



Proudly Presented by
VTScada[®]
Trihedral
A Delta Group Company

Getting Right to the Root of the Problem

**By Keith Donaldson BSc (Hons)
Director of Business Development -
Trihedral UK Limited**

SCADA FEST  **2025**  **ACCELERATE**



Advanced Diagnostic Tools

Take a tour of the debugging and diagnostic tools available to you to both troubleshoot issues and pre-emptively avoid problems.

- This talk is not about looking at problems with your plant or process
- It will look at the tools available to help diagnose and help prevent problems with your SCADA system
- Discuss what diagnostics to provide when reporting a problem
- Describe the reasons why we ask for them



Remote Procedure Calls

- Remote Procedure Call
- Used to invoke (Call) a piece of code (Procedure) on another machine (Remote)
- Colloquially used to describe the Transfer of Data between VTScada Servers
- An example RPC:

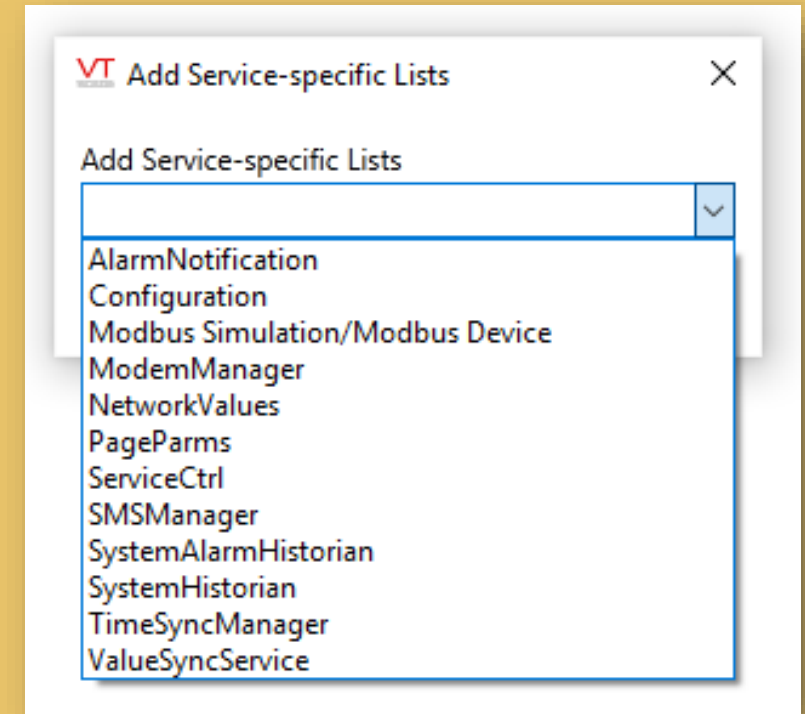
```
SetValue (Station4\Level, 42)
```

- Called to set variable Level in the Station4 Object to the value 42



RPC Services

- VTScada split into different Services
- Used to keep data Synchronised across machines
 - Services Synchronise bi-directionally
 - GetClientChanges
 - GetServerChanges
- Each Service runs independently
- Each Service can have its own Server List
- RPC Manager is responsible for overseeing all the RPC Services
- Drivers can specify Driver List parameter





Viewing Communications connections between VTScada Servers

VT Inter-machine sockets

Machine: KEITHLAPTOP/ 192.168.202.1		Local	Queue length: 0				
Discards	Floods	Execs	CtxErrs	ModErrs	RefCounts		
6	0	217	14	0	16		

Machine: KEITHLAPTOPVPM/ 172.17.240.217		Remote	Queue length: 1					
Discards	Floods	Execs	RemVersion	Session	Lost	SocketsOK	Resends	RefCounts
0	0	0	12.0139	Open	0	1	0	28
Socket: KEITHLAPTOPVPM/ 172.17.240.217								

