Tech Note 805 Validating Custom Developed Client Controls for Use in InTouch®

All Tech Notes, Tech Alerts and KBCD documents and software are provided "as is" without warranty of any kind. See the Terms of Use for more information.

Topic#: 002591 Created: November 2011

Introduction

With a managed InTouch application, you can develop your own Client Control objects, import them through IDE and embed them in InTouch windows via ArchestrA Symbols. This gives you the ability to add highly-customized functionality to InTouch and Application Server applications.

In the meantime, it may also silently introduce unwanted harm to the InTouch application which may go unnoticed until Runtime. For example

- The Client Control can cause View to leak memory and handle, eventually causing View to crash.
- The Client Control can cause View to leak GDI Handle, eventually causing View window to fracture, or to crash with an assertion error.

This *Tech Note* provides one easy and handy way to bulletproof a custom-developed Client Control to keep the InTouch application safe from potential harm, thus saving time for additional troubleshooting or even downtime later in production.

Application Versions

- Application Server 3.1 and later
- InTouch 10.1 and later

Validation Procedure

- 1. Import the custom developed Client Control(s) into the ArchestrA IDE.
- 2. Embed the custom Client Control objects into an ArchestrA symbol using the ArchestrA Graphic Editor. Configure the Client Control objects just as you would normally apply in regular application.
- 3. Create a test InTouch application, embed the ArchestrA symbol (containing the custom Client Control objects) in InTouch windows, and configure the ArchestrA symbol just as you would normally do with your regular InTouch application.
- 4. Add scripts and make other configurations to InTouch application as of necessary. For example, I created scripts that automatically open and close the windows in a fixed interval.
- 5. Start WindowViewer™; or create and deploy an InTouchViewApp object instance from the template, then run WindowViewer from the deployed InTouchView node.
- 6. Simulate regular runtime environment, such as opening/closing windows, interacting with the Client Control objects etc.

Tools and Validation Method

Performance Monitor (Perfmon)

This is a commonly available utility on all OS, and can be started by running Perfmon command line in DOS Window, or directly from Start/Run.

• Add Private Bytes and Handle Count counters of the Process object from InTouch View instance.

WindowViewer has to be running in order to have View showing in the instance list. The utility's UI looks slightly different on Windows XP, Windows 2003 to Windows 2008, Windows 7. The following graphics show Windows 2008 as the example.

Validating Custom Developed Client Controls for Use in InTouch®

		Counter	Parent	Inst	Computer	
<local computer=""></local>	Browse	Process	M 177 8		and additional	E
Print Queue	± 🔺	Private Bytes		view		
Process	Ξ	Indifue count	2000	VIEW		
% Privileged Time						
% Processor Time	-					
% User Time						
Creating Process ID	123					
Elapsed Time						
Handle Count	_					
and the second second second						
istances of selected object:						
svchost#9	<u> </u>					
taskeng						
taskeng#1						
taskeng#2						
view						
	Search					

FIGURE 1: ADDING COUNTERS IN PERFORMANCE MONITOR

• Choose the proper values for the Sample Interval, Graph Duration, and Vertical Scale properties etc. to fit your testing needs (Figures 2 and 2a below).

Validating Custom Developed Client Controls for Use in InTouch®

rformance Monitor	Properties	2	Performance Monitor Properties	
ieneral Source Dat	ta Graph Appeara	nce	General Source Data Graph Appearance	
Display elements	-	-	C Stroll @ Wrap	
I✓ Legend	I✓ Value bar	I✓ <u>T</u> oolbar		
Report and histogra	am data			
• Default	C Minimum	C Average	<u>T</u> ite:	
C Current	C Maximum		Vertical axis:	
	Sample every	30 seconds 3000 seconds	☐ Horizontal grid	
			Vertical scale	
			Maximum: 3000	
			Minimum: 400	
	OK	Cancel Apply	OK Cancel Apr	ply
IRE 2. CHART DISE	PLAY PROPERTIES >	GENERAL	FIGURE 24' CHART DISPLAY PROPERTIES > GRAPH	

• You can generate a Counter Log (file) for later analysis, or for when the testing may take a long time to complete.

GDIView

This is a unique tool that displays the list of GDI handles (to brushes, pens, fonts, bitmaps and others) opened by every process. It displays the total count for each type of GDI handle, as well as detailed information about each handle. This tool is very useful for detecting and tracing GDI resources leak in a process runtime, such as Client Control object embedded in InTouch View (process) through ArchestrA graphics.

Two different versions are available for 32-bit and 64-bit OS.

