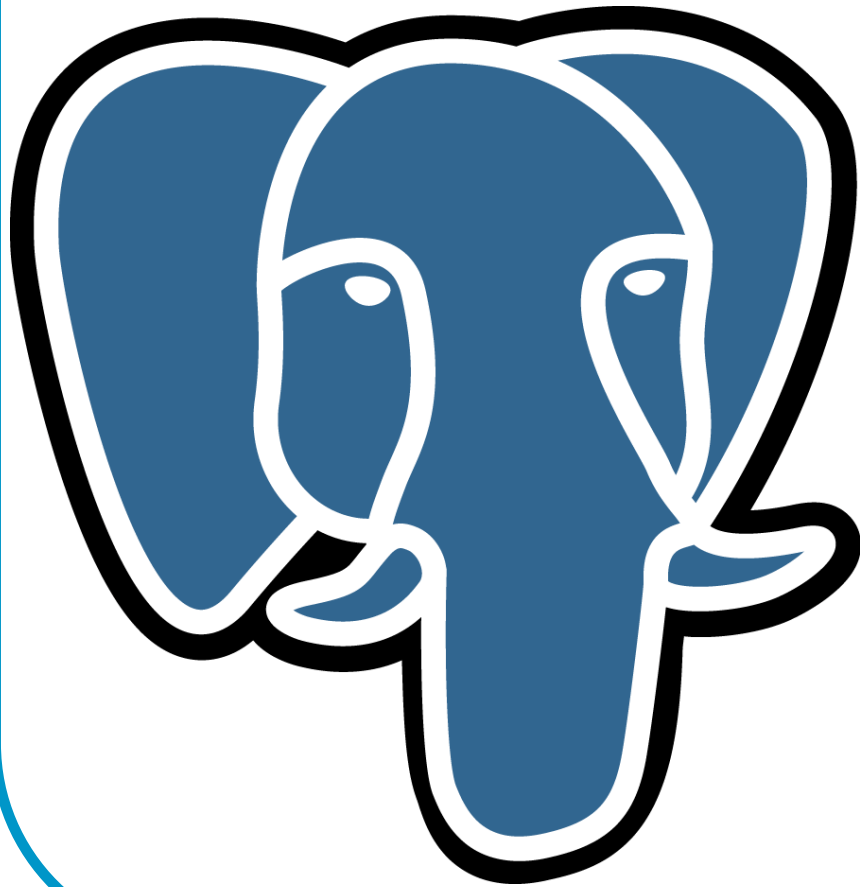


PostgreSQL on Windows



Magnus Hagander
magnus@hagander.net

PGCon, Ottawa
May 2007

Agenda

- Why PostgreSQL on Windows
- PostgreSQL for the Windows user
- Windows for the PostgreSQL user
- Advances in 8.3

Why PostgreSQL on Windows

- Isn't Linux better?
 - Often, but not always
- Several scenarios
 - Developer laptops
 - Desktop database
 - Migration scenarios
 - Corporate policies

Agenda

- Why PostgreSQL on Windows
- **PostgreSQL for the Windows user**
- Windows for the PostgreSQL user
- Advances in 8.3

PostgreSQL for the Windows user

- Unix inheritance
 - Multi-process, not multi-thread
 - Shared memory
 - Requires "modern" windows (2000+, no FAT)
- Commandline!
 - psql, pg_dump, pg_dumpall, etc
 - pgAdmin3 to the rescue!

PostgreSQL for the Windows user

- Environment variables
- Configuration files
 - No registry
 - Edit with any text editor (e.g. notepad)
 - Edit with pgAdmin3
 - Still just a textfile
 - Signal server to reload (pause service)
- Security
 - Will *not* run with administrative privileges

Agenda

- Why PostgreSQL on Windows
- PostgreSQL for the Windows user
- **Windows for the PostgreSQL user**
- Advances in 8.3

Windows for the PostgreSQL user

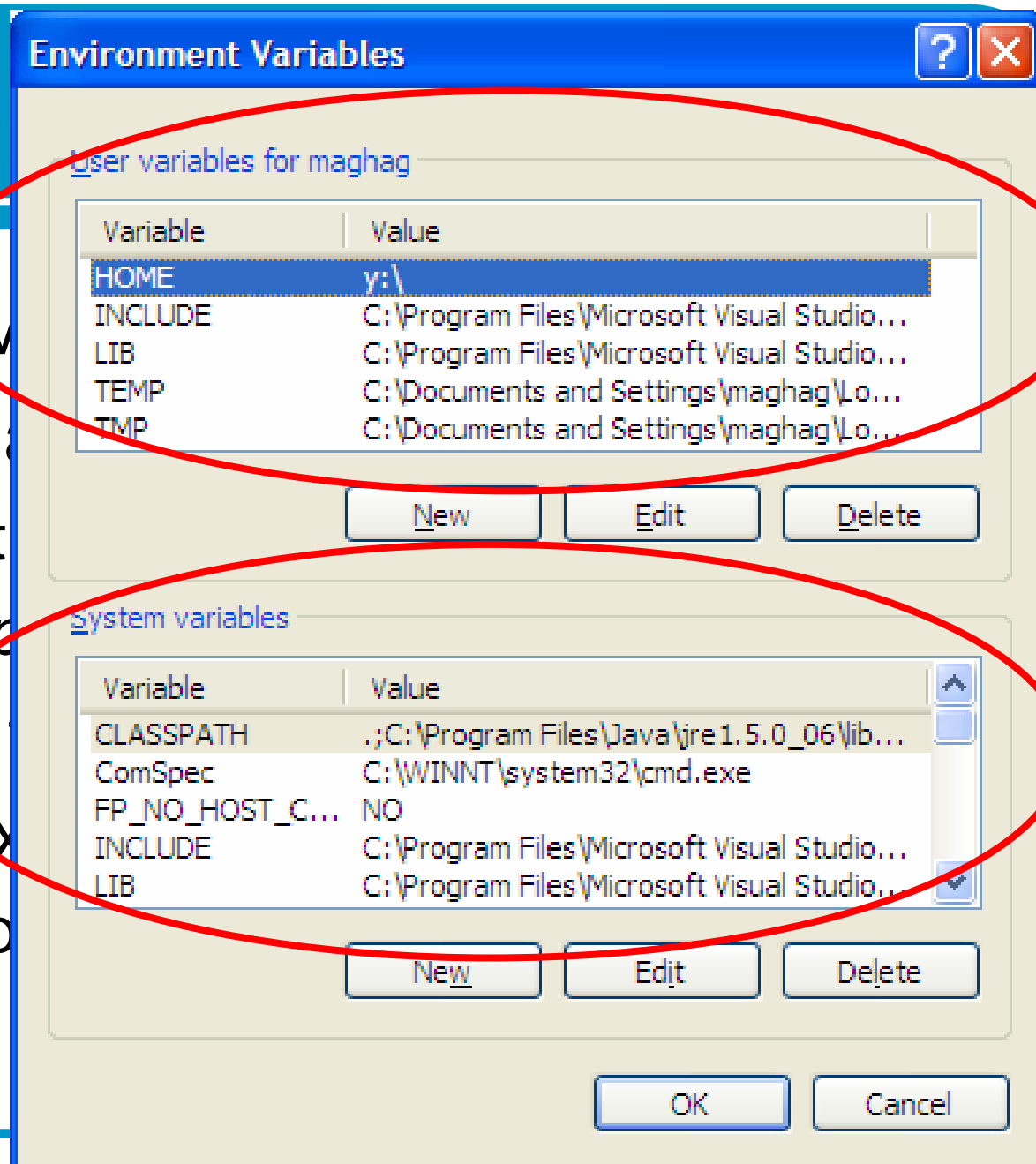
- It's a brave new world
- None of the normal tools
 - No ps
 - No kill
 - No top
 - No cron
 - Nothing at all (almost)
- Builtin Windows tools are generally bad

Architectural differences

- Runs as a service
 - Windows "version" of daemon
 - Started by "Service Control Manager"
 - Has it's own login and session
- Logs to a file or eventlog
 - Startup errors *always* to eventlog
- CreateProcess() instead of fork()
 - Extra important to use connection pooling

Installing

- Use the M...
- External l...
 - Perl, pyt...
 - Path is p...
- Or install...
 - Complex...
 - 8.3 supp...