Machine: KDNB/ 209.17.116.163		Remote	Queue length: 16				

Machine: KDNBVM/ 209.17.116.163		Remote	Queue length: 16				

Machine: ANOTHERSERVER/ 209.17.116.163		Remote	Queue length: 17				

Close



Viewing the Status of RPC Services

VT Services

Advanced Diagnostics VTScada

Service Status Service Architecture

Service	Server List	Current Server	This Workstation	Sticky
AlarmNotification	Default	KEITHLAPTOP	* Current Server	No
Configuration-Advanced Diagnostics	Default	KEITHLAPTOP	* Current Server	No
DriverDiscovery	Default	KEITHLAPTOP	* Current Server	No
Modbus Simulation\Modbus Device	Default	KEITHLAPTOP	* Current Server	No
ModemManager	Default	KEITHLAPTOP	* Current Server	No
NetworkValues	Default	KEITHLAPTOP	* Current Server	No
PageParams	Default	KEITHLAPTOP	* Current Server	No
ServiceCtrl	Default	KEITHLAPTOP	* Current Server	No
SlippyMapService	Default	KEITHLAPTOP	* Current Server	No
SMSManager	Default	KEITHLAPTOP	* Current Server	No
SystemAlarmHistorian	Default	KEITHLAPTOP	* Current Server	No
SystemHistorian	Default	KEITHLAPTOP	* Current Server	No
TimeSyncManager	Default	KEITHLAPTOP	* Current Server	No
ValueSyncService	Default	KEITHLAPTOP	* Current Server	Yes

14 services

Close

VT Services

Advanced Diagnostics VTScada

Service Status Service Architecture

Service	Server List	Primary Server	Backup Server 1	Backup Server 2
AlarmNotification	Default	* KEITHLAPTOP	KEITHLAPTOPVM	× ANOTHERSERVER Disconnected
Configuration-Advanced Diagnostics	Default	* KEITHLAPTOP	KEITHLAPTOPVM	× ANOTHERSERVER Disconnected
DriverDiscovery	Default	* KEITHLAPTOP	KEITHLAPTOPVM	× ANOTHERSERVER Disconnected
Modbus Simulation\Modbus Device	Default	* KEITHLAPTOP	KEITHLAPTOPVM	× ANOTHERSERVER Disconnected
ModemManager	Default	* KEITHLAPTOP	KEITHLAPTOPVM	× ANOTHERSERVER Disconnected
NetworkValues	Default	* KEITHLAPTOP	KEITHLAPTOPVM	× ANOTHERSERVER Disconnected
PageParams	Default	* KEITHLAPTOP	KEITHLAPTOPVM	× ANOTHERSERVER Disconnected
ServiceCtrl	Default	* KEITHLAPTOP	KEITHLAPTOPVM	× ANOTHERSERVER Disconnected
SlippyMapService	Default	* KEITHLAPTOP	KEITHLAPTOPVM	× ANOTHERSERVER Disconnected
SMSManager	Default	* KEITHLAPTOP	KEITHLAPTOPVM	× ANOTHERSERVER Disconnected
SystemAlarmHistorian	Default	* KEITHLAPTOP	KEITHLAPTOPVM	× ANOTHERSERVER Disconnected
SystemHistorian	Default	* KEITHLAPTOP	KEITHLAPTOPVM	× ANOTHERSERVER Disconnected
TimeSyncManager	Default	* KEITHLAPTOP	KEITHLAPTOPVM	× ANOTHERSERVER Disconnected
ValueSyncService	Default	* KEITHLAPTOP	KEITHLAPTOPVM	× ANOTHERSERVER Disconnected

14 services

Close



Viewing Driver Communications

- Selecting Communications traffic
- Logs to VTSCADA\TraceFiles (database)

