

DATRAN VI SCADA Software Training Course

Course Notes & Reference Material

1.0 CONTENTS

1.0	CONTENTS	2
2.0	QTECH TRAINING OBJECTIVES	5
3.0	OVERVIEW	6
3.1	WHAT IS DATRAN VI?	6
3.2	CAPABILITIES OF DATRAN VI.....	7
4.0	DATRAN VI SERVICES	8
4.1	OVERVIEW	8
4.2	TREE BASE SERVER	8
4.3	DATRAN SERVER MANAGER	9
4.4	DATRAN ALARM SERVICE.....	9
4.5	DATRAN PAGING SERVICE OR QPAGER3.....	10
4.6	XML DATA INTERFACE.....	10
4.6.1	<u>SMS Service</u>	10
4.6.2	<u>Named Pipe Server</u>	10
4.6.3	<u>Replication</u>	11
4.7	SMS DIRECT.....	11
4.7.1	<u>SMS Direct Standard</u>	11
4.7.2	<u>SMS Direct Full</u>	11
4.8	QTECH COMMS.....	11
4.9	DATRAN DERIVED VALUE GENERATOR (DVG)	11
4.10	DATRAN ADO DATABASE CONNECTIVITY MODULE (MSDE & MS SQL SERVER).....	11
4.11	DATRAN INSQL DATABASE CONNECTIVITY MODULE (WONDERWARE INDUSTRIALSQL/INSQL SERVER).....	12
4.12	TCP SERVER	12
4.13	DATRAN RTU CONSOLE	12
4.14	DATRAN HISTORICAL DATA INTERFACE	12
4.15	DATRAN DDE INTERFACES (CLIENT/SERVER).....	12
4.15.1	<u>Background DDE information</u>	12
4.15.2	<u>DDE Server Interface</u>	13
4.15.3	<u>DDE Client Interface</u>	13
4.16	DATRAN INTERFACES	13
4.16.1	<u>Suitelinks</u>	13
4.16.2	<u>DNP3</u>	13
4.16.3	<u>OPC Interfaces (Client / Server)</u>	13
4.17	CONTROLLING DATRAN SERVICES.....	13
5.0	APPLICATIONS – USER INTERFACES	15
5.1	DATRAN APPLICATIONS & SERVICES	16
5.1.1	<u>DATRAN Applications</u>	17
5.1.2	<u>Paging Console</u>	17
5.1.3	<u>DATRAN Supervisor</u>	17
5.1.4	<u>DATRAN Browser</u>	18
5.1.5	<u>DATRAN Diagnostics</u>	19
5.1.6	<u>DATRAN Trending</u>	19
5.1.7	<u>DATRAN Reporting</u>	19
5.1.8	<u>DATRAN DLP IDE</u>	19
5.1.9	<u>DATRANVue</u>	20
6.	DATRAN TRENDING	21
6.1.	SOFTWARE INSTALLATION.....	21

6.2.	FIRST RUN.....	23
6.3.	CREATING A PLOT.....	24
6.4.	LICENSING.....	26
7.	DATRAN REPORTING.....	27
7.1.	SOFTWARE INSTALLATION.....	27
7.2.	FIRST RUN.....	28
7.3.	CREATING A REPORT.....	29
7.4.	LICENSING.....	33
8.	DATRAN SUPERVISOR.....	34
8.1.	SOFTWARE INSTALLATION.....	34
8.2.	LICENSING.....	37
8.3.	CONNECTION TO DATRAN.....	37
9.	INSTALLING AND UPGRADING DATRAN VI.....	39
9.1.	SOFTWARE INSTALLATION.....	39
9.2.	DATRAN SERVICE LICENSING.....	39
9.3.	DATRAN VI – HUMAN MACHINE INTERFACES.....	41
8.3.1	<i>HMI Software & DATRAN</i>	42
9.4.	CLIENT/SERVER INTEGRATION.....	43
10.	THE DATRAN TBD FILE.....	44
10.1.	OVERVIEW.....	44
10.2.	TBD FILE, FILING STRUCTURE STANDARDS & RECOMMENDED LAYOUTS.....	44
10.3.	COMPUTER NODES IN THE TBD FILE AND BASIC FUNCTION/USE.....	46
10.4.	OPERATOR AND OPERATOR GROUPS NODES IN THE TBD FILE.....	47
10.4.1.	<i>Operator Aliases</i>	48
10.4.2.	<i>Operator Groups</i>	48
10.4.3.	<i>Operator</i>	48
10.5.	DATRAN VI – CHANNELS NODES IN THE TBD FILE AND BASIC FUNCTION/USE.....	48
10.6.	DATRAN VI – SYSTEM SCRIPTS NODE IN THE TBD FILE AND BASIC FUNCTION/USE.....	48
10.7.	DATRAN VI - SITES NODES IN THE TBD FILE AND BASIC FUNCTION/USE.....	49
11.	_COMMS_OBJECT_NODES, _ALARM_NODES, I/O, _FORMULA_ AND _COPY_TO_NODES.....	50
11.1.	THE _COMMS_OBJECT_.....	50
11.2.	I/O.....	50
11.3.	_FORMULA_NODES.....	50
11.4.	_COPY_TO_NODES.....	51
11.5.	_ALARM_NODES.....	52
12.	ADDING SITES AND THEIR ASSOCIATED CONFIGURATION ISSUES.....	54
12.1.	SITE TOPICS.....	54
12.2.	EDITING A COMMUNICATIONS CHANNEL.....	54
12.3.	ADDING A NEW COMMUNICATIONS CHANNEL – WORKED EXAMPLE.....	56
12.4.	STATION TEMPLATE.....	57
12.5.	ADDING A NEW SITE TO AN EXISTING SYSTEM.....	58
12.5.1.	<i>Third Party GUI Software Is Used</i>	60
12.6.	ADDING A NEW DISCRETE POINT TO A SITE.....	61
12.7.	ADDING A NEW ANALOGUE POINT TO A SITE.....	65
12.8.	REMOVING A SITE FROM A SYSTEM.....	65
12.8.1.	<i>Permanent Removal</i>	65
12.8.2.	<i>Temporary Removal</i>	66
13.	DATRAN VI – SCRIPTS, TEMPLATE’S AND MACROS IN THE TBD FILE AND BASIC FUNCTION/USE.....	67
13.1.	THE SCRIPT, MACRO AND TEMPLATE NODES.....	67
13.2.	THE SCRIPT EDITOR.....	67
13.3.	SCRIPT TRIGGERS.....	68

13.4.	TEMPLATE KEYS.....	69
14.	DATRAM VI – DISASTER AVOIDANCE AND RECOVERY	71
14.1.	DISASTER RECOVERY AND AVOIDANCE	71
14.2.	DISASTER AVOIDANCE	71
14.3.	ALARMS	71
14.4.	BACKING UP A DATRAM SYSTEM.....	72
14.5.	LOADING A BACKUP TBD	72
15.	DATRAM VI – THE DATRAM PROPERTIES HIERARCHY	74
16.	DATRAM VI – THE ALARMING SYSTEM.....	76
16.1.	THE ALARM SYSTEM.....	76
16.2.	DATRAM VI – THE ANNUNCIATOR	77
17.	ADO CONNECTIVITY MODULE.....	80
17.1.	SYSTEM REQUIREMENTS:.....	80
17.2.	OVERVIEW:.....	80
17.3.	MICROSOFT SQL SERVER AND SQL SERVER 2005 EXPRESS.....	81
17.4.	SETTING UP DATRAM VI TO LOG TO A SQL DATABASE.....	82
17.4.1	<i>Installing MS SQL Server 2005 Express Ed. on the DATRAM Server.....</i>	82
17.4.2	<i>Adding a SQL Server 2005 Express database to DATRAMVI.....</i>	82
17.4.3	<i>DATRAM ADO Database Connectivity Properties.....</i>	83
17.4.4	<i>System Scripts – SQL Database Logging Properties.....</i>	83
17.4.5	<i>Logging Data with DATRAM VI.....</i>	84
17.4.6	<i>Database Alarms.....</i>	85
17.4.7	<i>Database Backups.....</i>	85
17.4.8	<i>Manually querying the MSDE / MS SQL Database.....</i>	86
18.	INSQL DATABASE CONNECTIVITY MODULE	87
18.1.	SYSTEM REQUIREMENTS:.....	87
18.2.	OVERVIEW:.....	87
18.3.	WONDERWARE INDUSTRIALSQL/INSQL SERVER.....	88
18.4.	SETTING UP DATRAM VI TO LOG TO AN INSQL DATABASE	88
18.4.1	<i>Installing Wonderware IndustrialSQL Server on the DATRAM VI server.....</i>	88
18.4.2	<i>Adding an InSQL database to DATRAMVI.....</i>	88
18.4.3	<i>DATRAM InSQL Database Connectivity Properties.....</i>	89
18.4.4	<i>Logging Data with DATRAM VI.....</i>	89
18.4.5	<i>Database Alarms.....</i>	90

2.0 QTECH TRAINING OBJECTIVES

This training manual contains references to current and legacy versions of DATRAN SCADA Software and Databases. There are a number of older versions of the software still currently in use, and the trainee must be aware of functionality and compatibility issues that exist these and the latest releases of DATRAN.

Although this manual contains comprehensive detail and background to the DATRAN Software and Databases, it is strongly recommended that specific version release notes be consulted for further information.

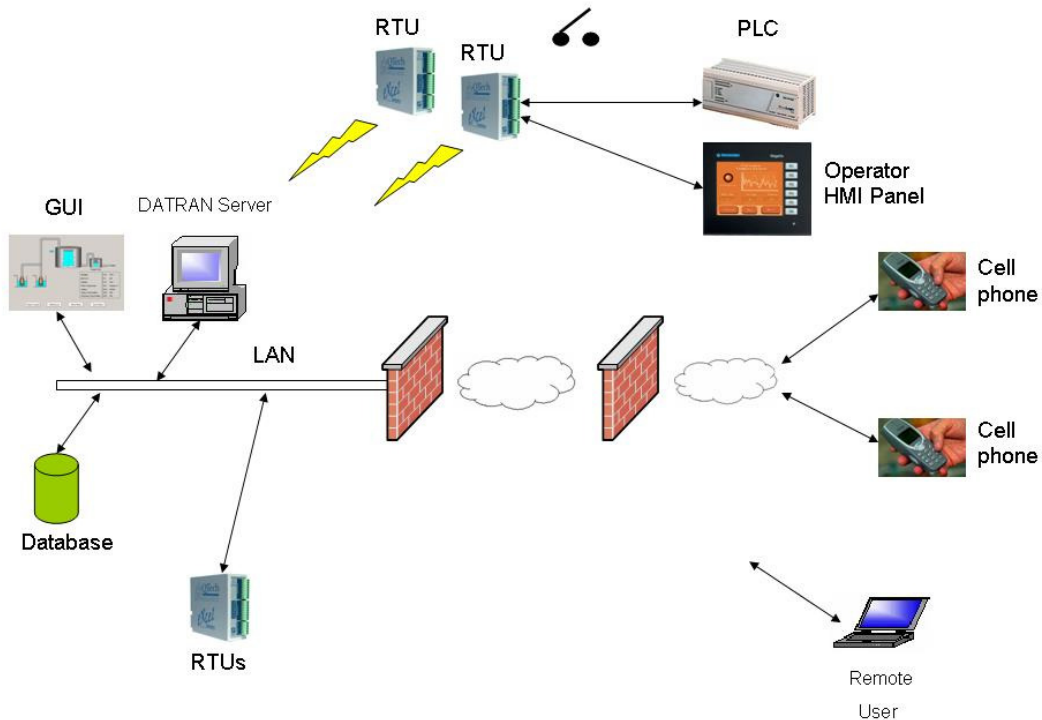
Upon successful completion of this training, the trainee will;

- Gain an appreciation of the various DATRAN VI windows SCADA services and understand what each does in the complete SCADA system
- Understand what user interface applications are available and how to use the major ones
- Understand which files need to be backed up to ensure that a system can be restored in the case of a catastrophic failure of the computer that DATRAN is running on
- Have an appreciation of the various steps involved in the creation of a new site on the telemetry system
- Be able to install, license and use the common DATRAN user applications, DATRAN Supervisor and DATRAN Trending

3.0 OVERVIEW

3.1 What is DATRAN VI?

- DATRAN VI is a Windows based SCADA engine. A SCADA program allows data to be gathered from local or remote devices it allows devices to be controlled and provides facilities for displaying this data.
- DATRAN VI runs as a number of Windows Services and user applications. Windows Services are programs that run in the background & usually does not have a GUI. A user does not have to be logged on. User applications are programs that a user can start or stop. In DATRAN for the most part these are associated with display of information.
- Some of the user applications have been designed to be backwardly compatible with obsolete operating systems such as Windows 95 or Windows 98. (i.e. if an old networked workstation to a DATRAN server was required to be used)



3.2 Capabilities of DATRAN VI

The DATRAN VI SCADA system is very versatile and has many features, which are being developed constantly. Features include...

- ❑ Database Logging
- ❑ Data Graphing
- ❑ Paging to Cell Phone/Pagers/Email
- ❑ SMS to Cell Phones with web based tracking
- ❑ Alarm Handling
- ❑ Alarm Acknowledgement (DTMF or SMS)
- ❑ OPC , DDE and interfaces to external Graphical User Interface (GUI)
Programs such as iFIX and InTouch, Citect
- ❑ Interface to the Base station of several PLC types
- ❑ Security monitoring and control
- ❑ Remote RTU Data Logging
- ❑ Replication between Base stations

The DATRAN VI engine has mostly been implemented in the local authority market but is capable of being implemented in many other SCADA applications and is indeed being used in a wide range of solutions today. There are also new products being developed to target smaller application of the system, to make it a viable option for factories and small business applications. Some of the varying solutions, which have been implemented around the country, are...

- ❑ Control of the extractor fans in the Lyttelton Tunnel.
- ❑ Part of the Stirling Cheese Factory.
- ❑ Control of the Ashburton Area's Electricity Load Management.
- ❑ Control of a generator system on Great Mercury Island.
- ❑ Pulse accumulation and transfer in the Natural Gas Fields
- ❑ Monitoring and automation of mining operations in New Zealand and Australia
- ❑ Control and Monitoring of the water intake for the Anchor milk plant in Reperoa
- ❑ Collection of meat inspection data for Alliance Group Ltd.

