **System Failure Reporting (Watson)**
 - Goals
 - System architecture
 - Operation & mechanisms
 - Querying failure data
- ◆ **Data Collection Agent (DCA)**
 - Goals
 - System architecture
 - What is tracked?
 - Progress & results

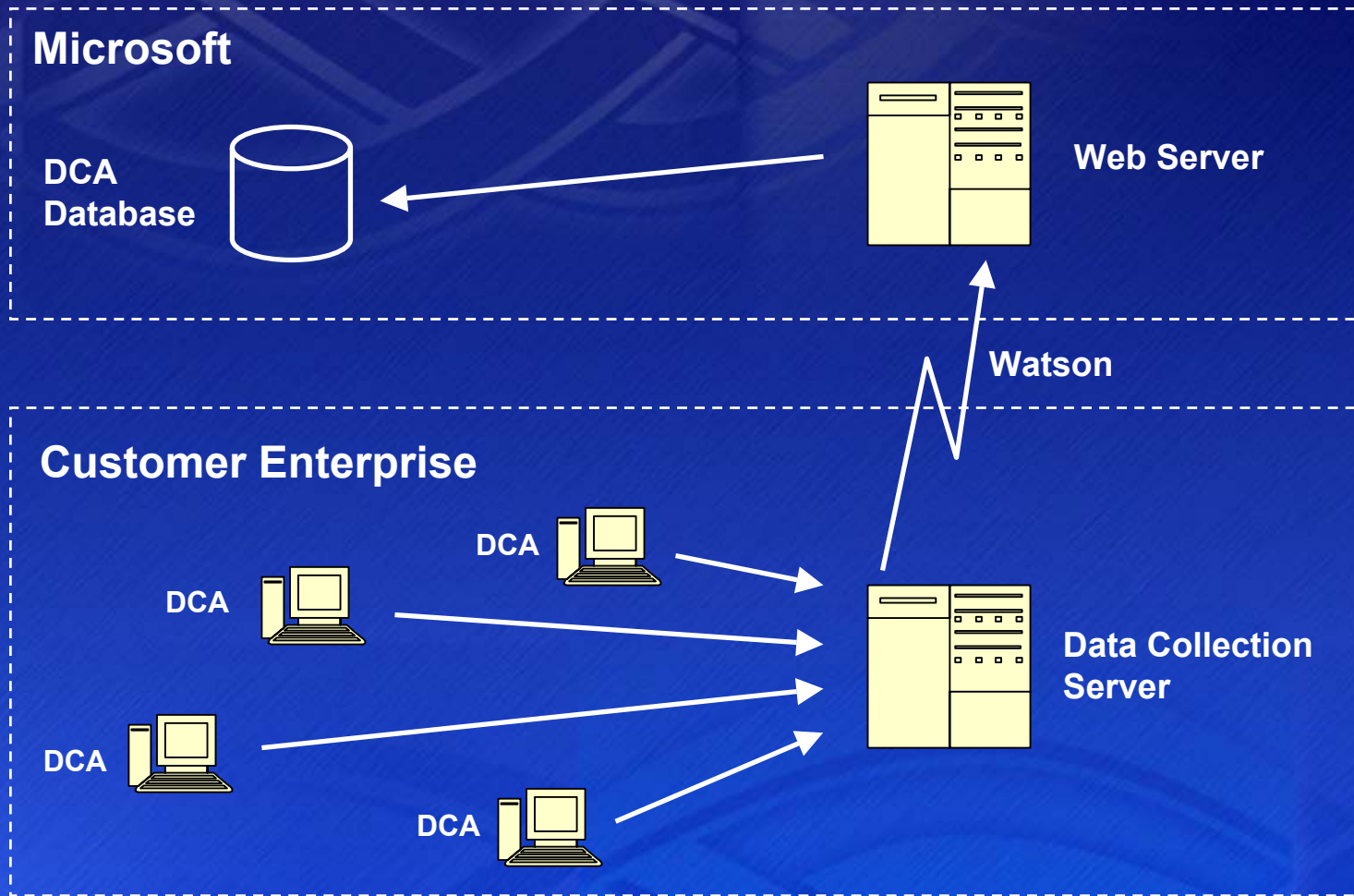
Data Collection Agent

- ◆ **Premise: can't fix what is not understood**
 - Even engineers with significant time with customers typically know less than 10 really well
- ◆ **Goal: Instrument systems intended to run 24x7**
 - Obtain actual customer uptime
 - Learn causes of system downtime – drive product improvement
 - Model after EMC & AS/400 “call home” support
 - Influenced by Brendan Murphy work on VAX availability
 - Track release-to-release improvements
 - Reduce product admin and service costs
 - Improve customer experience with product
 - Debug data available on failed systems for service team
- ◆ **Longer term Goal:**
 - Two way communications
 - Dynamically change metrics being measured
 - Update software
 - Proactively respond to failure with system intervention
 - Services offering with guaranteed uptime

DCA Operation

- ◆ **Operation:**
 - System state at startup
 - Snapshot select metrics each minute
 - Upload last snapshot every 5 min
 - On failure, upload last 10 snapshots & error data
- ◆ **Over 100 servers currently under management:**
 - Msft central IT group (ITG)
 - Goal: to make optional part of next release
- ◆ **Four tier system:**
 - Client: running on each system under measurement
 - Mid-tier Server: One per enterprise
 - Transport: Watson infrastructure back to msft
 - Server: Data stored into SQL Server for analysis