Driver Trace for Advanced Diagnostics

Time	Direction	Service Name	Driver Name	Driver Area	Driver Description	Port Name	Data
12:58:31.667	Sent	Modbus Sirm	Modbus Sirm			Modbus Sirm	01 03 00 00 00 06 C5 C8
12:58:31.682	Received	Modbus Sirm	Modbus Sirm			Modbus Sirm	01 03 0C 00 2A 00 00 00 00 00 00 00 00 00 00 19 68
12:58:32.668	Sent	Modbus Sirm	Modbus Sirm			Modbus Sirm	01 03 00 00 00 06 C5 C8
12:58:32.798	Received	Modbus Sirm	Modbus Sirm			Modbus Sirm	01 03 0C 00 2A 00 00 00 00 00 00 00 00 00 00 19 68
12:58:33.668	Sent	Modbus Sirm	Modbus Sirm			Modbus Sirm	01 03 00 00 00 06 C5 C8
12:58:33.674	Received	Modbus Sirm	Modbus Sirm			Modbus Sirm	01 03 0C 00 2A 00 00 00 00 00 00 00 00 00 00 19 68
12:58:34.670	Sent	Modbus Sirm	Modbus Sirm			Modbus Sirm	01 03 00 00 00 06 C5 C8
12:58:34.682	Received	Modbus Sirm	Modbus Sirm			Modbus Sirm	01 03 0C 00 2A 00 00 00 00 00 00 00 00 00 00 19 68
12:58:35.670	Sent	Modbus Sirm	Modbus Sirm			Modbus Sirm	01 03 00 00 00 06 C5 C8
12:58:35.805	Received	Modbus Sirm	Modbus Sirm			Modbus Sirm	01 03 0C 00 2A 00 00 00 00 00 00 00 00 00 00 19 68
12:58:36.672	Sent	Modbus Sirm	Modbus Sirm			Modbus Sirm	01 03 00 00 00 06 C5 C8
12:58:36.687	Received	Modbus Sirm	Modbus Sirm			Modbus Sirm	01 03 0C 00 2A 00 00 00 00 00 00 00 00 00 00 19 68
12:58:37.754	Sent	Modbus Sirm	Modbus Sirm			Modbus Sirm	01 03 00 00 00 06 C5 C8
12:58:37.754	Received	Modbus Sirm	Modbus Sirm			Modbus Sirm	01 03 0C 00 2A 00 00 00 00 00 00 00 00 00 00 19 68
12:58:38.674	Sent	Modbus Sirm	Modbus Sirm			Modbus Sirm	01 03 00 00 00 06 C5 C8
12:58:38.682	Received	Modbus Sirm	Modbus Sirm			Modbus Sirm	01 03 0C 00 2A 00 00 00 00 00 00 00 00 00 00 19 68
12:58:39.677	Sent	Modbus Sirm	Modbus Sirm			Modbus Sirm	01 03 00 00 00 06 C5 C8
12:58:39.730	Received	Modbus Sirm	Modbus Sirm			Modbus Sirm	01 03 0C 00 2A 00 00 00 00 00 00 00 00 00 00 19 68
12:58:40.676	Sent	Modbus Sirm	Modbus Sirm			Modbus Sirm	01 03 00 00 00 06 C5 C8
12:58:40.686	Received	Modbus Sirm	Modbus Sirm			Modbus Sirm	01 03 0C 00 2A 00 00 00 00 00 00 00 00 00 00 19 68
12:58:41.678	Sent	Modbus Sirm	Modbus Sirm			Modbus Sirm	01 03 00 00 00 06 C5 C8
12:58:41.697	Received	Modbus Sirm	Modbus Sirm			Modbus Sirm	01 03 0C 00 2A 00 00 00 00 00 00 00 00 00 00 19 68
12:58:42.679	Sent	Modbus Sirm	Modbus Sirm			Modbus Sirm	01 03 00 00 00 06 C5 C8
12:58:42.797	Received	Modbus Sirm	Modbus Sirm			Modbus Sirm	01 03 0C 00 2A 00 00 00 00 00 00 00 00 00 00 19 68
12:58:43.680	Sent	Modbus Sirm	Modbus Sirm			Modbus Sirm	01 03 00 00 00 06 C5 C8
12:58:43.687	Received	Modbus Sirm	Modbus Sirm			Modbus Sirm	01 03 0C 00 2A 00 00 00 00 00 00 00 00 00 00 19 68
12:58:44.681	Sent	Modbus Sirm	Modbus Sirm			Modbus Sirm	01 03 00 00 00 06 C5 C8
12:58:44.698	Received	Modbus Sirm	Modbus Sirm			Modbus Sirm	01 03 0C 00 2A 00 00 00 00 00 00 00 00 00 00 19 68
12:58:45.682	Sent	Modbus Sirm	Modbus Sirm			Modbus Sirm	01 03 00 00 00 06 C5 C8
12:58:45.718	Received	Modbus Sirm	Modbus Sirm			Modbus Sirm	01 03 0C 00 2A 00 00 00 00 00 00 00 00 00 00 19 68
12:58:46.682	Sent	Modbus Sirm	Modbus Sirm			Modbus Sirm	01 03 00 00 00 06 C5 C8