4.0 DATRAN VI SERVICES

4.1 Overview

There are several services that make up the DATRAN VI SCADA system. However not all of these are used or needed on every base station.

- ❑ DATRAN Tree Base Server
- ❑ DATRAN Server Manager
- ❑ DATRAN Alarm Service
- ❑ DATRAN Derived Value Generator
- ❑ DATRAN QTech Comms
- ❑ DATRAN DDE Server
- ❑ DATRAN DDE Client
- ❑ DATRAN Historical Data Interface
- ❑ DATRAN RTU Console
- ❑ DATRAN ADO Database Connectivity Module
- ❑ DATRAN Connectivity Modules (DNP3, DF1, Modbus, GE Fanuc)
- ❑ DATRAN XML Data Interface (XDI) [Includes SMS Service & Replication]
- ❑ DATRAN TCP Server
- ❑ DATRAN OPC I/O Driver Server
- ❑ DATRAN OPC Client
- ❑ DATRAN InSQL Database Connectivity Module

4.2 Tree Base Server

At the core of DATRAN is a service called DATRAN Server (TBSERV). This stores the entire configuration for the DATRAN System and the current state of the monitored equipment. The Tree Base server is responsible for co-ordinating the communication of the other services with one another. It also maintains the integrity of the configuration settings in a tree form.

4.3 DATRAN Server Manager

DATRAN Server Manager undertakes various maintenance and system integrity monitoring tasks as follows:

- Periodically saving the contents of DATRAN Server to a TBD (Tree Base Data) file.
- Monitoring the integrity of the various services.
- Supplies a heartbeat that can be used to detect failure of DATRAN VI or the computer that it is running on.

This is the service that deals with all the saving of files and checking of the system to make sure that all features are still running, as they should. It also contains information for the back up paths and locations and the location of where log files are saved etc.

4.4 DATRAN Alarm Service

The Alarm Service is a service program that provides sophisticated generation and management of alarms and events within the DATRAN Server.

Different classes of alarms can be created that specify the sequence of actions and escalations to be performed when an alarm event takes place. These actions can be performed by the Alarm Service itself or by client programs without the client being 'aware' of the existence of Alarm Service.

For example, the Alarm Service could wait ten seconds for the alarm condition to return to normal before escalating the alarm further. If it has not 'returned to normal' within that time, the service could create a text message and have that message sent to the on-duty operator. If an acknowledgment or a 'return to normal' is not received within, say fifteen minutes, the alarm service could take action itself by running a script that sets a digital output on a remote telemetry unit that closes a valve. Virtually any DATRAN client program can be used to annunciate the alarm event through Alarm Service.

Alarm Service provides several 'standard' Annunciators. It can print a message to a serial printer; play a sound file through a sound card; sound a 'beep' using the computer's internal speaker; or just wait a period of time for the alarm to 'return-to-normal' before escalating further.

This service handles all the alarming nodes within the TBD file and performs all the escalations within alarm classes.

4.5 DATRAN Paging Service or QPager3

The DATRAN Paging Service is a Service program that provides paging and telephony services for a DATRAN VI system. It was primarily used to communicate with Telecom's PACNET paging system, but Telecom has discontinued that service. Paging service can still be useful for the following:

- ❑ To send email messages
- ❑ To allow operators to dial into DATRAN and acknowledge alarms via a voice menu system
- ❑ To allow ordinary telephones to be dialled and pre-recorded messages to be played when the telephone is answered.

The gateway integrates with the DATRAN Server to allow messages to be paged, or pre-recorded messages to be played when specific conditions exist in the database, such as the occurrence of an alarm. Alarms generated by the Alarm service module can be acknowledged remotely through a touch-tone phone.

Currently, the only modem supported for the playing of pre-recorded messages and dial-up DTMF acknowledgment is the Zyxel U-1496 series modem, D-Link 560E & 560E+ modems. Other models that support the commands in the TIA TR29.2 committee PN2986 document should also work but have not been tested.

This is the service that deals with connection to the paging modem and the processing of operators and protocols and passing on pager calls from the Alarm Service to the outside world.

4.6 XML Data Interface

The XML Data Interface contains a group of sub-services that all use XML as a basis for their communications protocol.

4.6.1 SMS Service

This provides the ability to send and receive text messages from the DATRAN VI system. There is a standard Alarm Annunciator that will send an SMS to a given operator's cell phone. Operators can then reply to the text message to acknowledge the alarm.

This service can also be used to query or set nodes within DATRAN from the operator's cell phone.

This functionality requires an Internet connection in order to communicate with DATASHED (our SMS Gateway)

4.6.2 Named Pipe Server

This provides connection with QTech's OPC Server. The OPC Server (User Application) is described later. It is basically an alternative way of connecting to GUI applications such as InTouch and Fix.

4.6.3 Replication

DATRAN Replication provides the facility for separate base stations on the same LAN to share I/O. An example of this might be a system where the water and waste pump stations are being controlled by separate base stations. There may be a third base station on the LAN that the operators use where they want to see I/O from both the Water and the Waste systems. DATRAN replication can be used to achieve this.

4.7 SMS Direct

SMS Direct a service that provides the facility for the base station to directly utilise an SMS modem connected to one of its COM ports to send SMS messages to users.

4.7.1 SMS Direct Standard

SMS Direct Standard is provided free of charge for customers that use the DATASHED SMS gateway as their primary means of sending SMS alarms to their operators. It is intended to be used as a backup and messages sent via the cellular modem can be incorporated near the end of any existing alarm escalation sequence. The standard version of SMS direct is outgoing only. Alarms cannot be acknowledged.

4.7.2 SMS Direct Full

SMS Direct Full is an optional DATRAN service that operates in the same way as SMS Direct Standard, but it also allows incoming messages from the operators to be used by DATRAN to acknowledge alarms, or to query values, manipulate I/O etc.

4.8 QTech Comms

The QRTU/DATRAN Communications is a Windows service program for the transfer of information to and from QRTU, **eXcel**, DATRAN and DATRAN II remote telemetry units.

4.9 DATRAN Derived Value Generator (DVG)

This service is responsible for the entire house keeping of nodes and the running of scripts. It also deals with setting up traps, polling and the like within scripts and initialises nodes etc on start up. It basically acts as the calculation block that is used to display data to the DATRAN Browser Program for the user to use and gain feedback on what is happening in the SCADA system

4.10 DATRAN ADO Database Connectivity Module (MSDE & MS SQL Server)

This service is used by DATRAN to interface with the Microsoft ActiveX Database Object (ADO) communications interface, so that the system can connect to MS Access, MSDE and MS SQL Server databases.

4.11 DATRAN InSQL Database Connectivity Module (Wonderware IndustrialSQL/InSQL Server)

This service is used by DATRAN VI to interface with the Wonderware IndustrialSQL/InSQL Servers, so that the system can connect to Wonderware InSQL databases. It will allow DATRAN VI to log data whether through `_InSQL_Log_Analogue_ / _InSQL_Log_Digital_` nodes and/or through the DATRAN Historical Data Interface service.

4.12 TCP Server

This service provides efficient access to all of the nodes below *Data Root* in the TBD file. It is used by DATRANVue (GUI Package) and DATRAN Supervisor (alarm and system management). These are described later in detail.

4.13 DATRAN RTU Console

This service is used for the connection of LCD keypad units, and the Q50 interface (Keri security units). This service deals with all the construction and processing of the information sent to these units via QTech Comms (Datagrams).

4.14 DATRAN Historical Data Interface

This service deals with the set-up, retrieval, and logging to the database of remote RTU Data logging functionality.

4.15 DATRAN DDE Interfaces (Client/Server)

4.15.1 Background DDE information

The format of a DDE I/O reference follows the following syntax:

`<Server>|<Topic>!<Item>`

The Server is the name of the DDE program you are connecting to, i.e. DATRAN for DATRAN VI, VIEW for InTouch.

The Topic is the name given to the entry point into this program. I.e. in DATRAN it is the name of the `_DDE_Server_Topic_`, and Data (Tag database name).

The Item name is the path from the topic. i.e. in DATRAN 'Pump1\Fault' and in InTouch / iFIX, the tag name.

E.g.

DATRAN|Parklands!Pump1\Fault
View|Data!Pump1_Fault.ACK

The main point to note about DDE is that a program can be a *client and a server*. A DDE client asks information of a server, so for instance iFIX / InTouch is usually used

within the DATRAN VI system as a Graphical User Interface and thus asks for information about the values of various tags. In this case DATRAN is the Server and iFIX / InTouch the client as it is asking information of DATRAN VI.

In addition to this DATRAN sometimes needs to be a client as well. This is typically when customers wish to use the InTouch / iFIX Alarm display and acknowledgement tools, but use the DATRAN VI paging system. This means that DATRAN has to ask InTouch when an alarm has been acknowledged via the GUI so it can halt the escalation sequence. The "Ack" field is a sub-property of the tag itself, and so this is the only way DATRAN can access the information. There are many other complexities with this system but for these purposes you can see that DATRAN is now the Client asking for information of iFIX/InTouch as a Server. Both DDE Client AND DDE Server may be used at the same time.

4.15.2 DDE Server Interface

This service deals with the aspects of being a DDE server

4.15.3 DDE Client Interface

This service deals with the aspects of being a DDE client

4.16 DATRAN Interfaces

4.16.1 Suitelinks

This service allows the DATRAN base station to communicate with Wonderware products using the Suitelinks protocol.

4.16.2 DNP3

This service deals with construction and processing of DNP3 packets.

4.16.3 OPC Interfaces (Client / Server)

These components provide the opportunity for OPC clients to connect to the DATRAN system. These include but are not limited to HMIs like Wonderware's InTouch and Intellution's Fix32 or iFIX. DATRAN OPC Client is a DATRAN Service, where the OPC Server is an application that communicates with DATRAN via the named pipe server in DATRAN XML Data Interface.

4.17 Controlling DATRAN Services

A number of console commands are available to install, uninstall, stop and start DATRAN services:

- Using the Server\Start or Sever\Stop menu option from inside DATRAN Browser
 - Using Net Start and Net Stop.

- Using Setup Install, Setup Uninstall, Setup Stop and Setup Start functionality.

Using the menu options from inside DATRAN Browser is the quickest and easiest way to start and stop the DATRAN services. Be aware, however, that each of the DATRAN Services has its own login and password to authenticate under the windows environment. If you are not logged in as the same user that the services are logging on as, then you will not be able to see any error messages that may occur when the services start up. Keep this in mind if things don't seem to be going according to plan.

Alternatively you can use the Net Start/Stop commands. These are written in the format 'Net Start *Service Exe Name*'. This will Start/Stop/Restart the service. These are usually used from within the DATRAN Browser application in scripts. (The Mission Critical for instance uses the Net Start/Stop commands to automatically restart services that have stopped for one reason or other).

Another set of commands that is usually used from the command prompt is the 'Setup' command sequence. These are used in the format of '*Service Exe Name* Setup Start'. The commands that can be used are start stop install and uninstall. These commands need to be run from the directory in which the service '.exe' file resides.

5.0 APPLICATIONS – USER INTERFACES

A number of Windows Service programs communicate with RTUs, send pager messages, display alarms, or send to Operators' cell phones (SMS). The Windows Service programs carry out all tasks that need to happen automatically. The most common DATRAN Services in a typical DATRAN System are shown below.

The user may start one or more user interface programs. These are user applications and may be started or stopped at any time without interrupting automatic operation of the system. One such application is DATRAN Supervisor (shown below).

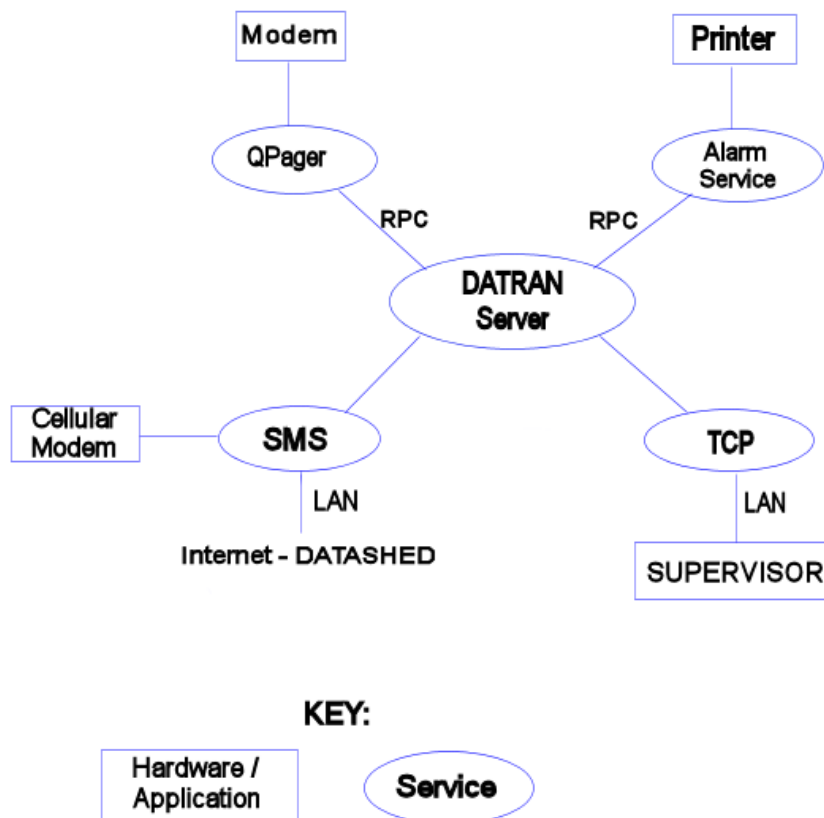


Figure 2: Simple DATRAN Server Communication via RPC

The services communicate with the DATRAN Server using a protocol called RPC (Remote Procedure Call). RPC can be used over networks. This allows DATRAN programs to run on networked PCs on a LAN. The TCP Service has replaced RPC for new Applications such as DATRAN Supervisor and DATRANVue

5.1 DATRAN Applications & Services

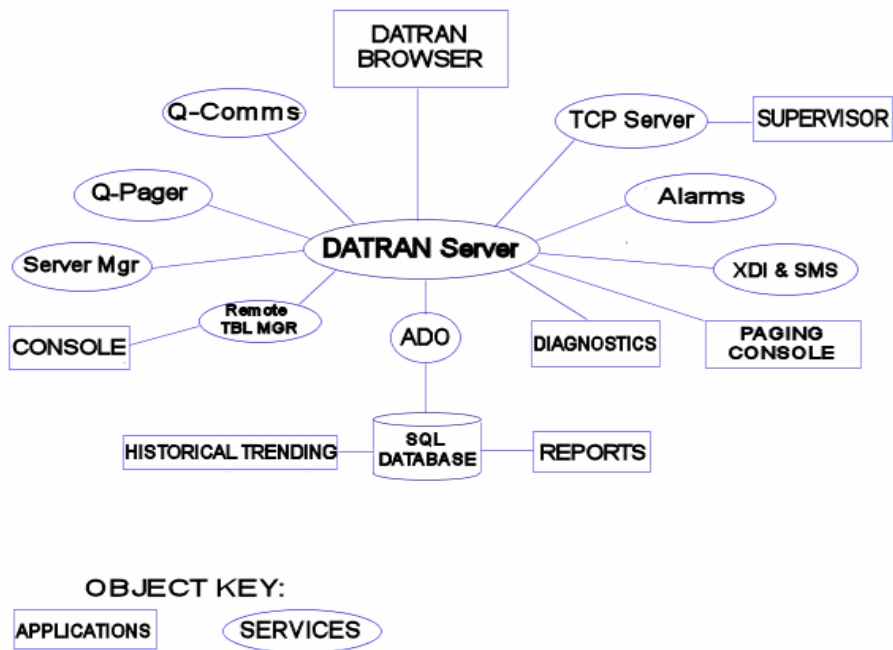


Figure 3: Relationship between DATRAN Services & Applications.