Installing PostgreSQL - tips

- Turn off all unnecessary services
- Install data on dedicated filesystem
 - If possible, on dedicated spindles
- Use a junction or disk mount for xlog
- Mount with "noatime"
`fsutil behavior set disablelastaccess 1`
- Disable 8.3 filename generation
`fsutil behavior set disable8dot3 1`

Configuration parameters

- shared memory
 - Workload dependant
 - Smaller is better?!
- fsync methods
 - open_datasync (o_direct in 8.3)
 - fsync_writethrough
- log_destination
- lc_xyz

Managing PostgreSQL

- Get the right tools!

[http://www.microsoft.com/
technet/sysinternals](http://www.microsoft.com/technet/sysinternals)

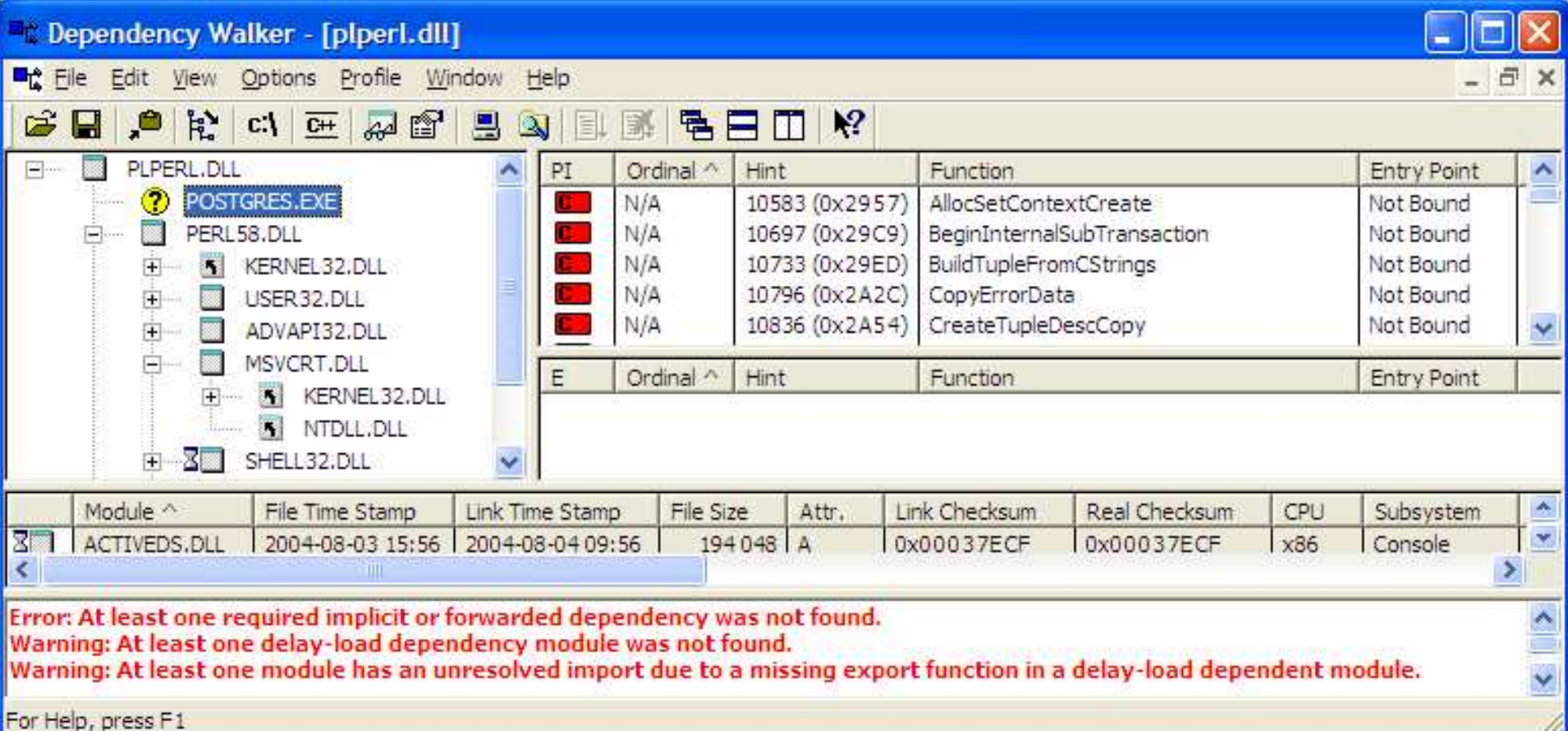
- Process Explorer (*the* tool for Win32)
- pstools
- Process monitor (debugging)
- Junction (tablespaces, xlog location)

Managing PostgreSQL

<http://www.microsoft.com/technet/sysinternals>

Library dependencies

- On Unix: "ldd"



Dependency Walker - [plperl.dll]

File Edit View Options Profile Window Help

PLPERL.DLL

- POSTGRES.EXE
- PERL58.DLL
 - KERNEL32.DLL
 - USER32.DLL
 - ADVAPI32.DLL
 - MSVCRT.DLL
 - KERNEL32.DLL
 - NTDLL.DLL
 - SHELL32.DLL

PI	Ordinal ^	Hint	Function	Entry Point
?	N/A	10583 (0x2957)	AllocSetContextCreate	Not Bound
?	N/A	10697 (0x29C9)	BeginInternalSubTransaction	Not Bound
?	N/A	10733 (0x29ED)	BuildTupleFromCStrings	Not Bound
?	N/A	10796 (0x2A2C)	CopyErrorData	Not Bound
?	N/A	10836 (0x2A54)	CreateTupleDescCopy	Not Bound

E	Ordinal ^	Hint	Function	Entry Point
---	-----------	------	----------	-------------

Module ^	File Time Stamp	Link Time Stamp	File Size	Attr.	Link Checksum	Real Checksum	CPU	Subsystem
ACTIVEDS.DLL	2004-08-03 15:56	2004-08-04 09:56	194 048	A	0x00037ECF	0x00037ECF	x86	Console

Error: At least one required implicit or forwarded dependency was not found.
Warning: At least one delay-load dependency module was not found.
Warning: At least one module has an unresolved import due to a missing export function in a delay-load dependent module.