Download this free utility

Process Explorer

- This is another very useful handy tool for debugging runtime resource access conditions of a process.
- Process Explorer is bundled in the Microsoft Windows Sysinternals package.
- Download this free utility

Results and Analysis

• With Performance Monitor, Handle Count and Memory leaking conditions were detected while operating InTouch windows that included the custom Client Control embedded in the InTouch window via ArchestrA graphic.



FIGURE 3: 'PRIVATE BYTES' AND 'HANDLE COUNT' COUNTERS VERSUS TIME. THE BLUE LINE IS FOR VIEW'S 'PRIVATE BYTES' COUNTER VERSUS TIME

• With GDIView, GDI Handles leaking condition was detected while operating InTouch windows with custom Client Controls (embedded in InTouch windows via ArchestrA graphics) (Figure 4 below).

GDIView											<u> </u>
Eile Edit View Q	ptions Help										
Process Name	GDI Total	All GDI	Pen	ExtPen	Brush	Bitmap	Font	Palette	Region	DC	
svchost.exe	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	-
svchost.exe	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	
svchost.exe	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	
svchost.exe	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	
svchost.exe	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	
svchost.exe	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	
taskeng.exe	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	
taskeng.exe	14 [0]	120 [+36]	0 [0]	0 [0]	2 [0]	5 [0]	2 [0]	0 [0]	0 [0]	5 [0]	
UdaterUI.exe	26 [0]	35 [0]	0 [0]	0 [0]	11 [0]	4 [0]	9 [0]	0 [0]	0 [0]	2 [0]	
view.exe	631 [+6172	9997 [+9797] 5 [+1]	0 [0]	8 [0]	3037 [+29	. 219 [+211]	2 [0]	6 [0]	3040 [+2	9
VsTskMgr.exe	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	
wininit.exe	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	
winlogon.exe	4 [0]	4 [0]	0 [0]	0 [0]	1 [0]	1 [0]	0 [0]	0 [0]	0 [0]	2 [0]	
wm.exe	467 [0]	563 [0]	15 [0]	0 [0]	16 [0]	296 [0]	62 [0]	2 [0]	8 [0]	68 [0]	
wmiprvse.exe	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	
wmiprvse.exe	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	
procexp.exe	4 [0]	4 [0]	0 [0]	0 [0]	1 [0]	1 [0]	0 [0]	0 [0]	0 [0]	2 [0]	
procexp64.exe	111 [+107]	587 [+583]	1 [+1]	0 [0]	29 [+28]	26 [+25]	26 [+26]	0 [0]	5 [+5]	24 [+22]	
mspaint.exe	81 [+36]	161 [+82]	1 [+1]	0 [0]	28 [+14]	30 [+8]	3 [+2]	1 [+1]	3 [+2]	15 [+8]	-
4	100 C									1	•
Handle /	Object Type	Kerne	Address	Extended In	formation	Detec	t Counter	Detec	ted On		
@ 0x0101171d	DC	0xffff	f900c2d92710	and the second second	- Aline College	1	200 2 2 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2	10/28	/2011 10:20:53	AM	- torac
@ 0x0101171e	DC	0xffff	f900c2f2b710			1		10/28	/2011 10:20:53	AM	
@ 0v01011727	DC	Ovffff	f900c2d09710			1		10/28	/2011 10-20-53	AM	16

FIGURE 4. VIEW'S GDI HANDLE COUNT KEEPS INCREASING WHILE VIEW IS RUNNING. THE CIRCLED NUMBERS SHOW A NET INCREASE

• With Process Explorer, Handle Count, Memory and GDI Handle leaking conditions were detected (Figure 5 below).