Note: Figure 3 above shows both legacy and new Applications Services. E.g. DATRAN Supervisor and Console have both been shown, even though Console has been superseded by Supervisor.

Note: Not all Applications / Services have been shown in the diagram. E.g. DDE Server / Client, Historical Data Interface (HDI), Connectivity Modules (DF1, DNP3, Modbus), and DATRANVue. These are covered in other sections.

5.1.1 DATRAN Applications

- ❑ DATRAN Supervisor
- ❑ DATRAN Browser
- ❑ DATRAN Paging Console
- ❑ DATRAN Diagnostics
- ❑ DATRAN Trending
- ❑ DATRAN Reporting
- ❑ DATRANVue (Not shown in Fig. 3)
- ❑ DATRAN DLP IDE

For each DATRAN User Interface application the following will be described:

- ❑ The general functionality
- ❑ The function of menus, buttons and other controls
- ❑ Relevant configuration in DATRAN Browser
- ❑ Other possible applications

5.1.2 Paging Console

This was historically used to manually page cell phones and pagers.

DATRAN Supervisor now has this capability built in, and Paging Console is no longer supported.

5.1.3 DATRAN Supervisor

DATRAN Supervisor has replaced DATRAN Console, as the Alarm & Event handling software package. Console was supported in DATRAN versions previous to V6.60.

Supervisor is a powerful and highly configurable program having the following features:

- ❑ Shows the user a list of the current alarms and a history of the most recent alarms and acknowledgements etc.
- ❑ Shows a list of the latest events (e.g. pump runs).
- ❑ Can show a list of all the sites on the system and a user friendly way of force polling and changing certain outputs etc.
- ❑ Displays the detail of the state of all of the I/O at a given site

- ❑ Shows comms channel statistics and general system health
- ❑ Allows system managers to change access levels and assign which operators are to be sent alarms and etc.
- ❑ Mini-trends allow the user to view the recent history of a specific I/O point.
- ❑ Alarm Notes allow users to enter comments on specific alarm events for future reference.
- ❑ Has the look & feel of a Windows Application (with scalable screens etc)
- ❑ User friendly with Online Help
- ❑ Improved Security & Integrity of Data
- ❑ Requires User Log-On Authentication
- ❑ Multiple DATRAN Servers can be accessed
- ❑ Users can edit Analogue Set Points
- ❑ Users can customize clickable buttons
- ❑ Network deployment is easy over LAN

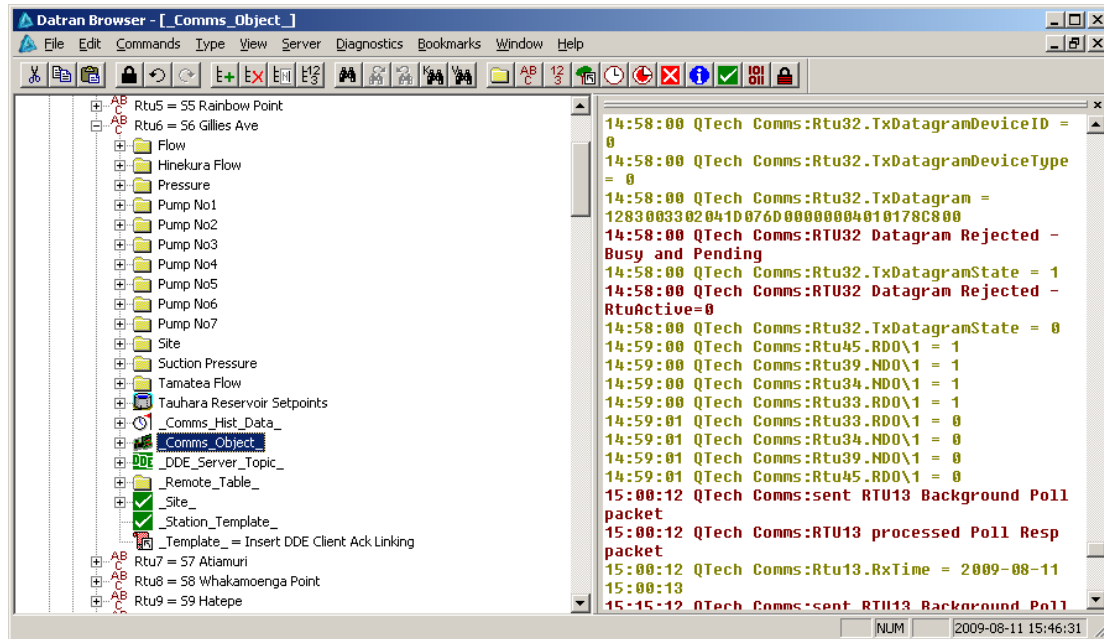
Note: The DATRAN Supervisor features and functions listed here are only giving a brief overview. Supervisor is capable of much more functionality, and this is detailed in its design documents and promotional material. If there is something more that you would like to see supervisor able to do, we welcome your feedback.

5.1.4 DATRAN Browser

This is the main configuration tool in the DATRAN VI system and is used to add sites to the system and configure everything about the system and supporting programs. In versions of DATRAN 6.62 and above, DATRAN Browser now incorporates a diagnostics window to allow the user to view both the TBD file and DATRAN Diagnostics in the same application.

This diagnostics pane can be accessed via the “View ⇌Diagnostics” drop-down menu, or by pressing F12 while in DATRAN Browser.

The “Diagnostics” drop down menu contains a number of options for controlling the behaviour of this window.



5.1.5 DATRAN Diagnostics

This is a powerful debugging tool that can be used to see the state of the system and its services and what they are doing. This is invaluable in fault finding. DATRAN Diagnostics is now also available as window inside DATRAN Browser.

See the note above regarding Diagnostics in DATRAN Browser.

5.1.6 DATRAN Trending

This program is a user interface to the SCADA historical database (DATRANdatabase) and gives the user the ability to graph the information within the database, which can be invaluable in the diagnostics of events in the field. It is also used for but not limited to:

- ❑ Finding out the current state of a site
- ❑ Ensuring data is being logged into the DATRAN database
- ❑ Viewing trends of past data
- ❑ Exporting data from the DATRAN Database to a .CSV file for third party applications to use.

5.1.7 DATRAN Reporting

This program is a user interface to the DATRAN historical database (DATRANdatabase) and gives the user the ability to create reports based on information within the database and the report design template defined by the user. It also gives the user the ability to schedule the reports to be printed later, saved to disk, emailed or uploaded to a FTP server. This is ideal for monthly reports for resource consent purposes etc.

5.1.8 DATRAN DLP IDE

This programs is a user interface to edit and compile ASM files into DLP files before they are uploaded to DATRAN II eXcel RTUs. It can also be used to test the source code in the ASM files before the they are commissioned.

5.1.9 DATRANVue

This is a GUI application for designing mimic screens in the same way as InTouch or iFIX. There are 2 parts to this application: the *Designer* and the *Runtime*.

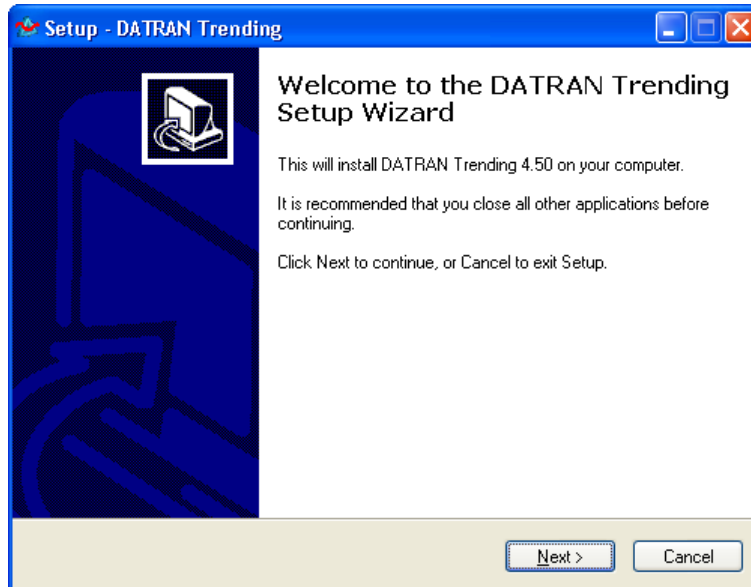
The Designer is used for creating screens using text and scalable graphics depicting meters, pumps, circuit breakers, pipe work, etc. These graphics and text controls can be linked back into DATRAN VI through the TCP Server Service to visually show the state of the DATRAN system. Mimic screen design is easy as objects are simply dragged and dropped from a library and then linked to DATRAN for their animation properties. Data links are created simply by navigating the DATRAN TBD file and clicking on the point required. This is unlike InTouch or iFIX which require a complicated and time consuming approach of creating tags and linking tags by defining their DDE address.

The Runtime is purely a means of displaying file created by the Designer. The screens are not editable in the Runtime.

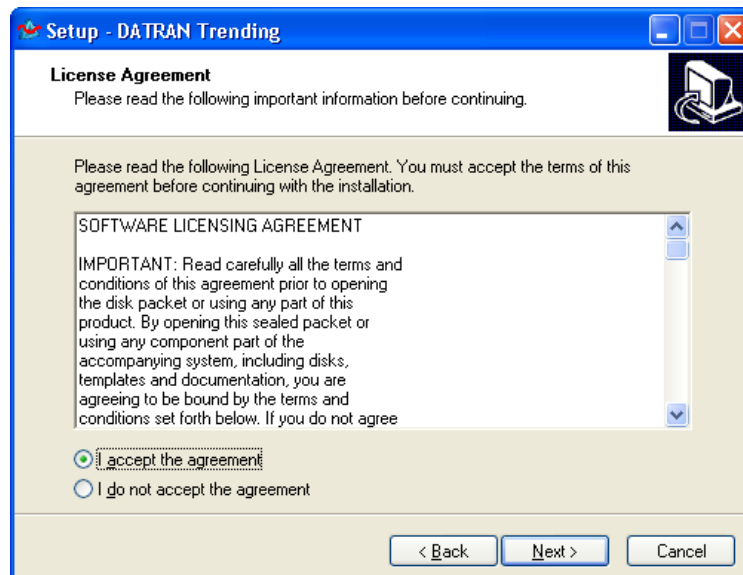
6. DATRAN Trending

6.1. Software Installation

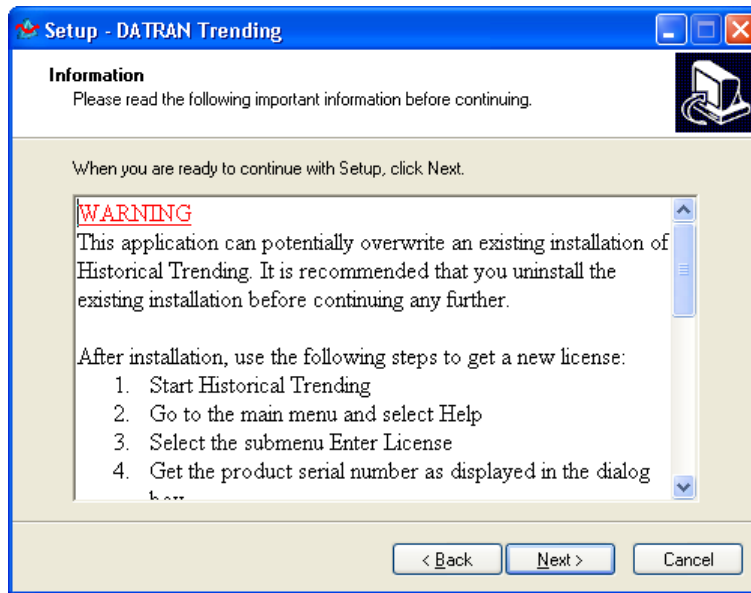
The installation archive can be found on the Trending Installation CD, or in the “Extras” folder on your DATRAN IV installation disc. To install DATRAN Trending, simply run “DATRAN_Trending_v4_50_Setup.exe”. The greetings screen will appear.