For Help, press F1

Process title

- Unix:

```
root@svr1:~# ps axuwf |grep postgres
pgsql      1397  0.0  0.1  5360 1744 ?        S      2006   0:23  \_ postgres: logger process
pgsql      1399  0.0  0.3 15708 4160 ?        S      2006   2:29  \_ postgres: writer process
pgsql      1400  0.0  0.1   6360 1808 ?        S      2006   0:03  \_ postgres: stats buffer process
```


Process title

- Win32:

The screenshot shows the Process Explorer window from Sysinternals. The main pane displays a list of processes, with a red oval highlighting the PostgreSQL-related entries:

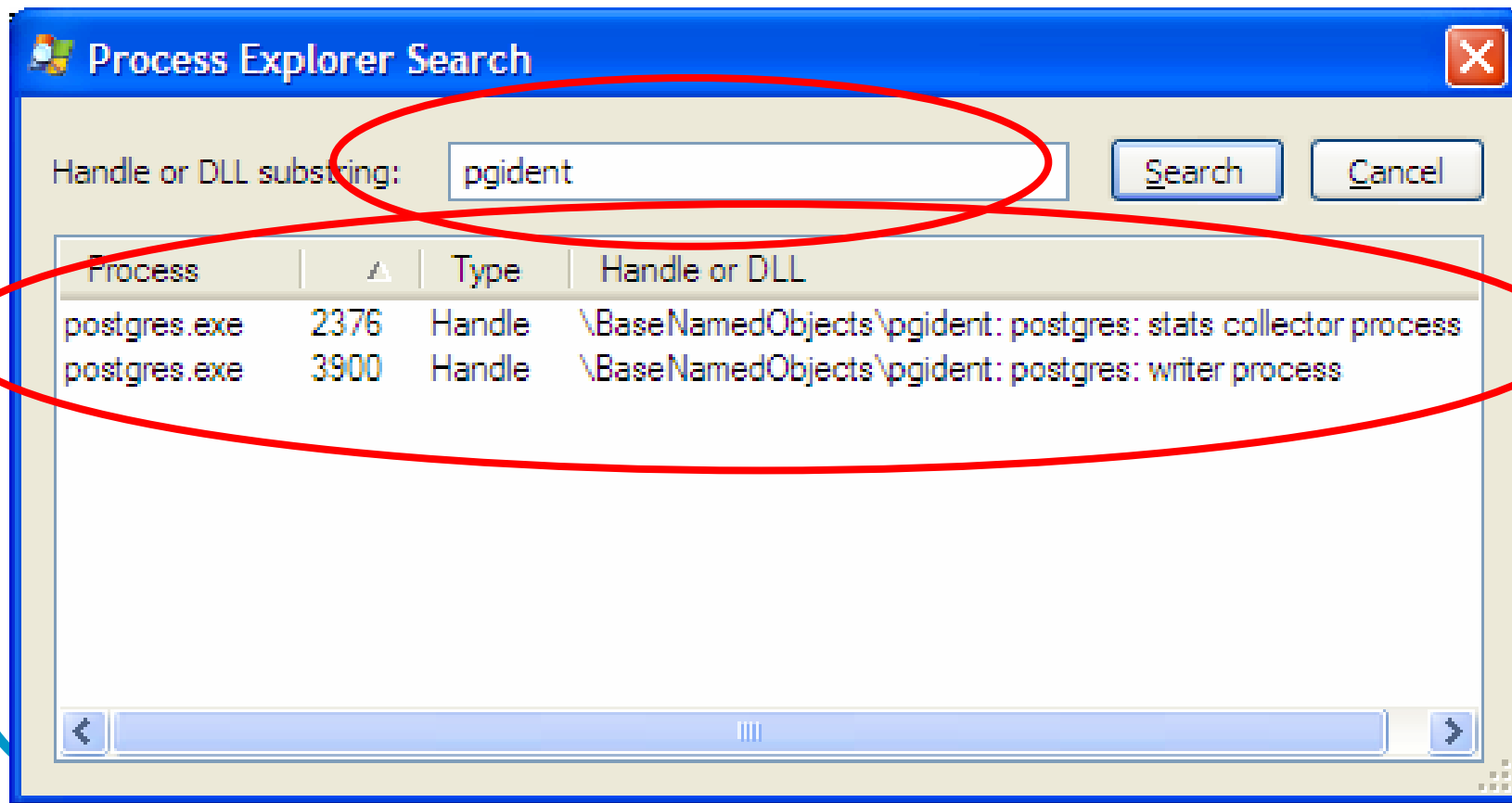
Process	PID	CPU	Description	Company Name
pg_ctl.exe	3648		pg_ctl - starts/stops/restarts t...	PostgreSQL Global Development Group
postgres.exe	4292		PostgreSQL Server	PostgreSQL Global Development Group
postgres.exe	3948		PostgreSQL Server	PostgreSQL Global Development Group
postgres.exe	260		PostgreSQL Server	PostgreSQL Global Development Group

The bottom pane shows the system tree with a red oval highlighting the registry path:

Type	Name
Directory	\KnownDlls
Event	\BaseNamedObjects\pgident: postgres: writer process
File	\Device\Udp
File	C:\prog\pgsql\move\data
File	C:\WINNT\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0...

Process title

- Win32:



Finding "stuck processes"

- Unix:

```
svr1.hagander.net - PuTTY
```

```
root@svr1:~# strace -p 12771  
Process 12771 attached - interrupt to quit  
recv(7, █
```

Finding "stuck processes"

- Win32:

The screenshot shows the 'postgres.exe:4404 Properties' dialog box with the 'Threads' tab selected. The thread list shows three threads: 'postgres.exe!mainCRTStartup', 'postgres.exe!pg_signal_thread', and 'postgres.exe!pg_timer_thread'. The 'Stack for thread 5164' dialog is open, displaying a stack trace for thread 5164. The stack trace is as follows:

Index	Function Name
0	ntkmlpa.exe!KiUnexpectedInterrupt+0xf0
1	ntkmlpa.exe!NtWaitForSingleObject+0x38c
2	ntkmlpa.exe!KeReleaseInStackQueuedSpinLockFromDpcLevel+0xb
3	ntdll.dll!KiFastSystemCallRet
4	postgres.exe!pgwin32_waitforsinglesocket+0x1cf
5	postgres.exe!pgwin32_recv+0x83
6	postgres.exe!secure_read+0x20d
7	postgres.exe!pq_recvbuf+0x97
8	postgres.exe!pq_getbyte+0x16
9	postgres.exe!SocketBackend+0xb

The 'Stack' button at the bottom right of the 'Stack for thread 5164' dialog is circled in red. The 'Stack' button in the main 'postgres.exe:4404 Properties' dialog is also circled in red.