Driver Trace for Advanced Diagnostics: total records=124



RPC Timings Application

- This application is included with VTScada, but you may need to add it to the VAM
- When run Queries VTScada RPCManager Server to retrieve Statistics
- The IP shown may not be the IP address expected
- Q Delay – the time for a message to propagate the RPC Queue on the local machine
- Average, Best and Worst round-trip times shown
- Tolerance – Link Tolerance section of Setup.ini
- Used to handle the differing response times from one workstation to another, which is often experienced due to CPU loading, but mostly due to link throughput limitations



Trace VTScada Actions Application

- This application is included VTScada, but you may need to add it to the VAM
- Writes data about selected actions to a text file named, "VTSTrace.txt"
- Additionally, VTScada traces all actions to disk, storing the data in a separate text file named, "VTSTraceAll.txt".
- Records in RAM - the number of trace actions saved to RAM prior to being written to the VTSTrace.txt file
- The Start/Stop button enables you to start and stop the tracing



- Located in the Data\TwilioLogs\
- Timestamped Rolling Log Files

```
"Parameters": null,  
  "Body": {  
    "VoiceUrl": "http://#####:#####@0445ba83.ngrok.io:80/Twilio/TELECOM/Twilio/Answer",  
    "StatusCallback": "http://#####:#####@0445ba83.ngrok.io:80/Twilio/TELECOM/Twilio/Status",  
    "FallbackUrl": "http://#####:#####@0445ba83.ngrok.io:80/Twilio/TELECOM/Twilio/Error",  
    "VoiceMethod": "POST",  
    "SmsUrl": "http://#####:#####@0445ba83.ngrok.io:80/Twilio/TELECOM/Twilio/SMSMessage",  
    "SmsMethod": "POST",  
    "SmsFallbackUrl": "http://#####:#####@0445ba83.ngrok.io:80/Twilio/TELECOM/Twilio/SMSError",  
    "SmsFallbackMethod": "POST",  
    "ApiVersion": "2010-04-01",  
    "StatusCallbackEvent": [  
      "initiated",  
      "ringing",  
      "answered",  
      "completed"  
    ]  
  },  
}
```



The Source Debugger

Viewing the code, variables and more

- Shipped with every VTScada Installation
- For Advanced Users only (not aimed at end users)
- Complex but useful

The screenshot shows the 'VT Debug Advanced Diagnostics - [Summary (Live)]' window. The interface includes a navigation pane on the left with a tree view of system components, a main summary pane on the right, and a variable watch pane at the bottom.

System Summary (Live)

Operating System: Windows version 10.0 (64-bit, 8 CPUs)

VTScada Info: Memory used: 412,588,592 bytes

Version: 11.3.22 (64-bit)
Serial number: 1
License type: Full
Free updates expiry: 16 Jul 2100
Max tags allowed: Unlimited
Max browser clients allowed: Unlimited
Alarm notification: Enabled
Memory used: 412,588,592

Applications: 3 running, 10 in VAM

Running

[System Library](#)
[Source Debugger](#)
[Advanced Diagnostics](#)

Stopped

Modules:

Static Scope Call

Name	Value
VTSDB	System
AreaList	Dictionary (2)
RealmDisplayVarDB	Dictionary (0)
LocalGUID	"{1111-2222-3333-4444}"
RemoteGUID	"{1111-2222-3333-4444}"
AppTitle	Advanced Diagnostics
AppStartupTime	34,367,000
StartTan	StartTan
Local Instance	Auto Global

Count: 8206

Location/Thread	Module	State	Statement	Name	Value
-----------------	--------	-------	-----------	------	-------

BreakPoints Use List Set List

Watch 1 Watch 2 Watch 3 Watch 4



Source Debugger Application – Summary Tab

- Provides specific information about the Operating System
- VTScada Information
 - Version
 - Serial Number
 - Licence Type Maintenance Expiry
 - Tag Limit
 - Thin Client Limits
 - Alarm Notification status
 - Memory Used

System Summary

(Live)

Operating System: Windows version 10.0 (64-bit, 8 CPUs)

VTScada Info: Memory Used: 155,970,792 bytes

Version 12.0.05 (64-bit)

Serial Number: 1

License Type: Full

Free Updates Expiry: 16 Jul 2100

Max Tags Allowed: Unlimited

Max Browser Clients Allowed: Unlimited

Alarm Notification: Enabled

Memory Used: 155,970,792 bytes



Using the Source Debugger to examine VTScada Threads

- List Threads
- Refresh button shows a live view of Thread Execution
- Module, State and Statement show exactly where code execution is running
- Double Click to navigate Source Debugger to source code line

The screenshot shows a window titled "Threads:" with a subtitle "22 running threads". It contains a table with the following columns: Name, Thread ID, Status, Module, State, and Statement #. The table lists several threads, including System, NTEventErrorThread, RPC, and several threads related to VTScada 12.0.05.