DCA Architecture



Startup: O/S and SQL Configuration

- ◆ Operating system version and service level
- ◆ Database version and service level
 - Syscurconfigs table
- ◆ SQL server log files and error dump files
- ◆ SQL Server trace flags
- ◆ OEM system ID
- ◆ Number of processors
- ◆ Processor Type
- ◆ Active processor mask
- ◆ % memory in use
- ◆ Total physical memory
- ◆ Free physical memory
- ◆ Total page file size
- ◆ Free page file size
- ◆ Total virtual memory
- ◆ Free virtual memory
- ◆ Disk info – Total & available space
- ◆ WINNT cluster name if shared disk cluster

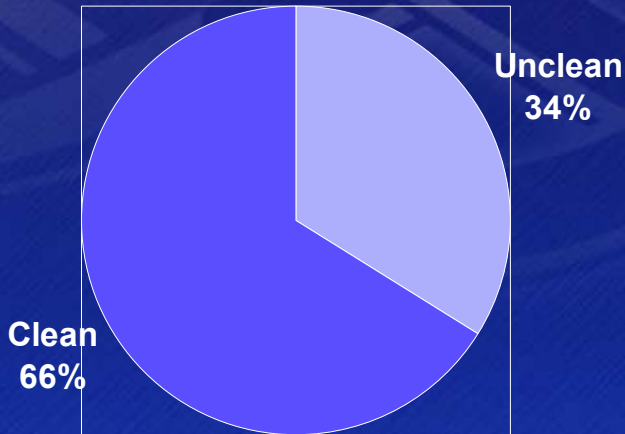
Snapshot: SQL-specific

- ◆ SQL Server trace flags
- ◆ Sysperfinfo table
- ◆ Sysprocesses table
- ◆ Syslocks table
- ◆ SQL Server response time
- ◆ SQL server specific counters
- ◆ \\SQLServer:Cache Manager(Adhoc Sql Plans)\\Cache Hit Ratio
- ◆ \\SQLServer:Cache Manager(Misc. Normalized Trees)\\Cache Hit Ratio"
- ◆ \\SQLServer:Cache Manager(Prepared Sql Plans)\\Cache Hit Ratio
- ◆ \\SQLServer:Cache Manager(Procedure Plans)\\Cache Hit Ratio
- ◆ \\SQLServer:Cache Manager(Replication Procedure Plans)\\Cache Hit Ratio
- ◆ \\SQLServer:Cache Manager(Trigger Plans)\\Cache Hit Ratio
- ◆ \\SQLServer:General Statistics\\User Connections

Snapshot: O/S-specific

- ◆ Application and system event logs
- ◆ Select OS counters
- ◆ \\Memory\\Available Bytes
- ◆ \\PhysicalDisk(_Total)\\% Disk Time
- ◆ \\PhysicalDisk(_Total)\\Avg. Disk sec/Read
- ◆ \\PhysicalDisk(_Total)\\Avg. Disk sec/Write
- ◆ \\PhysicalDisk(_Total)\\Current Disk Queue length
- ◆ \\PhysicalDisk(_Total)\\Disk Reads/sec
- ◆ \\PhysicalDisk(_Total)\\Disk Writes/sec
- ◆ \\Processor(_Total)\\% Processor Time
- ◆ \\Processor(_Total)\\Processor Queue length
- ◆ \\Server\\Server Sessions
- ◆ \\System\\File Read Operations/sec
- ◆ \\System\\File Write Operations/sec
- ◆ \\System\\Procesor Queue Length

DCA Results



- ◆ **34% Unclean shutdown:**
 - 5% windows upgrades
 - 5% SQL stopped unexpectedly (SCM 7031)
 - 1% SQL perf degradation
 - 8% startup problems

- ◆ **66% Clean shutdown:**
 - 16% SQL Server upgrades
 - 3% Windows upgrades
 - 10% single user (admin operations)
 - 30% reboots during shutdowns

- Events non-additive (some shutdowns accompanied by multiple events)
- Results from beta & non-beta (lower s/w stability but production admin practices)

Interpreting the results

- ◆ **66% administrative action:**
 - Higher than Gray '85 (42%) or '90 (15%)
 - Increase expected but these data include beta S/W
- ◆ **5% O/S upgrades in unclean shutdown category**
 - Note: 5% SQL not stopped properly
 - SCM doesn't shutdown SQL properly
 - O/S admin doesn't know to bring SQL Down properly
- ◆ **Perf degradation & deadlocks often yeild DB restart**
- ◆ **DB S/W failure not substantial cause of downtime in this sample**
- ◆ **S/W upgrades contribute many scheduled outages**
- ◆ **Single user mode contribution significantly**
- ◆ **System reboots a leading cause of outages**
 - O/S or DB S/W upgrade
 - Application, database, or system not behaving properly

Drill Down: Data from single Server

- ◆ Experiment in how much can be learned from a detailed look
 - Single randomly selected server
- ◆ Attempt to understand each O/S and SQL restart
- ◆ SQL closes connections on some failures, attempt to understand each of these as well as failures
- ◆ Overall findings:
 - All 159 symptom dumps generated by server mapped to known bugs
 - This particular server has a vendor supplied backup program that is not functioning correct and the admin team doesn't appear to know it yet
 - Large numbers of failures often followed by a restart:
 - events per unit time look like good predictor
 - Two way support tailoring data collected would help
 - Adaptive intelligence needed at the data collector

Detailed Drill Down Timeline