Monitoring PostgreSQL

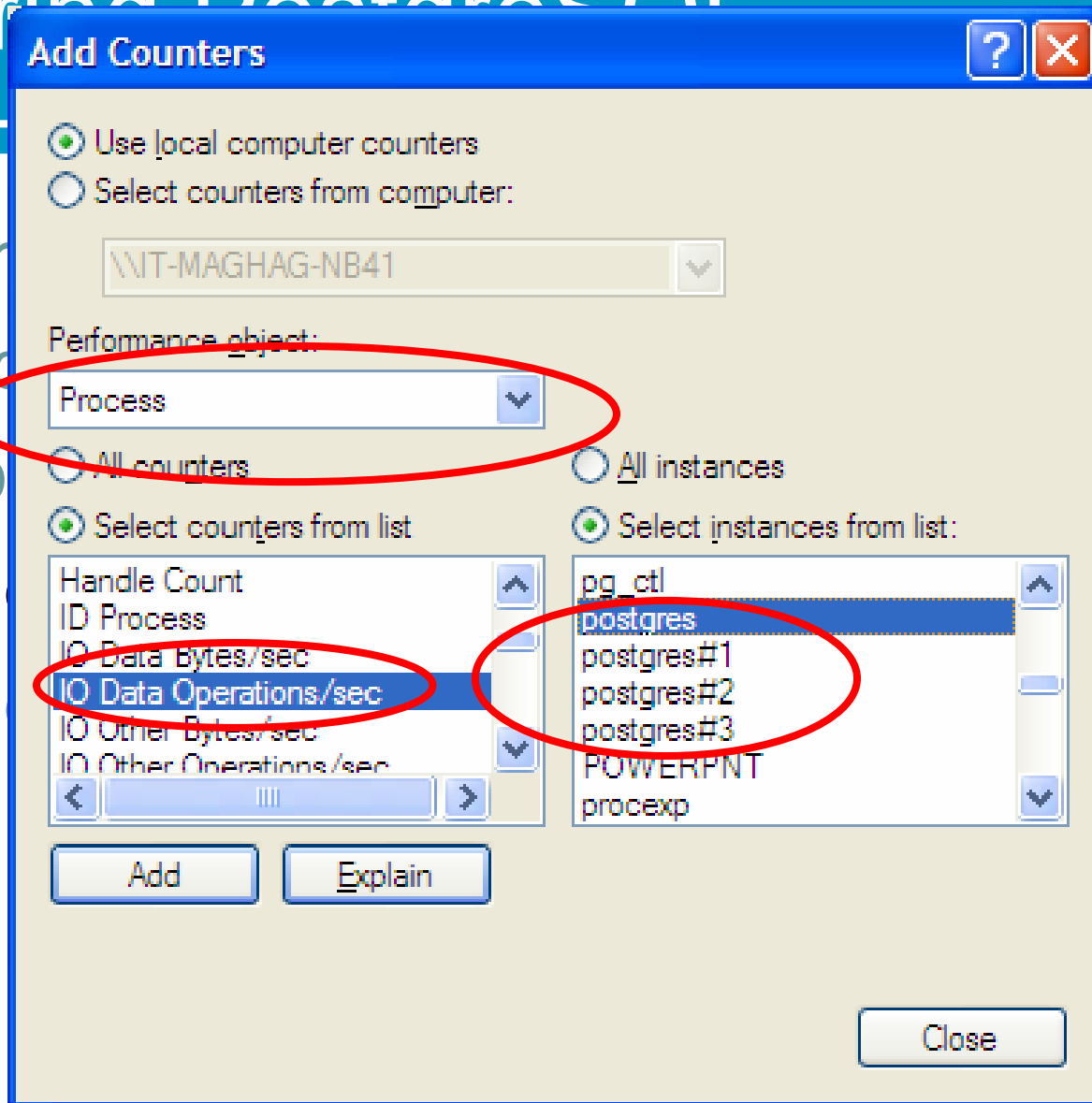
- Some things really didn't change
- pg_stat_xyz
 - *Has not changed at all*
- PostgreSQL logs
 - pg_data*.log
 - Don't forget eventlog!

Monitoring PostgreSQL

- Performance Monitor
 - Standard Windows monitoring still applies!
- Monitoring the whole server
- Looking at individual backends
- Looking at the whole cluster