Name	Thread ID	Status	Module	State	Statement #
System	26628	Idle	CheckActivationCode	CheckActivationCode	0
NTEventErrorThread	2860	Idle	LogNTEventThread	Main	0
RPC	15184	Idle	MachineNode	Remote	4
C:\VTScada 12.0.05\Activation\	5624	Idle	Watcher	Main	0
Tag Persisting	34596	Idle	PersistWorker	Main	0
Activation	30992	Idle	Activation	Main	0
C:\VTScada 12.0.05\VTS\	15772	Idle	DemoLicenseManager	Init	1
C:\VTScada 12.0.05\VTScada\	42712	Idle	DemoLicenseManager	Init	1



Source Debugger – Execution History

- Shows the 10,000, 100,000 or 1,000,000 history of execution
- Only displays the active threads which have executed any code within the sample period
- Can filter by Thread to aid viewing
- Double Click to navigate Source Debugger to source code line

Execution History:

Filter Options

Update Filter

Number of Entries: 10,000

Show

All History
 <No Module Selected>
 <No Object Selected>

Thread: All Threads

Hide this Debugger Thread

Scroll List by Thread: [Up] [Down]

Entry	Time	Module Name	State	Statement #	Thread Name	Thread ID
93	12:16:58.653	DrawApplicationList	Main	17	Application Manager	4068
94	12:16:58.578	DrawApplicationList	Main	17	Application Manager	4068
3988	12:16:57.727	CheckActivationCode	CheckActivationCode	2	System	26628
3989	12:16:57.727	CheckActivationCode	CheckActivationCode	1	System	26628
3990	12:16:57.727	CheckActivationCode	CheckActivationCode	0	System	26628
4503	12:16:57.667	GetParmPhraselmpl	Main	1	Application Manager	4068
4504	12:16:57.667	GetParmPhrase	Main	0	Application Manager	4068
4505	12:16:57.667	DrawApplicationLists	Main	29	Application Manager	4068
4506	12:16:57.667	DrawApplicationLists	Main	29	Application Manager	4068
4507	12:16:57.667	DrawApplicationLists	Main	65	Application Manager	4068
4508	12:16:57.667	CommaFormat	FormatNumber	6	Application Manager	4068
4509	12:16:57.667	CommaFormat	FormatNumber	5	Application Manager	4068
4510	12:16:57.667	AddCommas	AddCommas	5	Application Manager	4068
4511	12:16:57.667	AddCommas	AddCommas	4	Application Manager	4068



VTScada Script Code Statement Numbering

- Each statement in VTScada is assigned a unique Statement Number
- Statements are Numbered in Steady State first top to bottom (excluding scripts)
- Then each statement in a script is numbered top to bottom
- Placing a breakpoint will reveal the Statement Number of a line of code

```
923 <
924 <===== Operate =====>
925 {
926 }
927 Operate
928 [
929   AnaOper  Module;
930   NewValue;
931   Response;
932   User;
933 ]
934
935 ChkMan [ { Are manual operations allowed ? }
936
937   If Operable ChkMode;
938   If !Operable Done;
939
940 ]
941
942
943 ChkMode [ { Are we in the right operating mode ? }
944
945   Response = \OpMode\NeedMode(\SecurityManager\PrivBitControlOutputs);
946
947   If Response Ask; { got suitable mode - start operate dialogue }
948   [
949     User = \GetUserSession()\SMACSUserName;
950     Response = Invalid;
951     NewValue = Format(100, 5, PickValid(Value, ManValue));
952   ]
953 ]
```

Summary (Live) Start.SRC Analogue.SRC

Location/Thread	Module	State	Statement	Name	Value
Analogue.SRC/#949	Operate	ChkMode	4.		