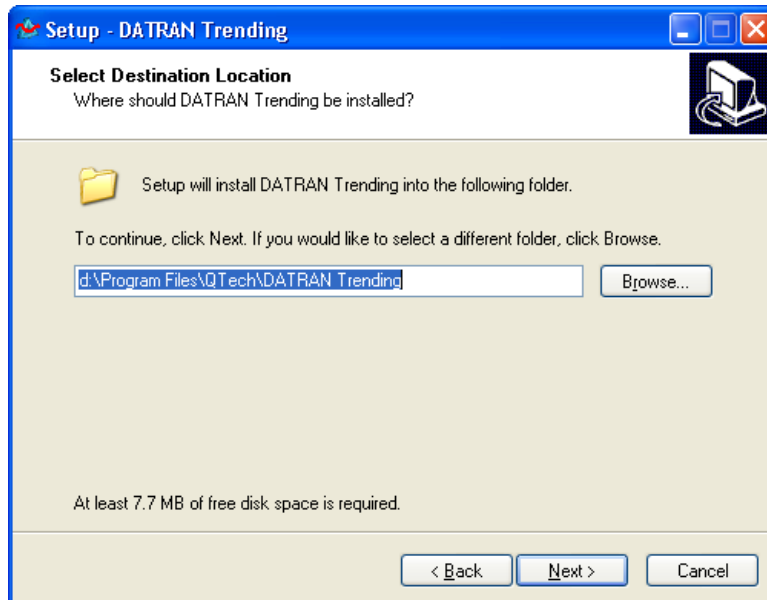
Click next to continue. The license agreement screen will appear. Click that you agree with the license, then click next to continue.



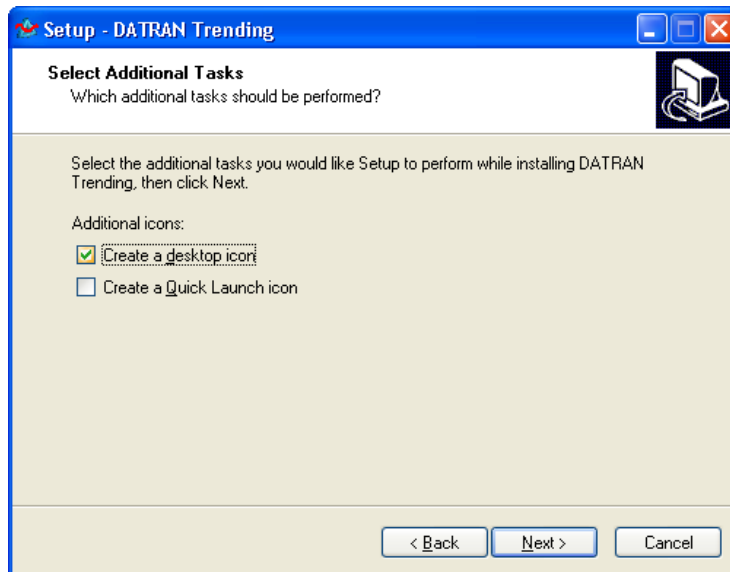
Read the information that appears on the next screen and when you understand it, click next again.



Read the information that appears on the next screen and when you understand it, click next again.



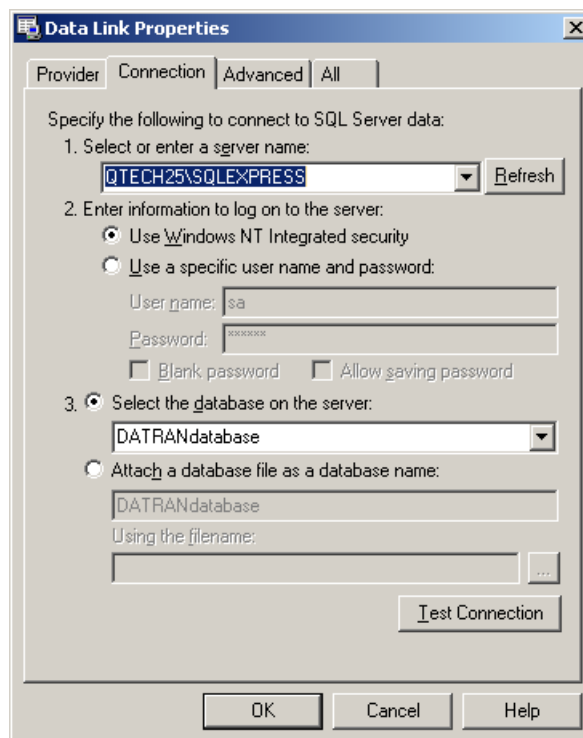
On the next two screens, select the name for the installation folder and the start menu folder. Click next again.



Select whether you want a desktop shortcut or quick launch icon created for Trending. If you are installing over an older copy that already has a desktop shortcut you should turn the desktop shortcut off or they will double-up. Click next, and then click install to install DATRAN Trending.

6.2. First Run

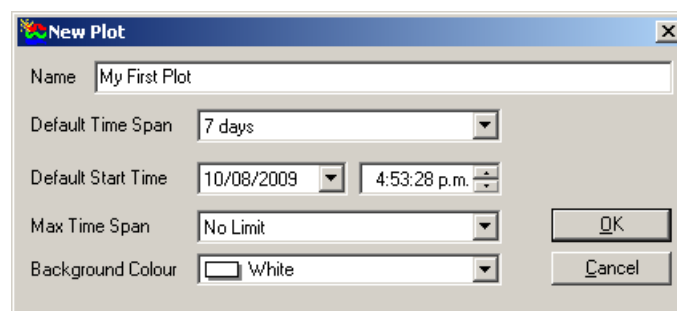
The first time DATRAN Trending is run, you will need to enter the information required so that DATRAN Trending can connect to the SQL Server that contains the DATRAN Database. This connection information can also be changed at any time from the “Database ⇨ Switch Database” drop down menu item inside Trending.



You will need to enter the SQL server name and authentication method for the SQL Server that contains the DATRAN Database. The screenshot shows the standard values for a SQL server running SQL Express 2005 on the same domain as the installation of Trending. In this case QTECH25 is the name of the machine running the SQL Server. The test connection button will provide a report on whether or not the information provided is sufficient to create a connection to the DATRAN Database. See your I.T. help desk if you require assistance.

6.3. Creating a Plot

The create new plot dialog will automatically appear the first time Trending is connected to a DATRAN Database, if no plots already exist. This dialog can also be accessed from the “File ⇒ Create New Plot” drop down menu item inside DATRAN Trending.



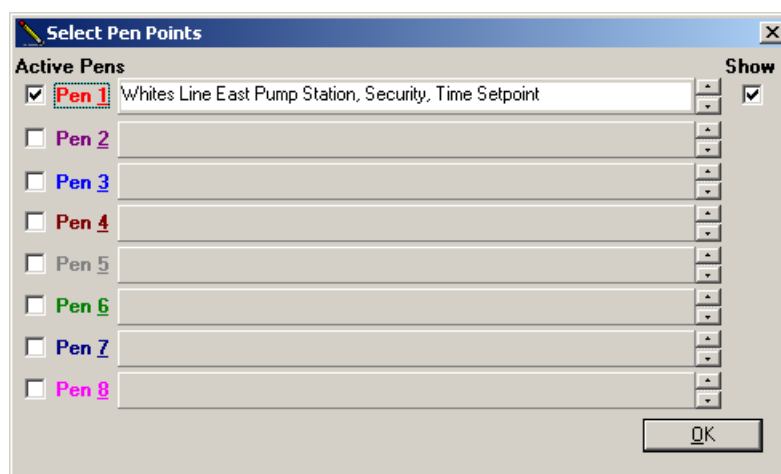
Default Time Span determines how much data is displayed on the screen each time the plot is opened.

Default Start Time sets the start time for the plot each time it is opened. This may be overridden by the global “Always Load on Today” property.

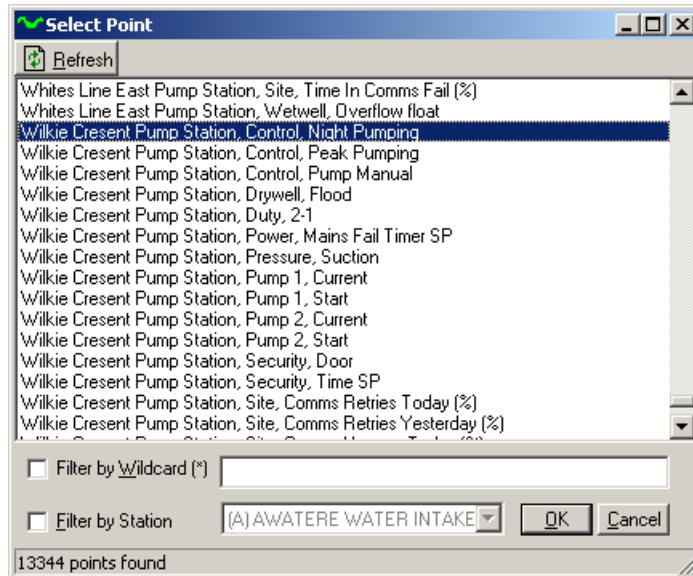
Max Time Span sets the maximum amount of data to load from the database into memory. Setting this value larger will allow the user to scroll back further in history, while setting it shorter will provide better performance.

Background Colour sets the background colour of the plot.

Once the user clicks “OK” the plot pen selection dialog will appear. This is where the user can select which pens from the database they wish to see on their trend.

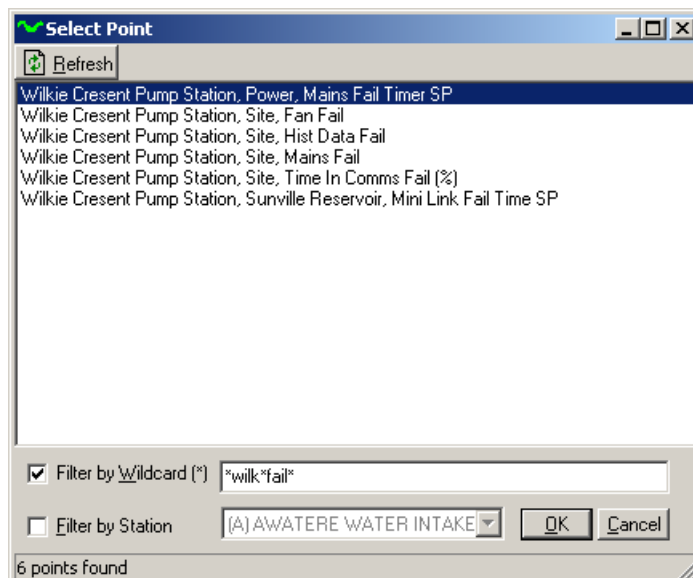


Tick the “Active Pens” box to enable each pen. Once a pen is activated, click on the name of the pen to activate the “Select Point” dialog.



The select point dialog allows the user to browse the list of pens available in the database for display on the plot. There is a pen for every point that has ever logged any data to the DATRAN Database, so be aware that some pens may not have any current data if they have been deleted or renamed at some point. There are a number of filter options available for finding the appropriate pens in the DATRAN Database, which typically contains several thousand. The user can filter by station name so that only pens for a given RTU site are visible, or the user can use a wildcard search, or a combination of both.

For example, the screenshot above shows an unfiltered list of all the pens in the DATRAN Database, 13,344 in total. If we were interested in finding the mains fail point for Wilkie Crescent Pump Station, the wildcard filter “*wilk*fail*” could be applied to shorten the list down.

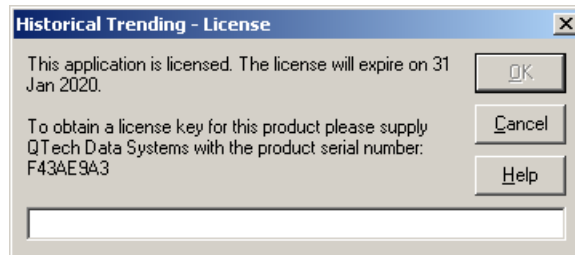


It is best to keep the wildcard search reasonably vague as sometimes the descriptions of the pens are not exactly what the user might expect. Consider if the user believed that the mains fail point above was actually called “Power Fail”. If the wildcard “*wilk*Power Fail*” had been applied, the correct point would not have been shown.

6.4. Licensing

DATRAN Trending is a licensed application and as such, a valid License Key is required to use the software. The key is tied to the machine that Trending is installed on.

To obtain a License Key, the user will have to supply a serial number to QTech from which the key will be generated. To get this serial number, go to “Help ⇒ Enter License”. The serial number in the screenshot below is F43AE9A3.



Once the License Key is supplied, simply enter it in the text box and click OK. The license is saved into the computer registry when you close DATRAN Trending, so it is advisable to do so immediately after entering the license, and then to open DATRAN Trending again to verify the license has been applied correctly.

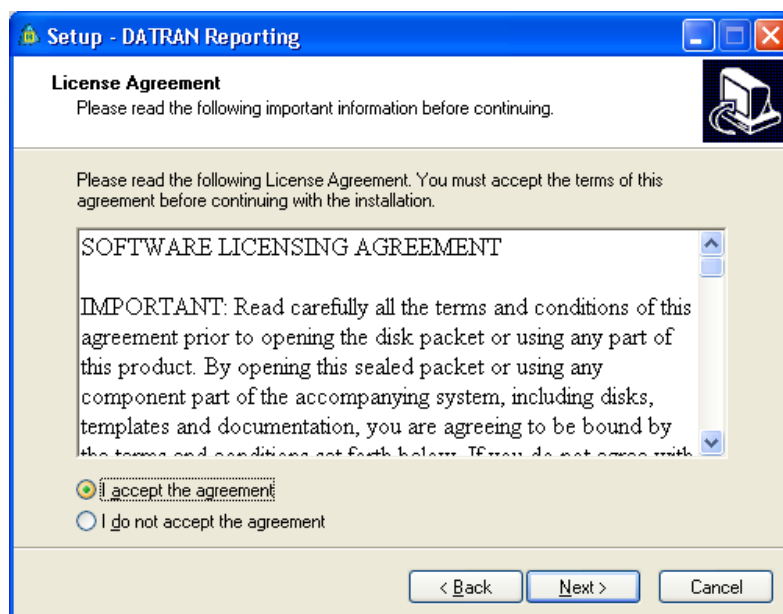
7. DATRAN Reporting

7.1. Software Installation

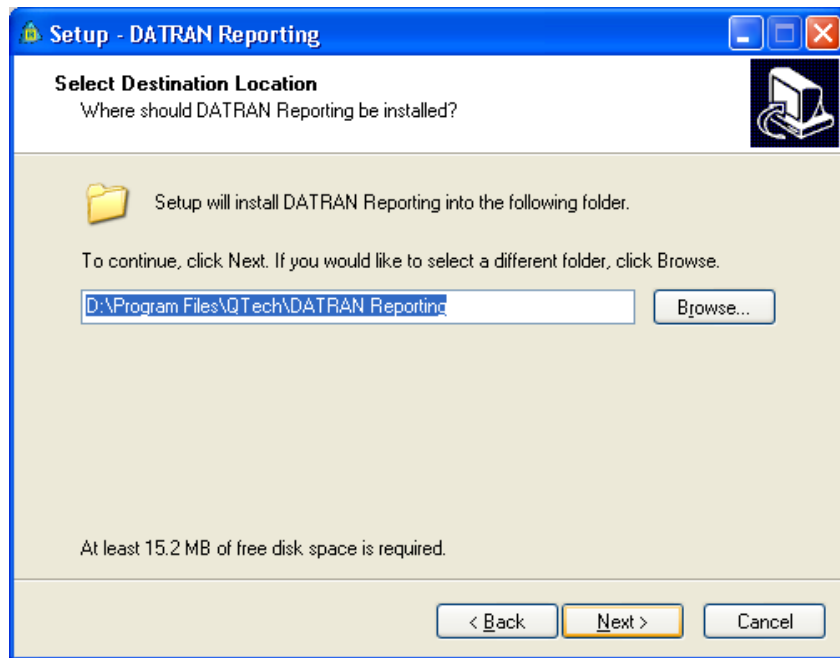
The installation archive can be found on the Reporting Installation CD, or in the “Extras” folder on your DATRAN IV installation disc. To install DATRAN Reporting, simply run “DATRAN_Reporting_v3_10_R01_Setup.exe”. The greetings screen will appear.