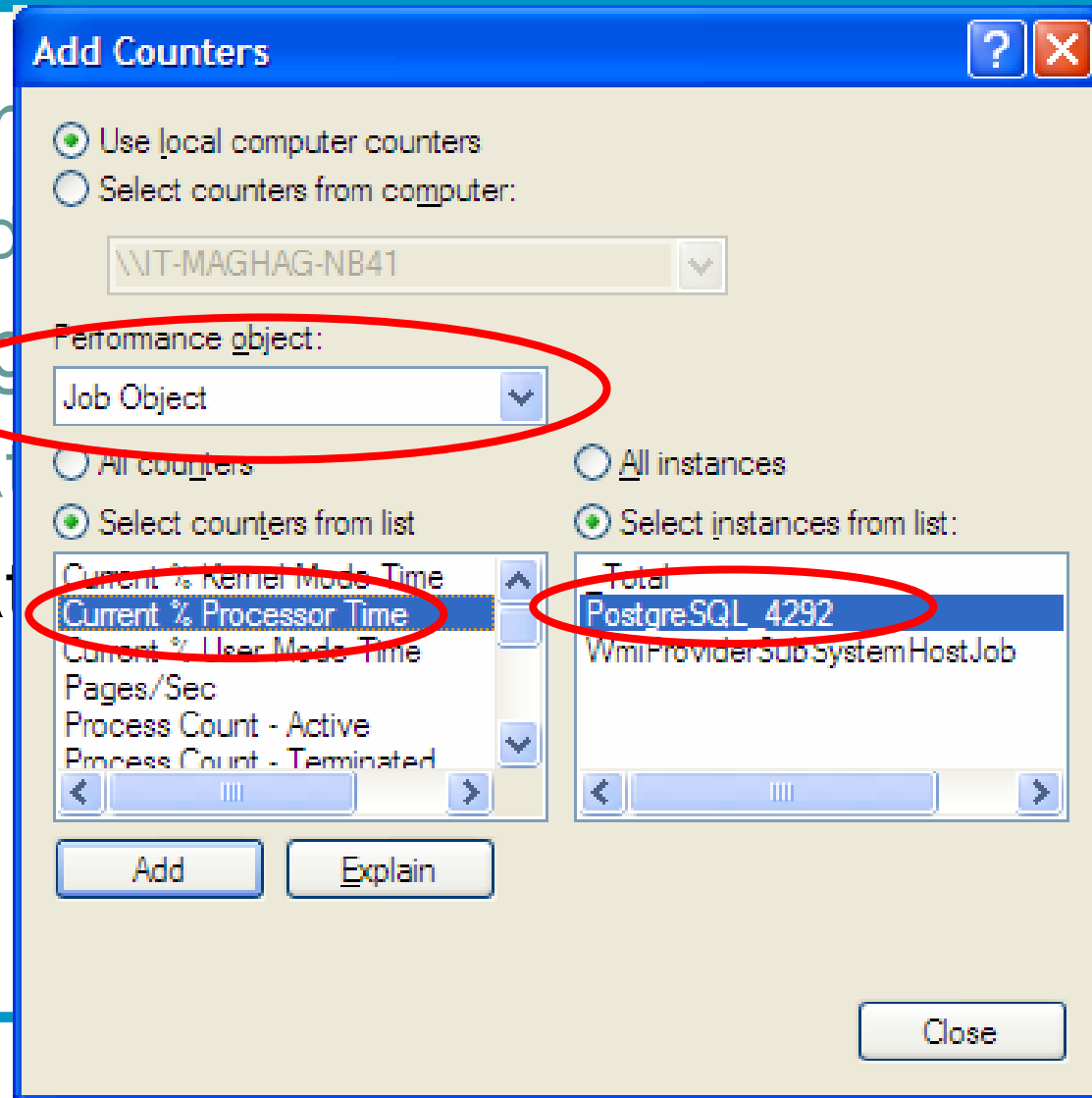
Monitoring PostgreSQL

- Performance
- Standby
- Monitoring
- Looking at
- Looking at



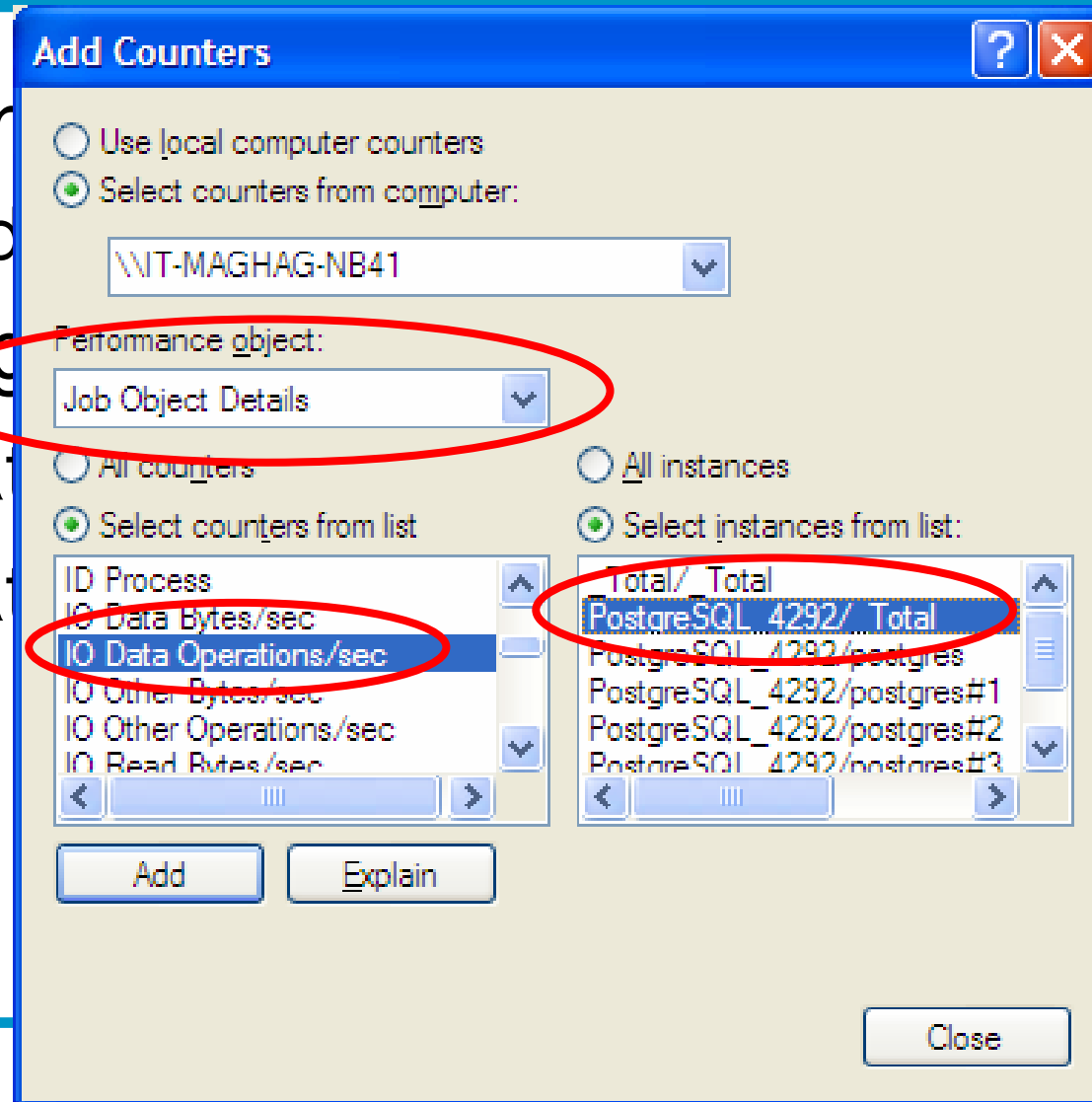
Monitoring PostgreSQL

- Performance
 - Standard
- Monitoring
- Looking at
- Looking at



Monitoring PostgreSQL

- Performance
 - Standard
- Monitoring
- Looking at
- Looking at



Interesting counters to watch

- Number of processes
- New process creations
- Memory usage (private bytes, working set)
- Context switches / second
- Processor queue length

Interesting counters to watch

- I/O operations / second
- I/O bytes / second
- Physical disk queue length
- Logical disk\avg sec / read or write
 - xlog: <10ms
 - data: <50ms
- Logical disk\Disk transfers / sec

Monitoring with Process Explorer

- Add extra columns
 - Private bytes
 - Local backend memory
 - WS Shareable / WS Shared
 - Shared buffers + shared code
 - Virtual Size
 - Just address space
 - Context switch delta
 - Expensive on Windows!

Agenda

- Why PostgreSQL on Windows
- PostgreSQL for the Windows user
- Windows for the PostgreSQL user
- **Advances in 8.3**

Advances in 8.3

- Main build is now built with Visual C++
 - More efficient binaries
 - Works with Windows Debugger Tools
 - Works with Visual Studio debugger and profiler
 - Support for detached symbols