Using the Source Debugger to Profile VTScada Execution

- Running the Profile
- Flipping the Tree
- Time vs Percentage
- Statistics
- CodeID and linking to Source Code
- Save and Load Profiles

VT Profiler

Profiler:

Running Profile

Duration: ---:--:--

Last Profile

Start Time: 12:27:25.923
Stop Time: 12:27:28.347
Duration: 00:00:02.424

Statistics

Statement Count: 323
Normal Pending: 5
Timers Pending: 968
Priority Pending: 1
Scheduler Lock: 505.70 (20.86%)

Messages

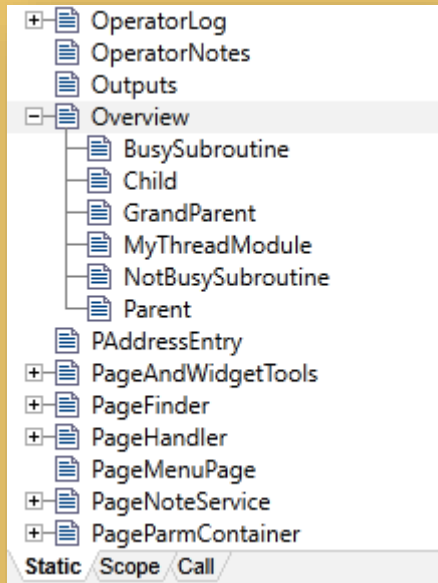
Status: Idle

Counts	Total Time	Self Time	Avg Time	Stalled Time	Module Name	Code ID
3628	330.03	7.35	0.09	0.00	Script Threads	Script Threads
6050	235.54	28.40	0.04	0.00	ReadLowest (UseLowest)	smacs\SMACS Core\.sync\ArbHandler.SRC\Arbitrate\ReadLowest 58141
78	41.81	2.54	0.54	0.00	VTSRead (Read)	smacs\SMACS Core\.sync\IPSCMDriver.SRC\IPSCMDriver\VTSRead 098960
7509	14.76	14.76	0.00	0.00	Priority List	Priority List
384	6.67	6.11	0.02	0.00	ReadLowest (UseLowest)	smacs\SMACS Core\.sync\ArbHandler.SRC\Arbitrate\ReadLowest 57932
2	3.03	0.19	1.51	0.00	DrawRow (DrawRow)	Source Debugger\.sync\Profiler.SRC\Profiler\InfoBox\DrawRow\DrawRow 00006-00
3	2.59	2.59	0.86	0.00	StatusBar (Show)	smacs\SMACS Core\.sync\Pages\SMACSPage.SRC\SMACSPage\StatusBar 51156
2	2.10	2.10	1.05	0.00	VTSInfo (Show)	Source Debugger\.sync\Summary.SRC\SummaryTab\Summary\VTSInfo\Show 00011-0000
135	2.06	2.06	0.02	0.00	Read (Main)	VTS\.sync\VTSDrvr.WEB\VTSDriver\ReadBlock\Read\Main 00007-00031
192	1.93	1.31	0.01	0.00	VTSRead (ArbitratedRead)	smacs\SMACS Core\.sync\IPSCMDriver.SRC\IPSCMDriver\VTSRead 107170
62	1.35	1.35	0.02	0.00	DrawRow (DrawRow)	Source Debugger\.sync\Profiler.SRC\Profiler\InfoBox\DrawRow\DrawRow 00005-00
58	1.10	1.10	0.02	0.00	DrawRow (DrawRow)	Source Debugger\.sync\Profiler.SRC\Profiler\InfoBox\DrawRow\DrawRow 00004-00
96	1.06	1.06	0.01	0.00	Read (Main)	VTS\.sync\VTSDrvr.WEB\VTSDriver\ReadBlock\Read\Main 00003-00000
2	0.95	0.10	0.47	0.00	SummaryTab (Main)	Source Debugger\.sync\Summary.SRC\SummaryTab\Main 00017-00000
288	0.89	0.45	0.00	0.00	VTSRead (ArbitratedRead)	smacs\SMACS Core\.sync\IPSCMDriver.SRC\IPSCMDriver\VTSRead 107670
2	0.58	0.58	0.29	0.00	VTSInfoHeader (Main)	Source Debugger\.sync\Summary.SRC\SummaryTab\Summary\VTSInfoHeader\Main 0000