Process	1.1	PID	CPU Description	Company Nan	ne	Private Bytes	GDI Objects
Isas:	s.exe	628	Local Security	Authority Proc Microsoft Corp	oration	8,192 K	0
Ism.e	exe	636	Local Session	Manager Serv Microsoft Corp	oration	3.660 K	0
(winlogo	on exe	576	Windows Log	on Application Microsoft Corp	oration	2.056 K	4
Bconime	exe	2744	Console IME	Microsoft Corp	oration	1,240 K	8
El a emiorer	C (7X)	4144	Windows Expl	lorer Microsoft Corp	oration	45 576 K	844
FILOOUT	TLOOK EXE	4756	Microsoft Out	ook Microsoft Corp	oration	65 568 K	780
	EXCELEXE	1996	Memorit Exer	Mcrosoft Corp	oration	14,720 K	193
CT 1 and	DEexe	3416	ArchestrA IDE			95 896 K	124
		4884	1 16 In Touch Wind	InuMaker Inueneus Suste	ama loc	49 324 K	563
	Status area	4412	0.58 In Touch Wind	touvMenuer Inventor Suste	ame loc	60 340 K	497
		Since I				00,540 K	
(STRICE)	Manu and	4240				2 220 4	74
and GDI	WRW RAD	4040	TCP/IP Security	Environment St	rings	2,000 K	1
El Le proc	exp exe	4040	Image Performance	Performance Graph Th	reads to som	3,420 N	500
	procexp-64,exe	4300	a contract of the desired and		AS.COM	27.572 K	036
Lisched	Jexe	4202	CPU	I/O		1,348 K	
(1) Tunest	Helper.exe	4292	Priority 8	I/O Priority	Normal	7,492 K	11
E Udatert	ULexe	4296	Kamel Time 0.00.02 313	Deads	3 901	3,072 K	35
McT	Tray.exe	4408	Nernel Time 0100103.213	Negus.	3,30 x	7,528 K	97
2 alamma	gr.exe	3908	User Time 0:00:03.900	Read Delta	38	11.972 K	8
(Ger mmc.ex	oe .	4140	Total Time 0:00:07.113	Read Bytes Delta 11	L3 KB	22.876 K	379
			Cycles 14,633,470,926	Writes	840		
Туре	Name *	2		Write Delta	0	Handle	Access
Thread	view.exe(4412): 4932		virtuai Memory	WITC CETU		0×930	0x001FFFFF
Thread	view.exe(4412): 4892		Private Bytes 60,340 K	Write Bytes Delta	0	0x9F0	0x001FFFFF
Thread	view.exe(4412): 4892		Peak Private Bytes 61,884 K	Other 1	18,465	0xA00	0x001FFFFF
Thread	view.exe(4412): 4800		Virtual Size 300,528 K	Other Delta	98	0x9C4	0x001FFFFF
Thread	view.exe(4412): 4748		Page Eaulter 35 735	Other Bytes Delta	8548	0x238	0x001FFFFF
Thread	view.exe(4412): 4748		Page radia 33,733	outer bytes beta	0010	0x744	0x001FFFFF
Thread	view.exe(4412): 4748		Page Fault Delta 11	Handles		0xA7C	0x001FFFFF
Thread	view.exe(4412): 4536		Dhusical Mamoru	Handles	840	Ox9F8	Ox001FFFFF
Thread	view.exe(4412): 4372		Physical Method y	CDI Handler	477	OKDOC	0x001FFFFF
Thread	view.exe(4412): 4248		Memory Priority 5	GULHARUES	747	OKIED	QX001FFFFF
Thread	view.exe(4412): 3968		Working Set 81,652 K	USER Handles	237	0x440	OKOO1FFFFF
Thread	view.exe(4412): 3880		WS Private 42,540 K			0,604	UXUU IFFFFF
Inread	view.exe(4412): 3864		WC Charaable 30 112 F			0,500	OCOUTFFFFF
Thread	view.exe(4412): 3864		Wo of a code 33,112 K			0,504	OKOO1FFFFFF
Thread	view.exe(4412). 3864		WS Shared 35,652 K			OCOPC 0.000	CXCOTFFFFFF
Thread	view.exe(4412): 3772		Peak Working Set 83,304 K			0,340	0-001EEEEE
Thread	view.exe(4412): 3620			8		0-244	0-001EEEEE
Thread	view.exe(4412): 3620		12			0.450	0-001EEEEE
Thread	view.exe(4412): 3408			OK 0	Cancel	OCAF4	0.001FFFFF
Thread	View.exe(4412): 3406	1.0			11.	0.100	C. CONFEFFE