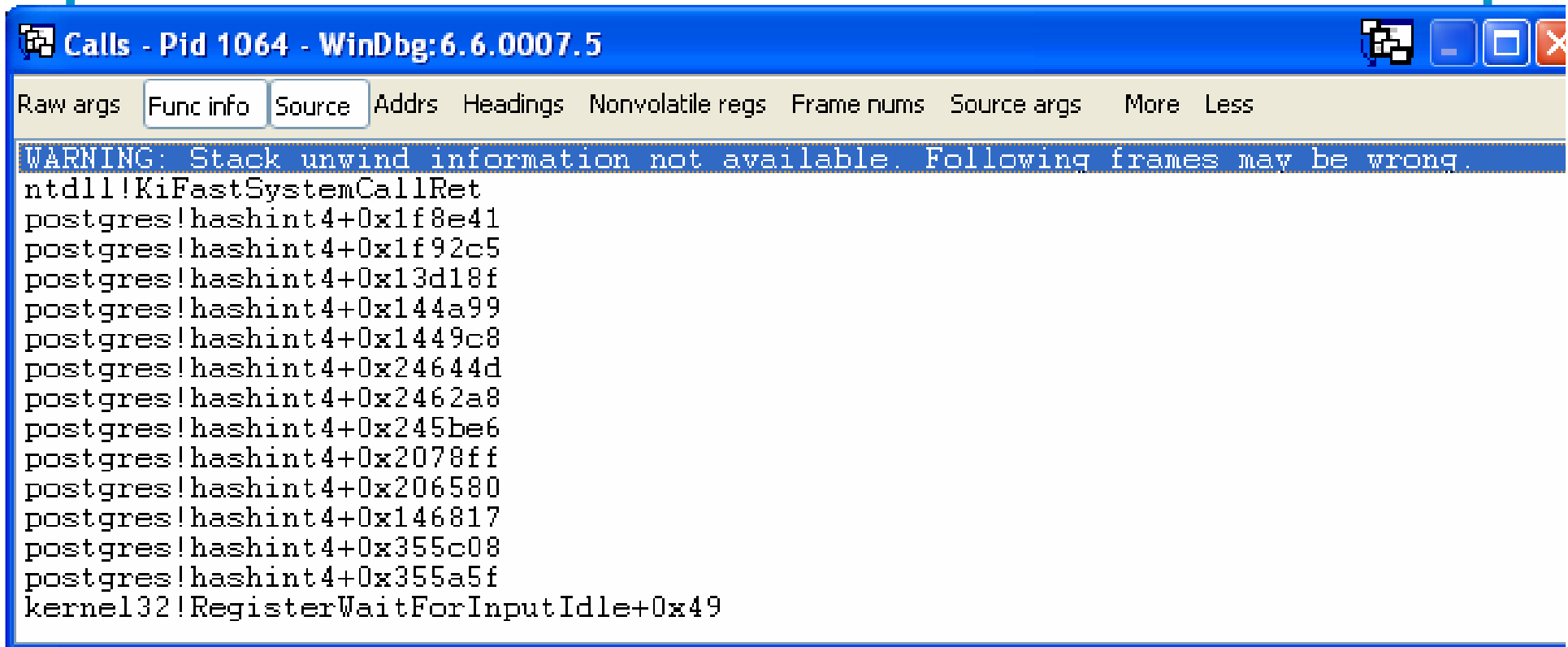
Windows debugger support in 8.3

- Debugger backtrace of mingw build

```
Calls - Pid 776 - WinDbg:6.6.0007.5
Raw args  Func info  Source  Addr  Headings  Nonvolatile regs  Frame nums  Source args  More  Less
WARNING: Stack unwind information not available. Following frames may be wrong.
ntdll!KiFastSystemCallRet
postgres!pgwin32_waitforsinglesocket+0x217
postgres!pgwin32_recv+0x8b
postgres!secure_read+0x2d
postgres!TouchSocketFile+0x93
postgres!pq_getbyte+0x22
postgres!PostgresMain+0x1058
postgres!SubPostmasterMain+0x804
postgres!main+0x568
postgres+0x11e7
postgres+0x1238
kernel32!RegisterWaitForInputIdle+0x49
```

Windows debugger support in 8.3

- Debugger backtrace of msvc build



The screenshot shows a debugger window titled "Calls - Pid 1064 - WinDbg:6.6.0007.5". The window has a menu bar with "Raw args", "Func info", "Source", "Addr", "Headings", "Nonvolatile regs", "Frame nums", "Source args", "More", and "Less". Below the menu bar, a warning message reads: "WARNING: Stack unwind information not available. Following frames may be wrong." The call stack is displayed as follows:

```
ntdll!KiFastSystemCallRet
postgres!hashint4+0x1f8e41
postgres!hashint4+0x1f92c5
postgres!hashint4+0x13d18f
postgres!hashint4+0x144a99
postgres!hashint4+0x1449c8
postgres!hashint4+0x24644d
postgres!hashint4+0x2462a8
postgres!hashint4+0x245be6
postgres!hashint4+0x2078ff
postgres!hashint4+0x206580
postgres!hashint4+0x146817
postgres!hashint4+0x355c08
postgres!hashint4+0x355a5f
kernel32!RegisterWaitForInputIdle+0x49
```

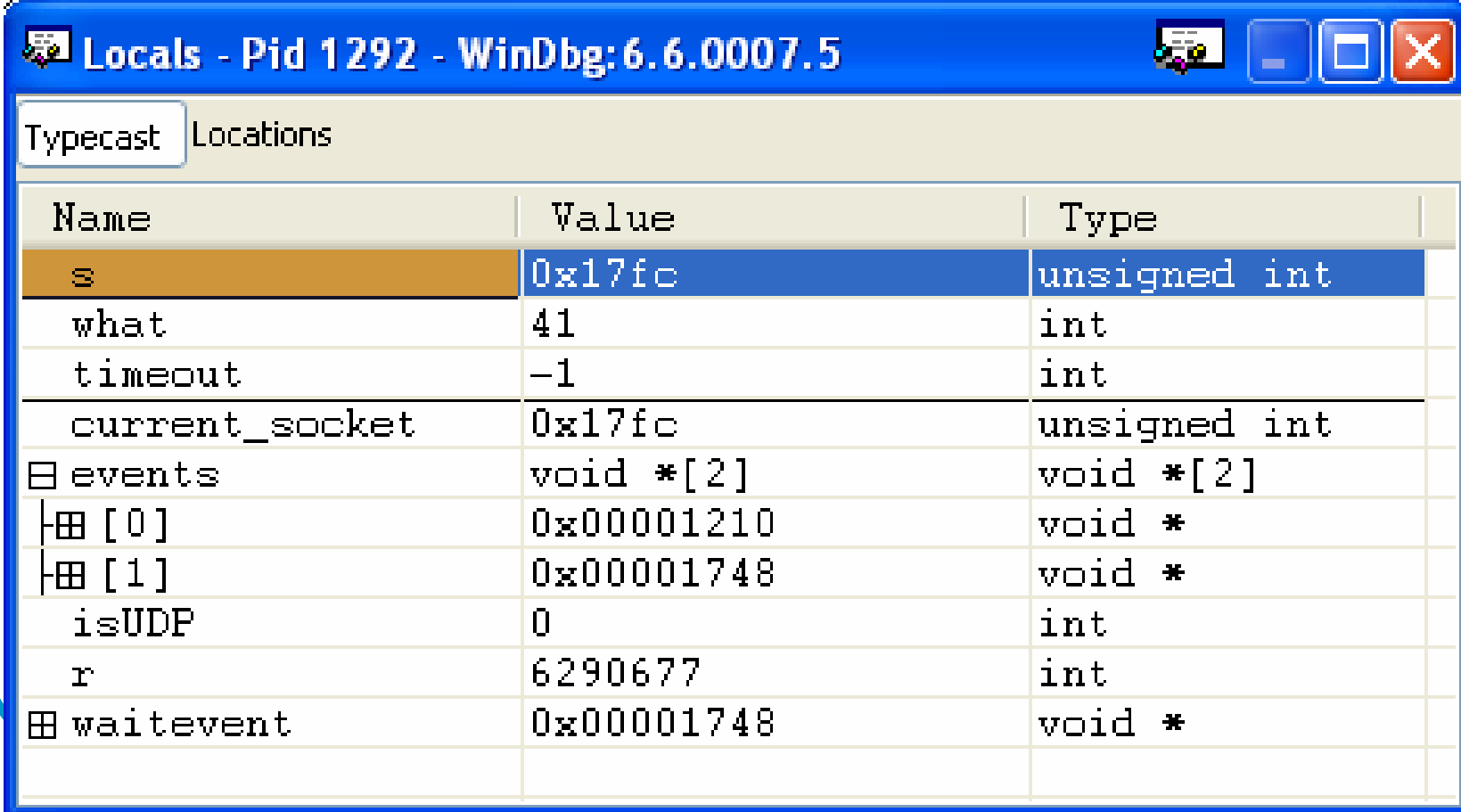

Windows debugger support in 8.3

- Debugger backtrace of msvc build

```
Calls - Pid 1292 - WinDbg:6.6.0007.5
Raw args  Func info  Source  Addr  Headings  Nonvolatile regs  Frame nums  Source args  More  Less
-----
[WARNING: Stack unwind information not available. Following frames may be wrong.]
ntdll!KiFastSystemCallRet
postgres!pgwin32_waitforsinglesocket+0x1cf [d:\prog\pgsql\pgsql\src\backend\port\win32\socket.c @ 198]
postgres!pgwin32_recv+0x83 [d:\prog\pgsql\pgsql\src\backend\port\win32\socket.c @ 316]
postgres!secure_read+0x20d [d:\prog\pgsql\pgsql\src\backend\libpq\be-secure.c @ 326]
postgres!pq_recvbuf+0x97 [d:\prog\pgsql\pgsql\src\backend\libpq\pqcomm.c @ 721]
postgres!pq_getbyte+0x16 [d:\prog\pgsql\pgsql\src\backend\libpq\pqcomm.c @ 761]
postgres!SocketBackend+0xb [d:\prog\pgsql\pgsql\src\backend\tcop\postgres.c @ 299]
postgres!ReadCommand+0x16 [d:\prog\pgsql\pgsql\src\backend\tcop\postgres.c @ 421]
postgres!PostgresMain+0xf64 [d:\prog\pgsql\pgsql\src\backend\tcop\postgres.c @ 3452]
postgres!BackendRun+0x2bd [d:\prog\pgsql\pgsql\src\backend\postmaster\postmaster.c @ 2987]
postgres!SubPostmasterMain+0x1fe [d:\prog\pgsql\pgsql\src\backend\postmaster\postmaster.c @ 3438]
postgres!main+0x1c5 [d:\prog\pgsql\pgsql\src\backend\main\main.c @ 165]
postgres!__tmainCRTStartup+0x1a6 [f:\sp\vctools\crt_bld\self_x86\crt\src\crtexe.c @ 597]
postgres!mainCRTStartup+0xd [f:\sp\vctools\crt_bld\self_x86\crt\src\crtexe.c @ 414]
kernel32!RegisterWaitForInputIdle+0x49
```