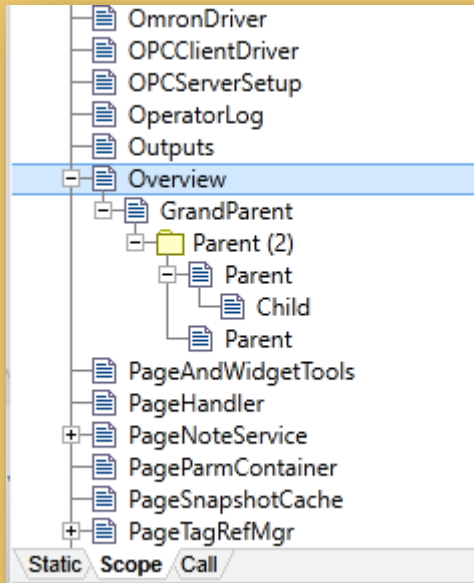


Navigating the Source Debugger Module Trees

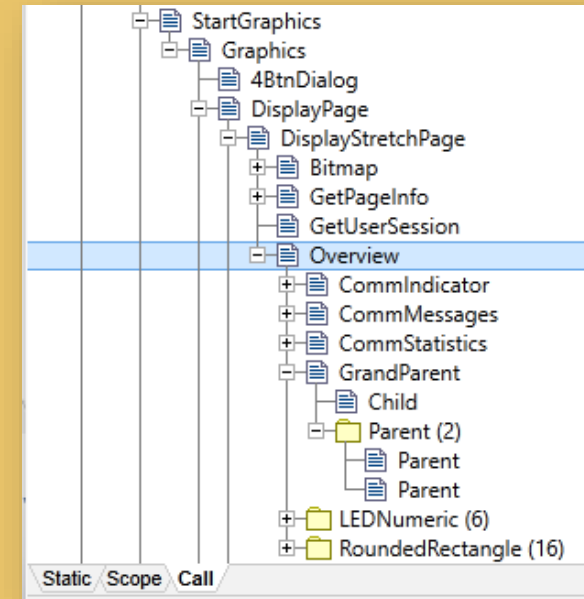
- Three different views of the Module Tree, Static, Scope and Call tree



Static –
As declared



Scope –
Resolve Variables

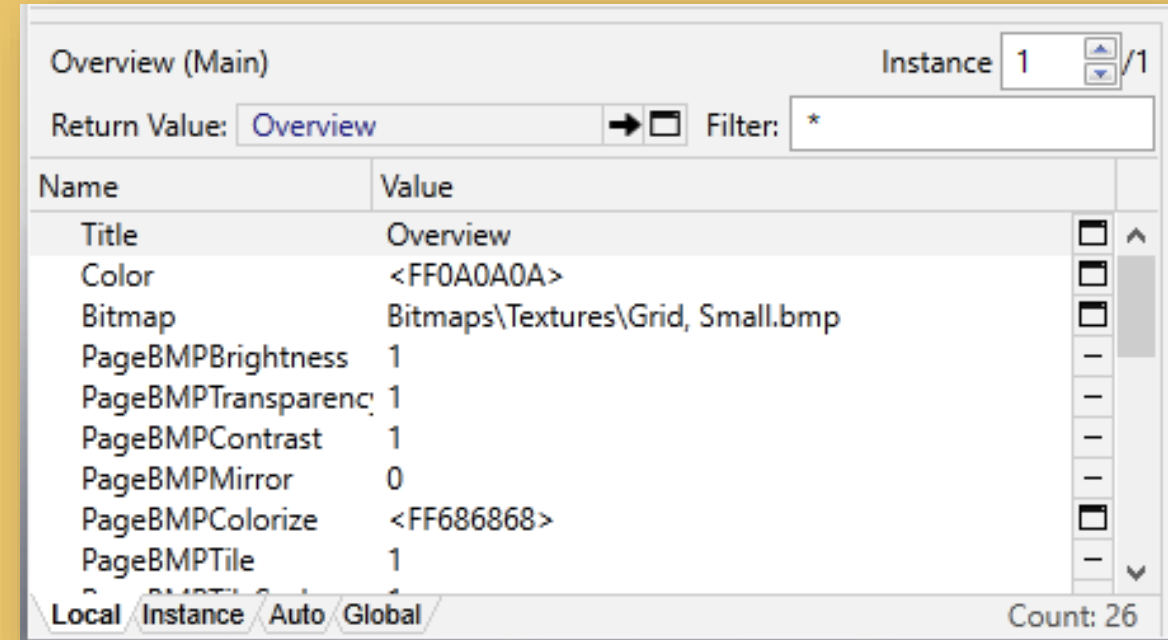


Call –
Module Lifetime



Inspecting Variables using the Source Debugger Variable Window

- Module Name & Current State
- Number of Instances
- Return Value
- Filter
- Local / Instance / Auto / Global
- Variable Count
- Variable Name and Value (Editable)