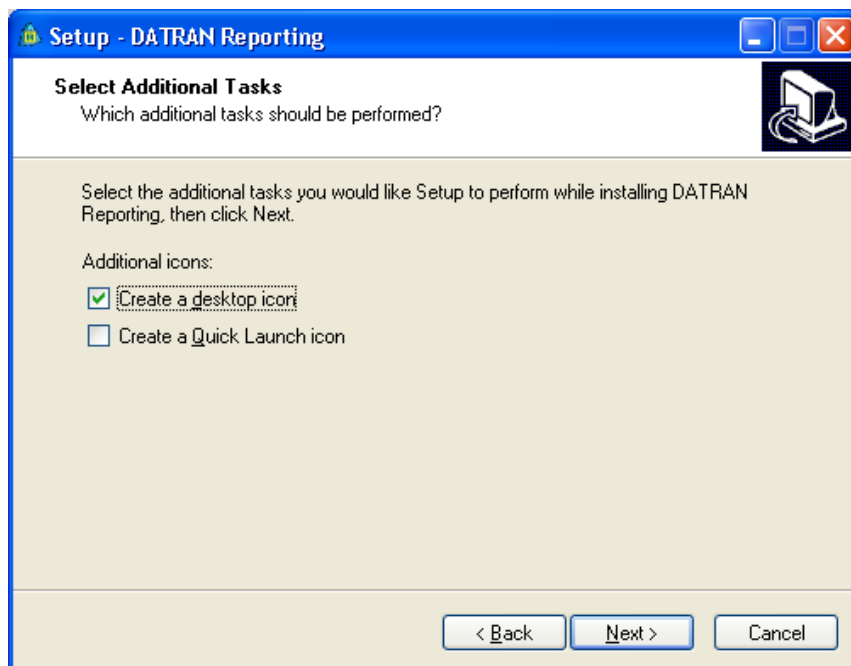
Click next to continue. The license agreement screen will appear. Click that you agree with the license, then click next to continue.



Read the information that appears on the next screen and when you understand it, click next again.



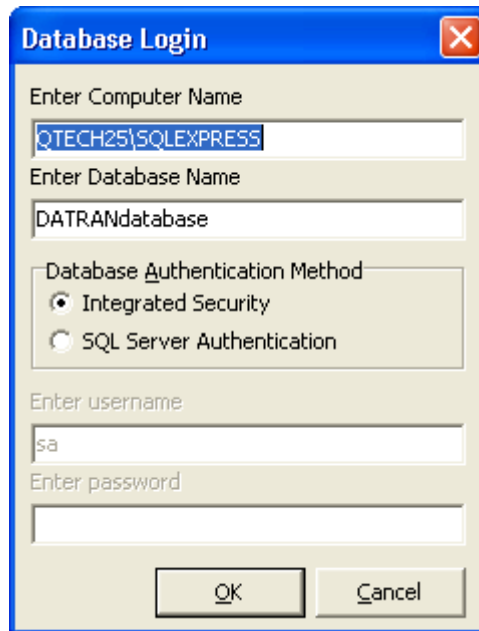
On the next two screens, select the name for the installation folder and the start menu folder. Click next again.



Select whether you want a desktop shortcut or quick launch icon created for DATRAN Reporting. Click next, and then click install to install DATRAN Reporting.

7.2. First Run

The first time DATRAN Reporting is run, you will need to enter the information required so that DATRAN Reporting can connect to the SQL Server that contains the DATRAN database. This connection information can also be changed at any time DATRAN Reporting is started up.

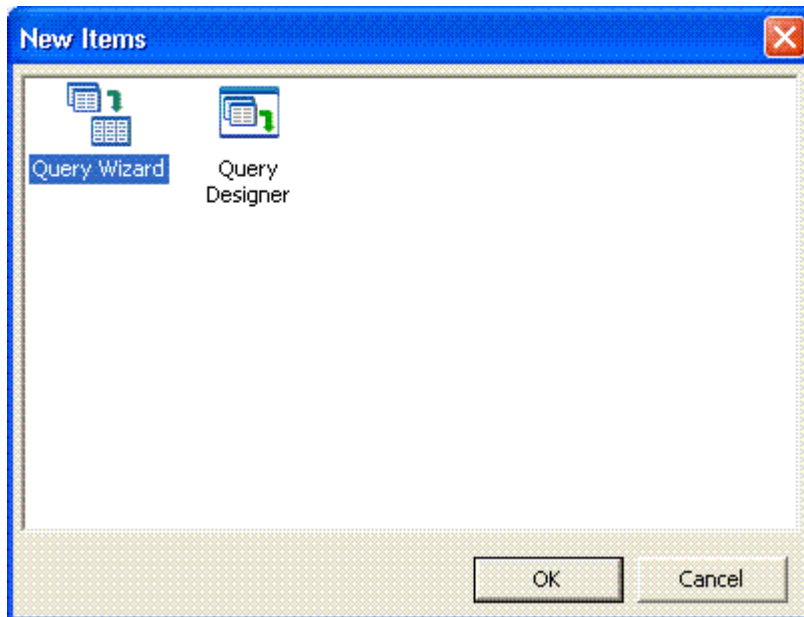


You will need to enter the SQL server name and authentication method for the SQL Server that contains the DATRAN database. The screenshot shows the standard values for a SQL server running SQL Server 2005 Express edition on the same domain as the installation of DATRAN Reporting. In this case QTECH25 is the name of the machine running the SQL Server. See your I.T. help desk if you require assistance.

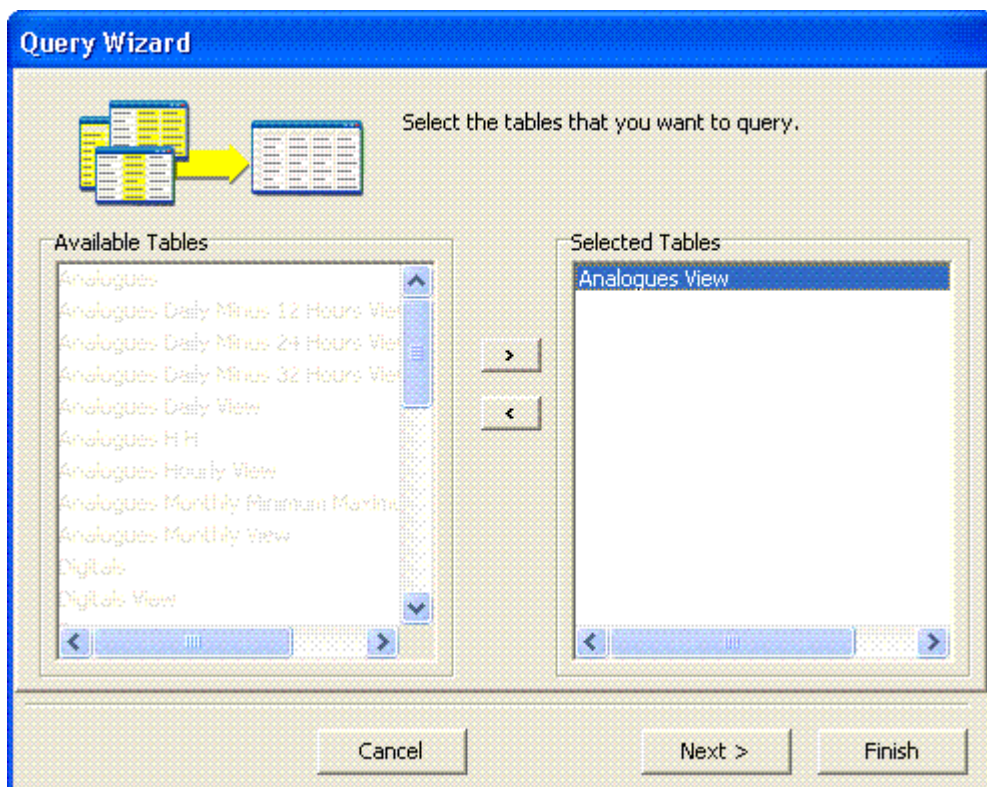
7.3. Creating a Report

To create a new report using Datran Reporting you need to perform the following steps:

1. Click on the main menu and select File
2. Select New
3. In the ReportBuilder Enterprise: New Report window, click on Data tab
4. Click on the File menu and select New
5. Choose the Query Wizard and click the OK button



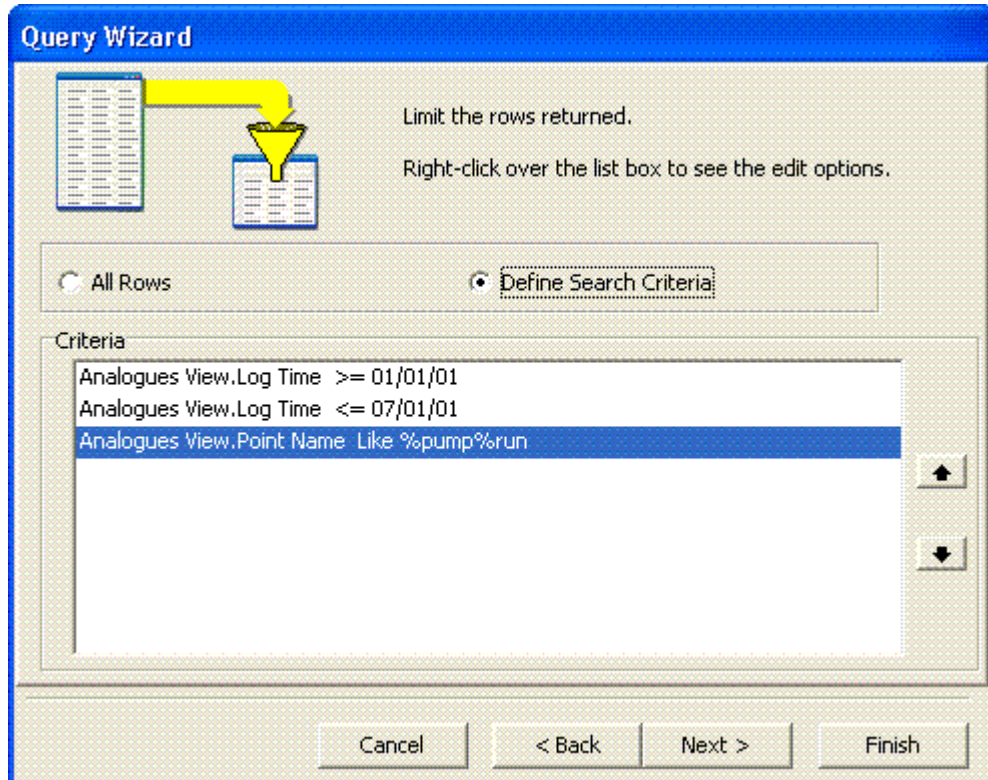
6. Double click on Analogues View to choose that (virtual) table
7. Click on the Next > button



8. Ensure that the selection is on All Fields. This will determine that all the fields in the tables are chosen
9. Click on the Next > button
10. Ensure that the selection is on No Calculations. We will skip this for this simple example.