Windows debugger support in 8.3

- Local symbols supported!

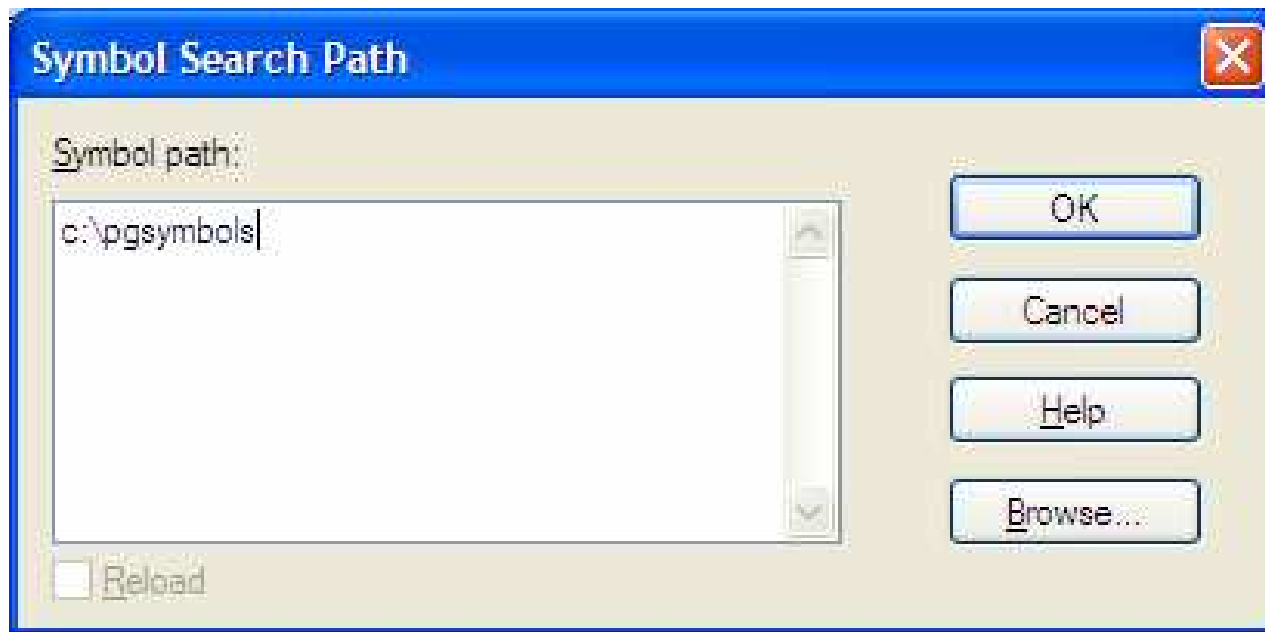


The screenshot shows the 'Locals' window in WinDbg, titled 'Locals - Pid 1292 - WinDbg:6.6.0007.5'. The window has two tabs: 'Typecast' and 'Locations'. The 'Locations' tab is active, displaying a table of local variables. The table has three columns: 'Name', 'Value', and 'Type'. The variable 's' is highlighted in orange. Other variables include 'what', 'timeout', 'current_socket', 'events' (an array of two void pointers), 'isUDP', 'r', and 'waitevent'.

Name	Value	Type
s	0x17fc	unsigned int
what	41	int
timeout	-1	int
current_socket	0x17fc	unsigned int
events	void *[2]	void *[2]
[0]	0x00001210	void *
[1]	0x00001748	void *
isUDP	0	int
r	6290677	int
waitevent	0x00001748	void *