FIGURE 5: 'VIEW' PROCESS CONDITIONS AT START OF WINDOWVIEWER

		manie			A			
							Private Bytes	GDI Objects
							3,776 H	(0
							2,056 8	(4
							1,240 8	(8
							45,500 H	853
			0.55				65,692 8	(782
							14,720 1	193
							96,012 8	(124
E 💓	vm.exe	4884	1.10 lr	I ouch WindowMaker	Invensys Systems	, Inc.	49,220 1	563
E	view.exe	4412	3.85 k	Touch Window Viewe	nvensys Systems	, Inc.	324,584 P	9,991
-	Thd.exe	5048	k	Touch Historical Data	Server Invensys Systems	, Inc.	3,584 H	(8
GDI	View.exe	4240	2.20 G	DIView	NirSoft		2.924 1	(74
	exp.exe	4040	9	veintemals Princess Fi	nlorer Sveintemale - www	v eveintemale com	3,428 8	(4
010	procexp64 exe	4380	view.exe:4417	Properties		- ×	28.924 1	588
11 msp	aint.exe	4164		C		tool of the second seco	39.004 H	(164
[] iusched	exe	4252	TCP/IP	Security	Environment	Strings	1.348 8	(4
(m) Tunesh	Helper exe	4292	Image	Performance	Performance Graph	Threads	7.492 1	(11
I Udater	JI exe	4296					3.076 1	35
McT	frav exe	4408	CPU		I/O		7 528 1	97
alammo	T exe	3908	Priority	8	I/O Priority	Normal	15 352 6	(8
ammc ex	· · · · · ·	4140	Kernel Time	0:00:22.089	Reads	15,467	23 328 4	392
0.0	~	4146	Licer Time	0.00.44 304	Read Delta	41	20,0201	
Туре	Name *		Total Time	0:01:06 394	Read Bytes Delta	11.7 KB	Hand	e Access
Thread	view.exe(4412): 3404		Cudes	104 001 007 000	Weiter	4 201	0x288	3 Ox001FFFFF
Thread	view.exe(4412): 3404		Cycles	124,981,687,890	writes	4,201	0x480	0x0000084A
Thread	view.exe(4412): 3404		Virtual Memor	Y	Write Delta	7	Ox5A8	3 Ox001FFFFF
Thread	view.exe(4412): 3252		Private Bytes	324,592 K	Write Bytes Delta	703 B	0x9E4	0x001FFFFF
Thread	view.exe(4412): 3212		Peak Private	Bytes 325,440 K	Other	98,597	0x984	0x001FFFFF
Thread	view.exe(4412): 3212		Victual Cine	560 750 V	Other Delta	254	OxC94	0x001FFFFF
Thread	view.exe(4412): 3128		virtual Size	300,732 K	Other Deita	234	0xA00	Cx001FFFFF
Thread	view.exe(4412): 3128		Page Faults	267,504	Other Bytes Delta	4.4 KB	0xA10	0x001FFFFF
Thread	view.exe(4412): 300		Page Fault De	elta 93	Handles		0x920	0x001FFFFF
Thread	view.exe(4412): 2536		-		Handles	1.091	0x464	0x001FFFFF
Thread	view.exe(4412): 2536		Physical Memo	лу	norues .	1,091	0x474	0x001FFFFF
Ihread	view.exe(4412): 1988		Memory Prior	ity 5	GDI Handles	9,991	0xD90	0x001FFFFF
Process	Mew.eoe(4412)		Working Set	336,396 K	USER Handles	4,309	OctoA/	
Process	view.exe(4412)		WS Privat	294,400 K	-		0x10D8	UXUUTFTFFF
Process	VIEW 808(4412)		WC Character	able 41.000 K			UK 115	
Descars	View exe(4412)		ws share	able 41,996 K			UKT IS	0-00151555
Key	HKII		WS Share	d 38,444 K			UKF6	
Key	HKI M\SYSTEM\Control	Set001\Services\Win Sack	Peak Working	Set 336,408 K			0-20	0-00050035
Key	HKI M\SYSTEM\Control	Set001\Services\WinSock2	1	ana			0-295	0x000F003F
Key	HKI M\SYSTEM\Control	Set001\Services\crunt22				0	0~403	3 0x00020019
Key	HKLM\SYSTEM\Control	Set001\Control\SESSION N			OK	Cancel	0x24	4 0x00000001
					-		0.42	

FIGURE 6: 'VIEW' PROCESS CONDITIONS AT CERTAIN POINT AFTER WINDOWVIEWER HAS BEEN RUNNING FOR A TIME

Summary

The cause of memory, GDI Handle and other resource leaks can come from many different aspects in programming practices. When developing a custom Client Control, it is always a good practice to remember to use implicit and explicit ways to dispose of (even override Standard Dispose methods) the resources allocated during the

Validating Custom Developed Client Controls for Use in InTouch®

initialization and run of the Client Control.

The information provided on this website (Click HERE) provides guidelines for implementing Finalize and Dispose methods to clean up unmanaged resources.

C. He, E. Xu

Tech Notes are published occasionally by Wonderware Technical Support. Publisher: Invensys Systems, Inc., 26561 Rancho Parkway South, Lake Forest, CA 92630. There is also technical information on our software products at Wonderware Technical Support.

For technical support questions, send an e-mail to wwsupport@invensys.com.

Back to top

©2011 Invensys Systems, Inc. All rights reserved. No part of the material protected by this copyright may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying, recording, broadcasting, or by anyinformation storage and retrieval system, without permission in writing from Invensys Systems, Inc. Terms of Use.