11. Click on the Next > button
12. Ensure that the selection is on No Grouping. We will skip this for this simple example.
13. Click on the Next > button
14. Click on the Define Search Criteria radio button
15. Right click on the Criteria list box and select Insert Criteria
16. In the Field combo box, select Analogues View.Point Name
17. In the Operator combo box, select Like
18. In the Value text box, type in "%pump%run%". This will include any points that have the word 'pump' followed by 'run' in it regardless of case.
19. Click on the OK button
20. Repeat the last 5 steps for these additional criteria:
 - Analogues View.Log Time >= 01/01/01
 - Analogues View.Log Time <= 07/01/01

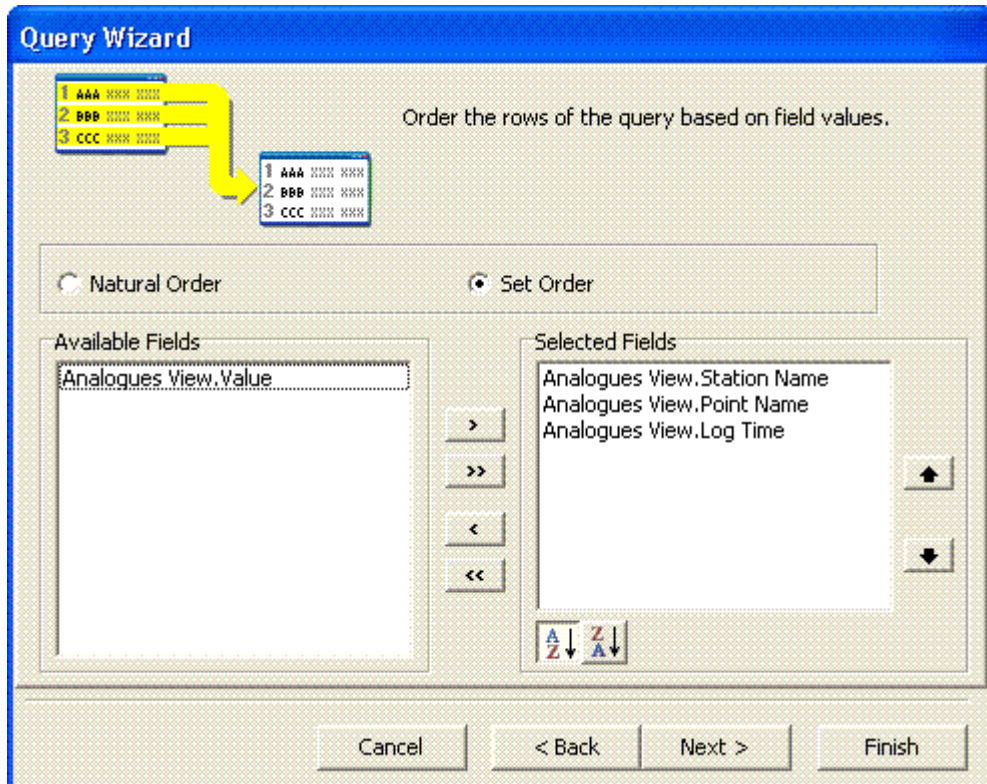
The dialog box should look something like this:



21. Click on the Next > button
22. Click on the Set Order radio button

23. Select the fields by double clicking on them in this order:

- Analogues View.Station Name
- Analogues View.Point Name
- Analogues View.Log Time



24. Click on the Finish button.

25. Notice that the Analogues_View query appears within the Data tab.

26. Click on the Design tab.

27. Click on the DBText button  on the Data Components toolbar.

28. Click on the Detail section.

29. Notice that a DBText will be placed there.

30. Use the Edit toolbar

 Analogues_View Station Name to select the source of the data. For the first one, select Station Name.

31. Repeat the previous 4 steps for Point Name, Log Time and Value.

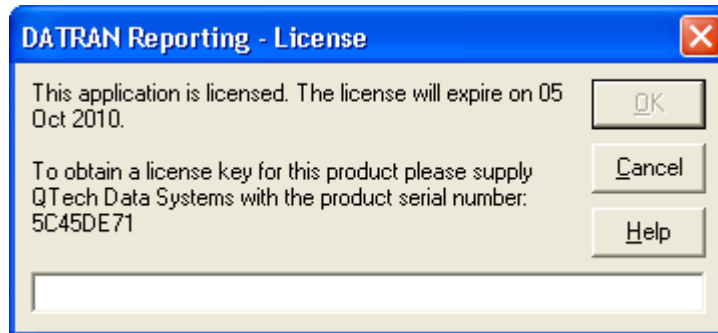
32. Click on the Preview tab to get a preview of the report.

33. Close the ReportBuilder Pro and save the report as "Pump Run".

7.4. Licensing

DATRAN Reporting is a licensed application and as such, a valid license key is required to use the software. The key is tied to the machine that DATRAN Reporting is installed on.

To obtain a license key, the user will have to supply a serial number to QTech Data Systems from which the key will be generated. To get this serial number, go to “Help ⇒Enter License”. The serial number in the screenshot below is 5C45DE71.

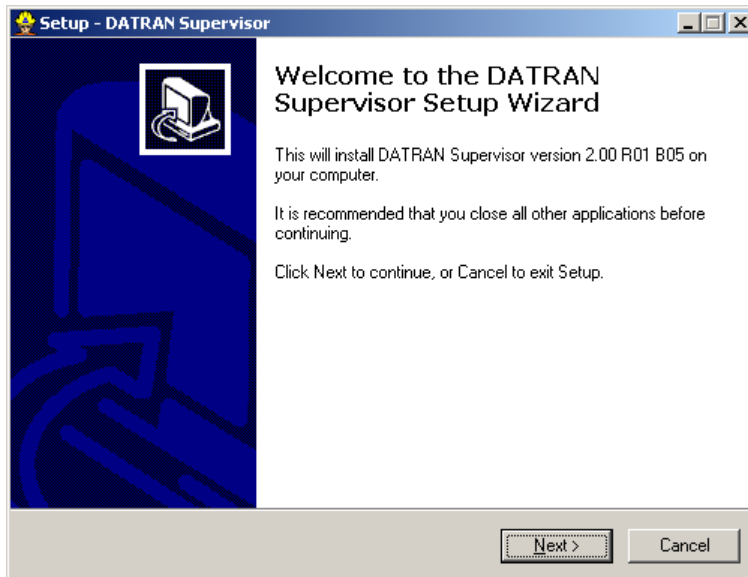


Once the license key is supplied, simply enter it in the text box and click OK. The license is saved into the computer registry when you close DATRAN Reporting, so it is advisable to do so immediately after entering the license, and then to open DATRAN Reporting again to verify the license has been applied correctly.

8. DATRAN Supervisor

8.1. Software Installation

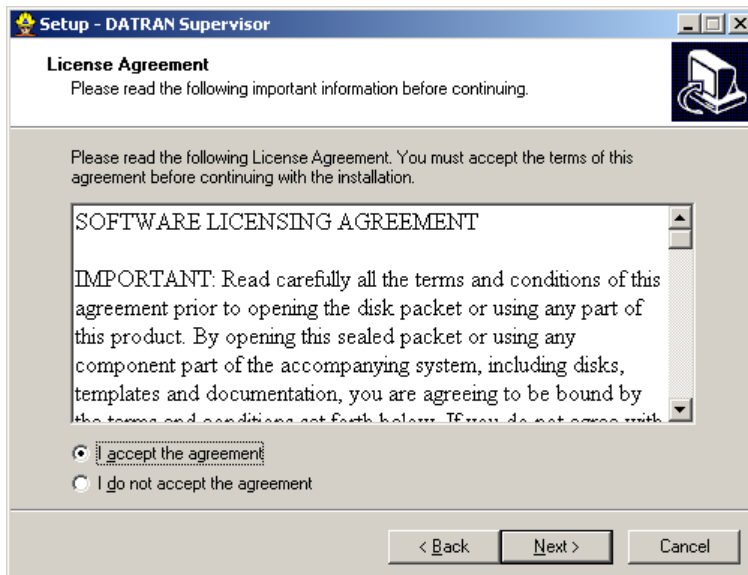
The installation archive can be found on the DATRAN Supervisor Installation CD, or in the “Extras” folder on your DATRAN IV installation disc. To install DATRAN Supervisor, simply run “SetupDATRANSupervisor.exe”. The greetings screen will appear.



This is the Welcome screen.

This tells you the product name and its version number.

Click Next.

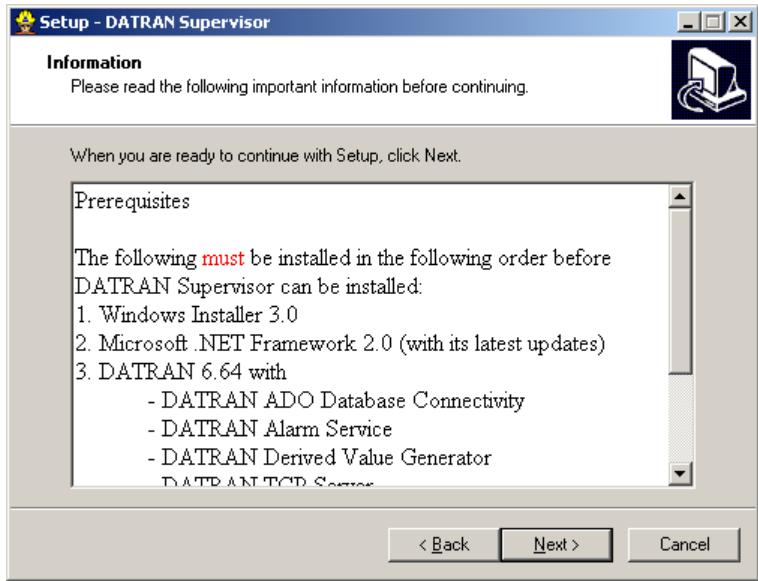


The Software License Agreement screen.

Read through the license agreement.

Click on the “I accept the agreement” radio button.

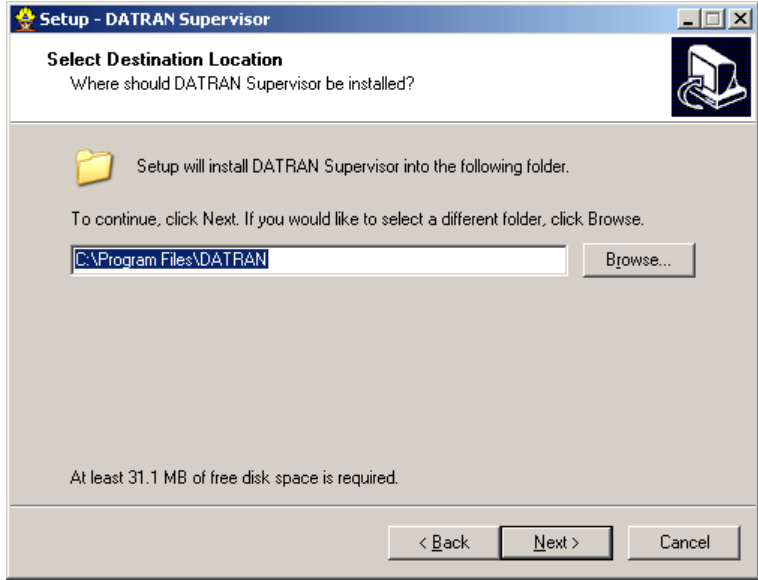
Click Next.



The Information screen.

This shows the components required to run DATRAN Supervisor.

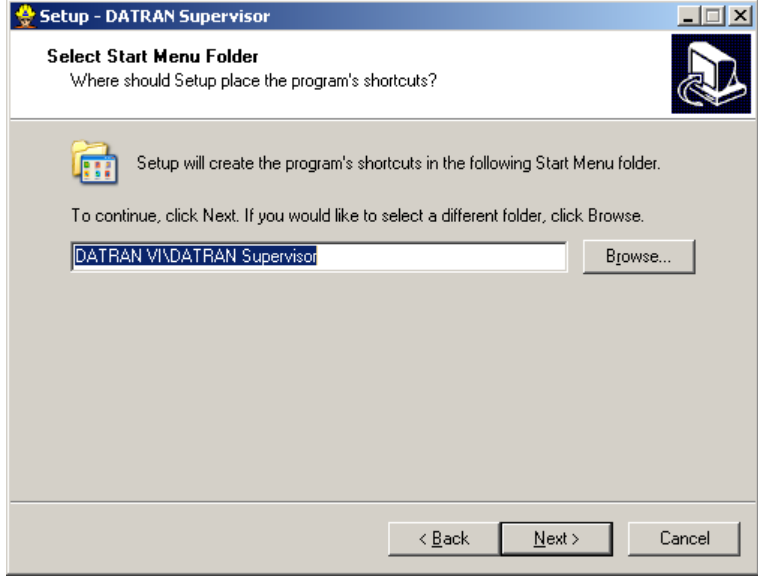
Read through them all and then Click Next.



Select installation folder.

This screen allows you to change where DATRAN Supervisor will be installed.

By default it is installed at:
C:\Program Files\DATRAN
Click Next

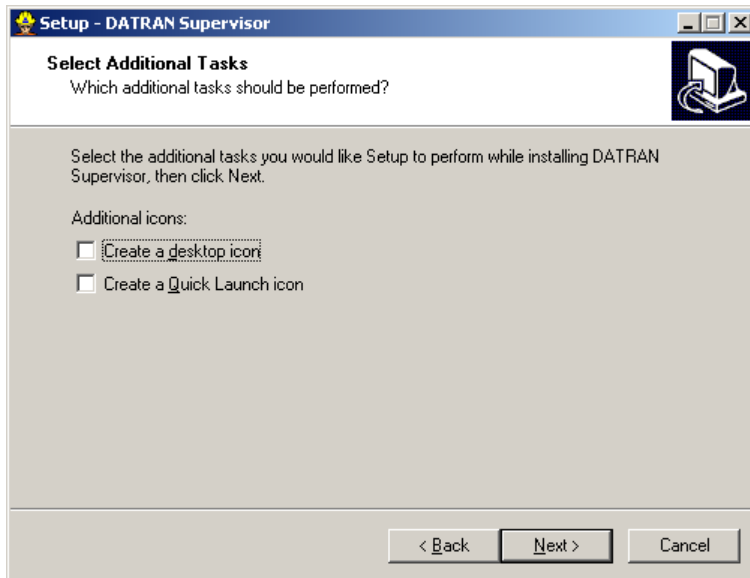


Select Start Menu Folder

This screen allows you to change where the DATRAN Supervisor icons appear in the Start menu. By default they are installed in a DATRAN subfolder called

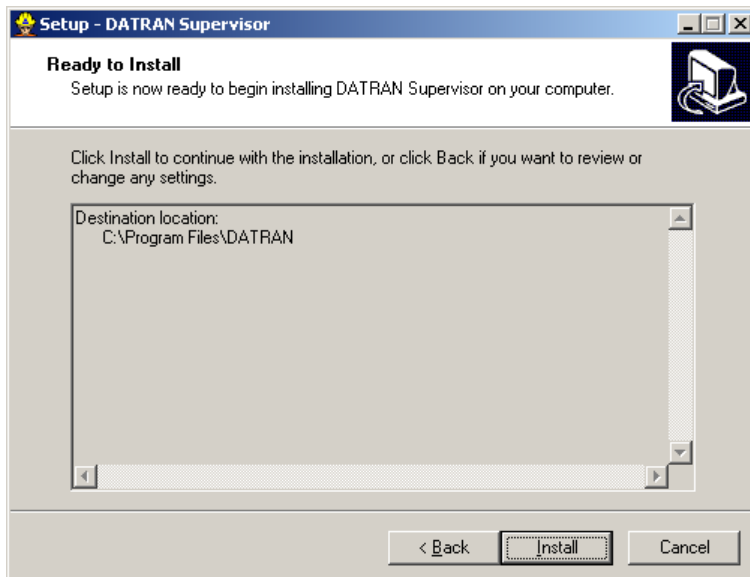
DATRAN Supervisor

Click Next



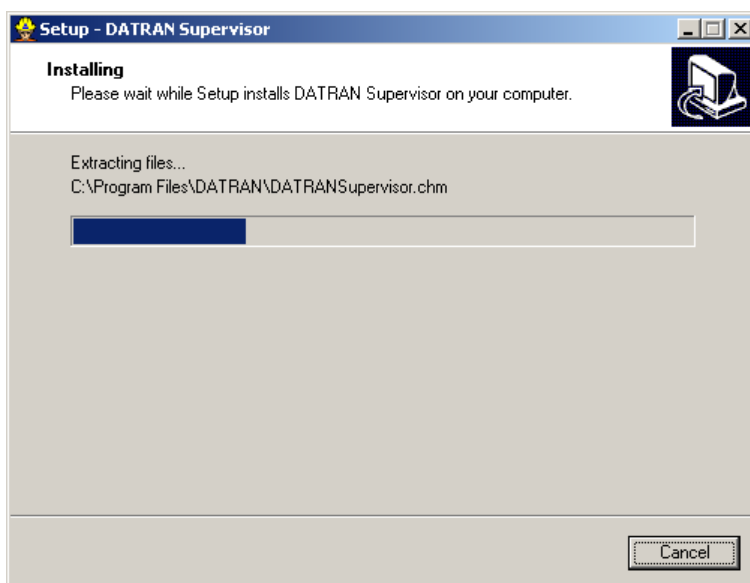
Select Additional Tasks
 You have the option of creating a desktop icon for DATRAN Supervisor.
 If you would like the installer to create one then ensure the “Create a desktop icon” checkbox is checked.
 You also have the option of creating a Quick Launch icon.