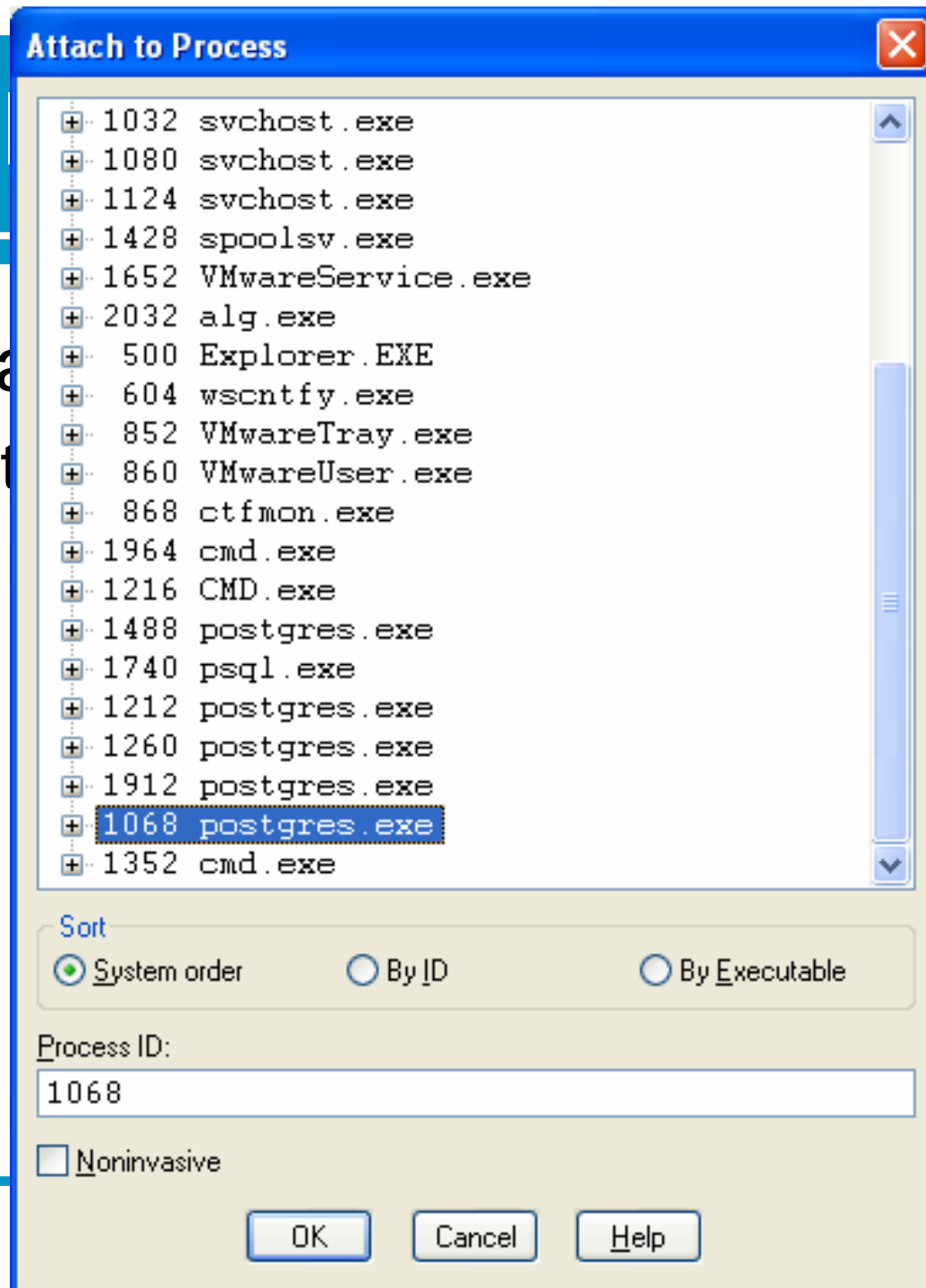
Using windbg

- Step 0 – configure symbol directory

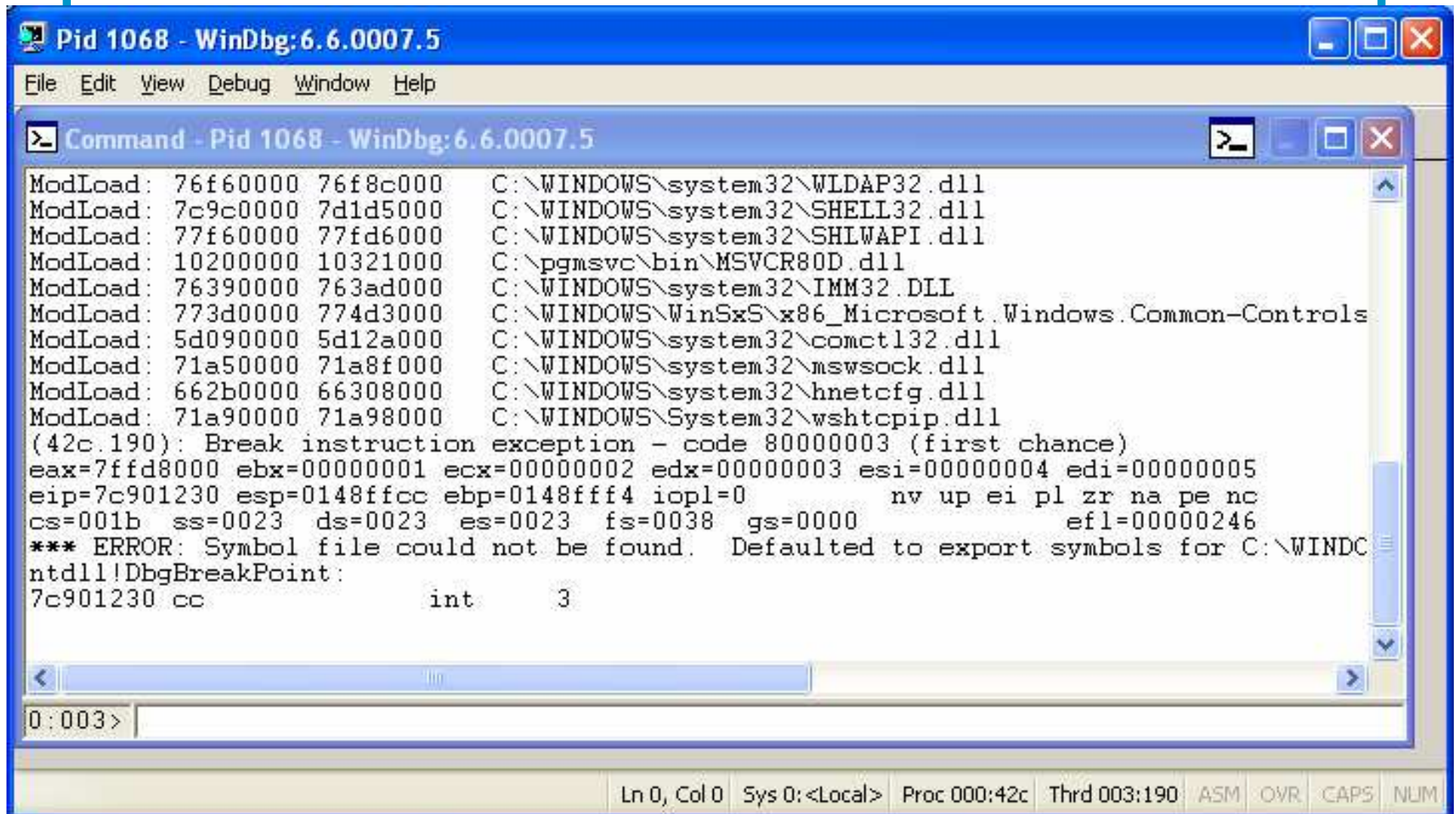


Using wind

- Step 1 – attach
- Figured out



Using windbg



```
Pid 1068 - WinDbg:6.6.0007.5
File Edit View Debug Window Help

Command - Pid 1068 - WinDbg:6.6.0007.5
ModLoad: 76f60000 76f8c000 C:\WINDOWS\system32\WLDAP32.dll
ModLoad: 7c9c0000 7d1d5000 C:\WINDOWS\system32\SHELL32.dll
ModLoad: 77f60000 77fd6000 C:\WINDOWS\system32\SHLWAPI.dll
ModLoad: 10200000 10321000 C:\pgmsvc\bin\MSVCR80D.dll
ModLoad: 76390000 763ad000 C:\WINDOWS\system32\IMM32.DLL
ModLoad: 773d0000 774d3000 C:\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls
ModLoad: 5d090000 5d12a000 C:\WINDOWS\system32\comctl32.dll
ModLoad: 71a50000 71a8f000 C:\WINDOWS\system32\mswsock.dll
ModLoad: 662b0000 66308000 C:\WINDOWS\system32\hnetcfg.dll
ModLoad: 71a90000 71a98000 C:\WINDOWS\System32\wshtcpip.dll
(42c.190): Break instruction exception - code 80000003 (first chance)
eax=7ffd8000 ebx=00000001 ecx=00000002 edx=00000003 esi=00000004 edi=00000005
eip=7c901230 esp=0148ffcc ebp=0148fff4 iopl=0         nv up ei pl zr na pe nc
cs=001b  ss=0023  ds=0023  es=0023  fs=0038  gs=0000             efl=00000246
*** ERROR: Symbol file could not be found.  Defaulted to export symbols for C:\WINDC
ntdll!DbgBreakPoint:
7c901230 cc                int     3

0:003>
```

Ln 0, Col 0 Sys 0: <Local> Proc 000:42c Thrd 003:190 ASM OVR CAPS NUM

Using windbg

- Step 3 – set a breakpoint

```
*** ERROR: Symbol file could not be found for
ntdll!DbgBreakPoint:
7c901230 cc int 3
0:004> bp int8in
*** WARNING: Unable to verify checksum
```

Using windbg

- Step 4 – run and hit breakpoint

```
> Command - Pid 1068 - WinDbg:6.6.0007.5
```

```
0:0003> g
```

- Step 5 – load whatever views are needed

Visual Studio debugger

- Much better actual debugger
- Work off symbols or source tree
- Suitable for development, not production

Thank you!

Questions?