Source Debugger – The Breakpoints Window

- Breakpoints Place and disable
- Stepping through Code
- Variable Breakpoints
- Conditional Breakpoints
- Use List and Set List
- Module State and Statement
- NEW! Step Out

Location/Thread	Module	State	Statement
● Overview.SRC/#267	Overview	Main	218 .
● Overview.SRC/#277	Overview	Main	33 .
● K	Overview	On Set	
● ? Count	Overview	On Set	
✦ Break on thread DisplayManager setting Count			
\Count == 3			OK
BreakPoints Use List Set List			



Source Debugger – The Watch Window

- Add a Variable Watch
- Adding a Watch Expression
- Multiple Watch Tabs
- NEW! Script Statement Calls

Name	Value	
? Count	4	-
\Count > 3	1	-

Watch 1 / Watch 2 / Watch 3 / Watch 4



Source Debugger – Code Coverage

- Enable code coverage to see what code has executed

The screenshot shows a source debugger interface with a file explorer on the left and a code editor on the right. The file explorer lists various modules, with 'Overview' expanded to show sub-modules like 'BusySubroutine' and 'NotBusySubroutine'. The code editor displays a C++-style event handler function for a GUI button. The code is annotated with green and yellow markers indicating execution status. Line 250 is highlighted in pink, and line 280 is highlighted in red. The code includes conditional logic for button visibility and a while loop for processing the button click.

```
248 "Test Modules");
249
250 If GUIButton(539, 150, 629, 127 { Outline of the button },
251             1, 1, 1, 1, 1 { No scaling },
252             0, 0, 1, 0 { No movement; visible },
253             64 + 4, 1, 0 { Triggered by left mouse
254             button release },
255             GetSystemColor(15){ Windows button face color },
256             GetSystemColor(20){ Windows button highlight color},
257             GetSystemColor(16){ Windows button shadow color },
258             GetSystemColor(18){ Windows button text color },
259             0, 0 { Windows standard attributes },
260             "If 1",
261             "If 1",
262             0, 0, 1, 2 { No variable assignment });
263 [
264     On = !On;
265 ]
266
267 If On;
268 [
269     K = 100;
270     WhileLoop(K > 0,
271             A = 0;
272             B = 0;
273             NotBusySubRoutine();
274             BusySubRoutine();
275             K--;
276     );
277     I++;
278 ]
279
280 If GUIButton(539, 186, 629, 163 { Outline of the button },
281             1, 1, 1, 1, 1 { No scaling },
282             0, 0, 1, 0 { No movement; visible },
283             64 + 4, 1, 0 { Triggered by left mouse
284             button release },
285             GetSystemColor(15){ Windows button face color },
286             GetSystemColor(20){ Windows button highlight color},
287             GetSystemColor(16){ Windows button shadow color },
288             GetSystemColor(18){ Windows button text color },
289             0, 0 { Windows standard attributes },
290             "If 1",
291             "If 1",
292             0, 0, 1, 2 { No variable assignment });
```



Dump Files

- Warning – creating a dump file will halt all system activity
- Often called a crash dump (Win32 system debugger, Dr Watson, may be configured to produce a file after an unrecoverable program error)
- UserDump in the VTScada\DumpTools directory
- Right Click on the VTS.EXE process in Task Manager
- By default stores dumps in %appdata%\local\Temp\VTS.DMP
- Click on the Create Dump File icon on the Source Debugger Window
- Allows path to be specified



Getting the most from Technical Support

- You can **NEVER** provide too much information!
- Dump files zipped and uploaded to FTP site (<ftp://ftp.trihedral.com>)
- Traces from both client and server (when working with a RPC issue)
- Information, Version Number, Operating System, 64bit / 32bit
- Information to reproduce
- A changeset of the application
- User name and password for admin level access to the application



Thank you for attending this session!

- Fill out the 2-question feedback survey via the App! *Ukova*
- Especially if you need a Continuing Education Credit or Professional Development Hours certificate.
- Don't have the app, fill out the paper form included in your conference kit and hand into Natashaia.

SAVE THE DATE!

MARCH 23-27, 2026 | ORLANDO, FLORIDA

SCADA FEST > 2025 < ACCELERATE