Click Next

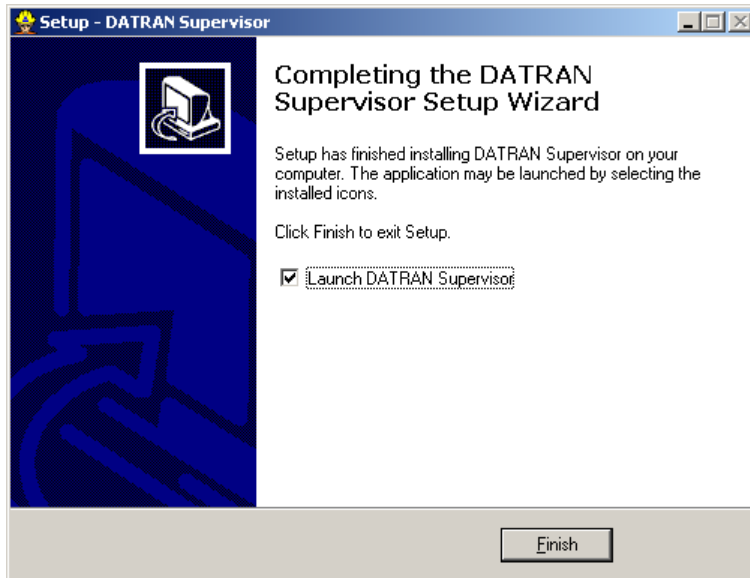


Ready to Install.
 This screen shows you a summary of the components that will be installed.
 Microsoft .NET Framework components will be shown if your computer does not have them.

Click Install.



This screen shows files being extracted to your temporary directory which will be cleaned up when the installation is completed.
 If the .NET Framework components did not install properly, you might have to obtain them from Microsoft via Windows Update.
 .NET takes several minutes to install.



Complete.

Once the installation is complete this screen is displayed it gives you the choice of running DATRAN Supervisor straight away.

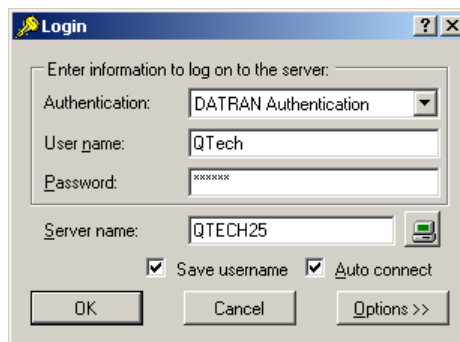
Click Finish.

8.2. Licensing

License information is stored on the DATRAN Server itself and allows for a certain number of users to be connected simultaneously. This allows for greater flexibility in the way DATRAN Supervisor is deployed.

8.3. Connection to DATRAN

When DATRAN Supervisor is started, the user will be presented with a login screen.



The login screen allows the user to select the authentication type, and a Username and Password. The name of the machine on which the DATRAN server is running must be entered into the “Server Name” box.

Windows Authentication allows DATRAN Supervisor to use your windows login to access the system. For this to work, your DATRAN System Admin will need to add your WINDOWS username to the DATRAN operator list.

Once you have decided to use Windows Authentication, DATRAN Supervisor will automatically use your Windows user credentials and not prompt you with the Login dialogue box.

DATRAN Authentication is the traditional method of connecting with DATRAN Supervisor. This will allow the user to enter their username and password, as defined in the DATRAN operator list. Passwords are case sensitive.

Wonderware Authentication allows users to automatically gain access to DATRAN Supervisor if they are currently logged in to InTouch Window Viewer with an appropriate user account. As with Windows Authentication, ask your DATRAN VI system administrator to add your InTouch/ArchestrA user into the DATRAN VI system, if it is not already there. You must always run Wonderware InTouch Window Viewer first before running DATRAN Supervisor or you will be prompted to use another type of authentication.

9. INSTALLING AND UPGRADING DATRAN VI

9.1. Software Installation

DATRAN Versions 6.48 - 6.64

There have been many versions of the DATRAN VI system over the years. The release format is that of even numbers being released tested product (e.g. 6.50) and odd numbers being the testing versions which have not yet been released (e.g. 6.51). The most current released version of DATRAN is '6.64'.

There are now two types of installation packages, the '*6.46 and below*' and the '*6.48 and above*'. Both of these programs have the same basic principles of installation however they look and are driven slightly differently.

DATRAN can be installed as a 'Fresh Install', or an 'Upgrade to Existing'. For further information, please refer to V6.64+ Release Notes & 'Quick Start Guide'.

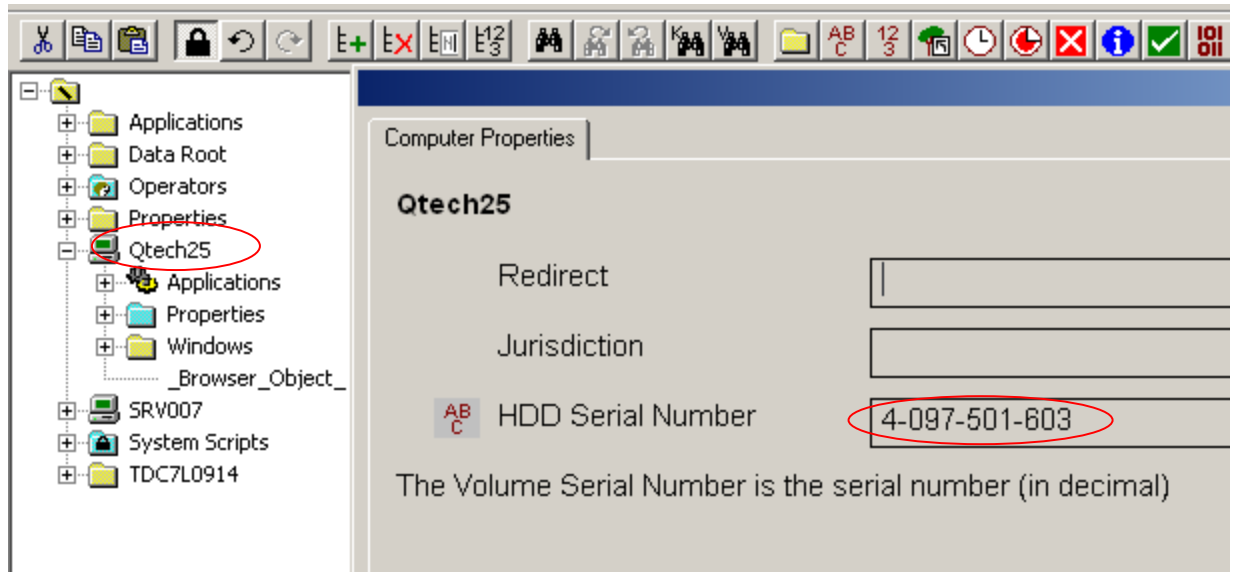
Note: There is a 'grace period' that applies to fresh installs and provides a temporary license to enable the base station to be configured. License keys must be obtained from QTech.

9.2. DATRAN Service Licensing

Most DATRAN components require individual licenses to operate. When the grace period is over these components will cease to function if an appropriate License Key has not been entered.

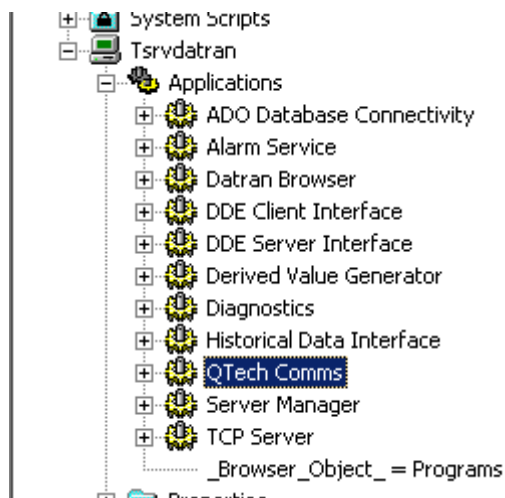
To obtain a valid licence key for each of the DATRAN Services that are running on a system, contact QTech and provide us with the HDD serial number of the DATRAN server.

This can be found in DATRAN Browser by double clicking on the machine node for the machine that requires licensing. The screenshot below shows the HDD Serial Number for a machine called QTECH25.

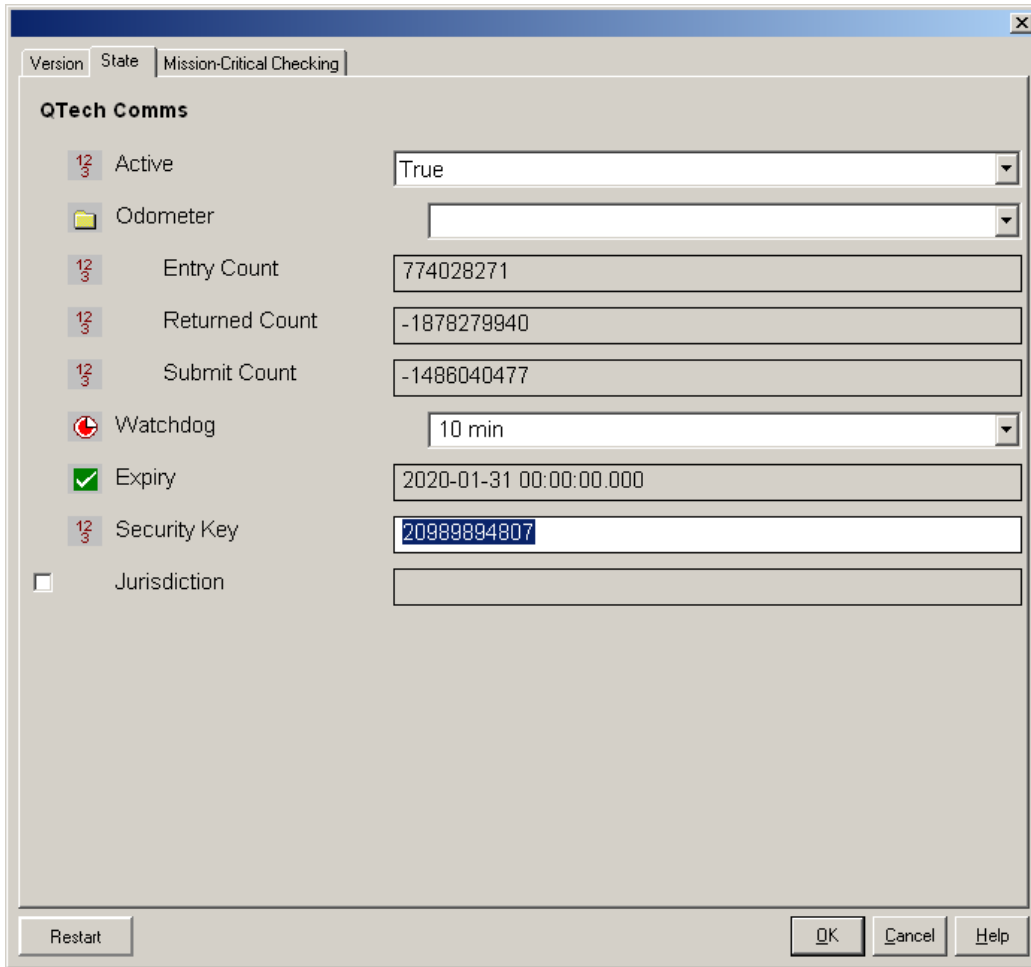


Once you have provided QTech with the HDD Serial Number, you will be provided with a number of license keys; One for each service that needs to be licensed.

To enter the licence for each service, double-click on its application node in DATRAN Browser. The example below shows the user entering a license for QTech Comms service for the machine called TSRVDatran.



Step 1: Locate the QTech Comms service application node and double click it.



Step 2: Open the “State” tab in the Application Dialog and enter the license key in the “Security Key” text box, then click “OK”.

Step 3: Move on to the next service and repeat until all keys have been entered.

There only exception to this is with the DATRAN XML Data Interface service. Because this service carries out a number of tasks, it has its own dedicated “License” tab for entering the licenses rather than entering them in the “State” tab.

Note: Some of the services will pick up the license and begin using it as soon as “OK” has been clicked, but others may require a restart before the license is applied. It is recommended that after all the applicable licenses have been entered, the system should be restarted to ensure trouble free operation.

9.3. DATRAN VI – Human Machine Interfaces

DATRAN connects to Human Machine Interface (HMI) software or Graphical User Interface (GUI) via either:

- **DDE (Dynamic Data Exchange).** This is a Microsoft protocol designed to allow communication between programs on a Windows platform. DATRAN can connect to a number of these HMI/GUI software packages but usually iFIX or InTouch.

- **OPC (Object Link Embedding {OLE} for Process Control)**. This is a well defined, 'open' standard designed for software interoperability in the automation industry. Specifically for the areas of data access. The OPC Server application and the XML Data Interface DATRAN service provide this functionality allowing GUI applications like InTouch and Fix to access DATRAN using OPC.
- **TCP Server** DATRAN service. This provides a QTech proprietary interface to the nodes within the TBD file. It is used by the DATRANVue GUI and DATRAN Supervisor and it is more efficient than the other methods in both configuration and bandwidth.

8.3.1 HMI Software & DATRAN

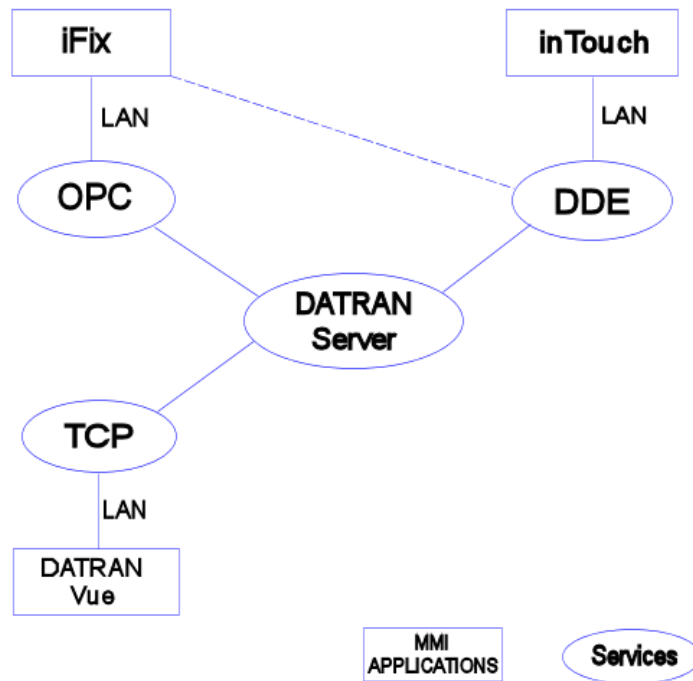


Figure 4: HMI Communication with DATRAN VI

Note: iFIX and InTouch are used here as Examples of HMI software. OPC & DDE are generic services that can interface to other HMI software such as Citect etc.

Note: iFIX can use either OPC or DDE services to connect to DATRAN, whereas InTouch can use OPC, DDE or a third party software called Suite Links which is not shown.

9.4. Client/Server Integration

Historically, the DATRAN System has been a client / server application and as such can be installed as server and can have many clients. There should only ever be one primary server, and this server runs all of the DATRAN VI services that the particular base station requires. There are usually several of the user interface programs are run on this PC as well. In addition to this several of the DATRAN VI programs can be used as clients on different PCs and can use several medium to make this connection as stated in the diagram above.

A typical application of this in the local authority market is the use of a base station at either the council buildings or the nominated contractor. A client is then installed at the site without the server running DATRAN Supervisor and DATRAN Trending so that both parties have the ability to monitor the system and use the data and the system to its maximum value.

The need for a DATRAN client installation is reducing. DATRAN Supervisor, DATRANVue and DATRAN Trending call all be installed on any machine on a LAN with out the need for DATRAN Client to be installed along side them. DATRAN Client is now only necessary when the user requires DATRAN Browser or DATRAN DDE Server to be installed on a machine that is not the DATRAN Server.

10. THE DATRAN TBD FILE

10.1. Overview

The DATRAN 'Tree Base Database' (TBD) file contains the entire configuration for the DATRAN VI system. It contains everything from the license codes for the services, to the configuration of all the sites on the system to the names and security access codes and pager IDs (numbers) of all the operators on the system. This is the most important file in the whole system and is thus backed up at several predetermined intervals and should also be backed up to another form of media on a regular basis, as without this file the system is useless.

10.2. TBD File, Filing Structure Standards & Recommended Layouts

The DATRAN VI TBD file and filing system has gone through many changes and forms over its development lifetime. A standard has emerged on how best to set up a TBD file while conforming to these standards. There are many benefits for adhering to the standards, which include:

- ❑ Makes it easier for all users to make changes to the system. The standards have been developed because they make sense and make the format of the TBD file easier to understand and easier to find components of the TBD file.
- ❑ Makes use of the Generic scripts which QTech develops. This is because you cannot economically write scripts that work on everybody's system, which means that they are written for a certain format. If the format is too different they may or may not work correctly, or even have damaging results.
- ❑ If you are the User/Administrator of several systems, you do not have to remember each sites certain layout, only one. This makes it easier, and more importantly quicker to make changes and house keep your systems. It is a good idea to spend a little more time in the short term to get a system up to standard and save over the long run.

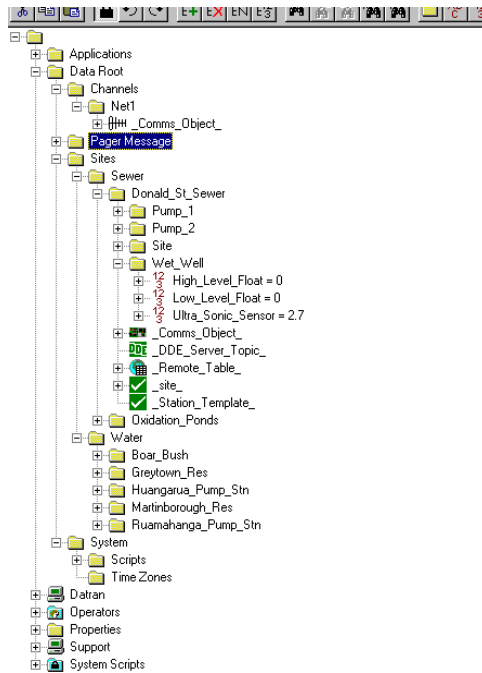


Figure 5: Tree Base Hierarchical Structure.

'Figure 5' shows the main body of the user configuration resides under the Data Root Node. In most cases all data to do with the sites and application specific information should be placed under this node. There are 3 main nodes of the Data Root node: *Channels*, *Sites* and a *System* node. These nodes should be used to store data as follow...

- ❑ **Channels** – All comms channel configuration should be placed under this node.
- ❑ **Sites** – This is where all sites and site specific data should be found.
- ❑ **System** – All custom scripts which are not found in the system scripts should be found under this node. In addition to this any other calculation type information should be stored under this node.

In addition to this there is also a correct way to set out a site. That is there should be a maximum of four tiers of information, (i.e. no more than four layers of nodes out from the 'Sites' node). In addition to this there should never be a node with a value e.g. '*Pump 1 Fault*' or '*Site Comms Fail*' directly under a Site node. It should always have at least one level of grouping,

E.g. '*Sites\Kohonunu\Comms Fail*' should read '*Sites\Kohonunu\Site\Comms Fail*'

Note that while not strictly adhering to this convention, many systems also include an extra level so that sites can be divided into categories. This is also generally acceptable.

E.g. '*Sites\Water\Delaney Park\Site\Comms Fail*'

or '*Sites\Eastern Water\Kohonunu\Site\Comms Fail*'

10.3. Computer Nodes in the TBD file and Basic Function/Use

Each *Computer Node* in the TBD file represents a computer that is connected to the DATRAN Server; this could be the server itself or a computer that has been connected to the server in the past. The computer node contains information about the volume label of the disk, a list of the services that are installed or have been installed in the past. The security and diagnostics for the services, and the default and global properties of each service will be discussed in the later sections.

For the most part all of the services will appear the same, and indeed all of the applications found under this node have a lot of things in common. However a couple of the applications have unique and very important additions to them and as such will be described in a little more detail.

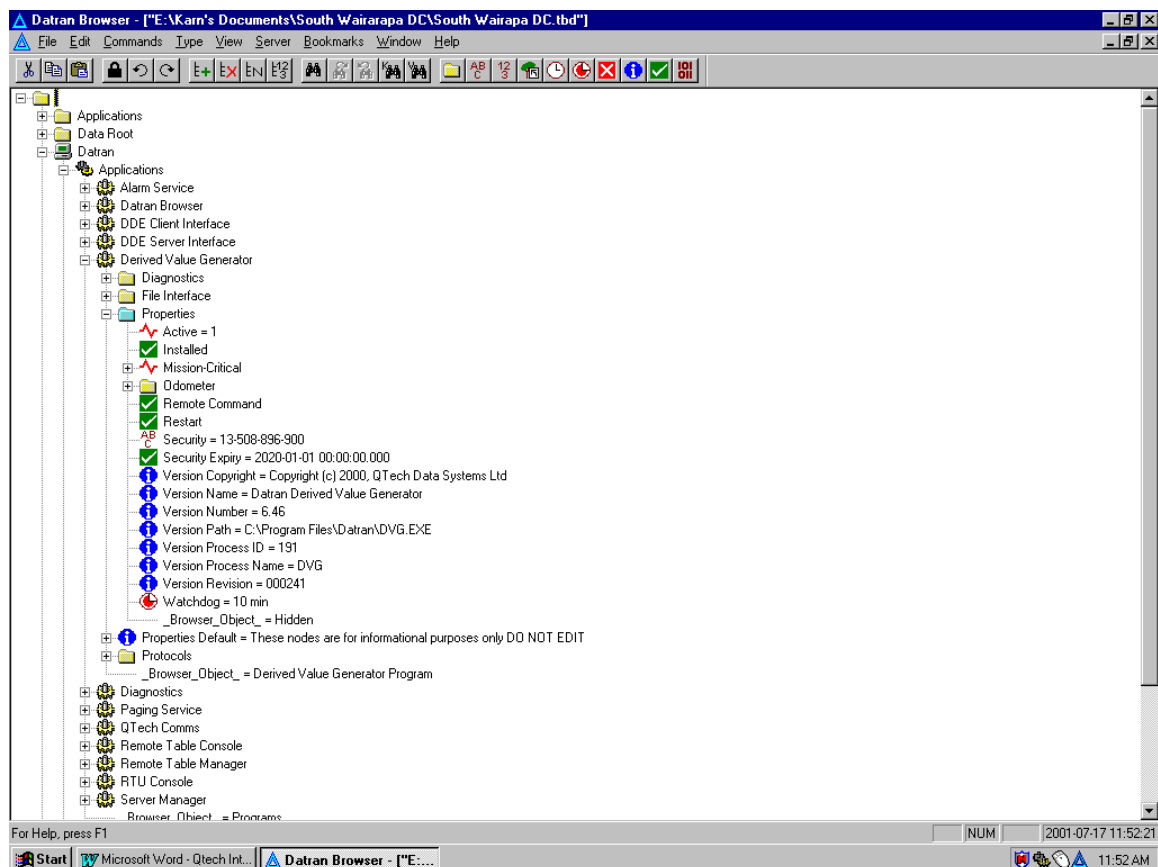


Figure 6: Application Properties - Common configuration

Common components for all applications found under this node are as follows and can be seen in 'Figure 6' above.

There are several main nodes structures under each of the applications nodes:

- ❑ **Diagnostics** – This node is used by the DATRAN Diagnostics Console application and should not be touched.
- ❑ **Properties** – This contains several of the standard features of diagnostics for this service and also where the global properties are set and held.
- ❑ **Properties Default** – This node contains all of the default properties for this service

Main features of each application's *properties* are...

- ❑ **Active** – This shows the current state of the service running/stopped.
- ❑ **Mission Critical** – This shows the state of the actual service and can be setup to call the DVG service to attempt to start it again automatically and also page out to some one that the service has had to have been restarted by the DATRAN VI engine.
- ❑ **Restart** – This node's purpose is as its name, it is there to be used to restart the service from the DATRAN Browser application. However it should be noted that it is not a full stop and start.
- ❑ **Version Number** – This shows the version number of the currently installed service/application.
- ❑ **Version Path** – This shows the path to the executable that is being used.
- ❑ **Security and Security Expiry** – These nodes are where you enter your security unlock codes and where to find when the security expires respectively.

10.4. Operator and Operator Groups Nodes in the TBD File

There is always one and only one '*Operators*' node in every TBD file. This contains the '*Operators*' and '*Operator Groups*' sub nodes and Aliases. These two nodes contain a list of all the operators that use the DATRAN VI system. It also contains such information as

- ❑ Pager/Cell phone numbers and protocol information.
- ❑ DATRAN Console/Supervisor password and access level information.
- ❑ Keri Security (Q50) access codes.

10.4.1. Operator Aliases

The use of operator Aliases allows the user to dynamically change the operator that alarms are being sent to. For example: An alarm escalation sequence can be set up to page to an Alias with the name of "Duty Operator". DATRAN Supervisor can then be used at any time to select which operator the alias "Duty Operator" applies to and the alarms will begin going to a different person without the need to edit the alarm escalation sequence.

10.4.2. Operator Groups

These contain groups of 'Operators' and grouped properties. The properties under an 'Operator Group' apply to all operators unless there is a local property, which overrides it.

10.4.3. Operator

These nodes contain the local properties for the given operator

10.5. DATRAN VI – Channels Nodes in the TBD File and Basic Function/Use

The '*Channel*' or '*Net*' nodes are connection information for the DATRAN VI '*QComs*' (QTech Communications) service to set up various com ports. Each channel represents a unique com port and its associated configuration. These ports are used by each site in the TBD file to communicate with the outside world, via DATRAN Modems and various mediums of communication.

There are two main properties in a '*Comms Channel*', the '*Mode*' and the '*Port*'. The Mode contains the *Baud Rate*, *Parity*, *Byte Size* and the number of *Stop Bits*. And the Port designates the com port that the DATRAN modem is connected to, up to a maximum of '*COM9*'.

The best way to check that the channel is open and talking to the DATRAN Modem is to apply the channel, and using diagnostics check whether the port was opened without error.

10.6. DATRAN VI – System Scripts Node in the TBD File and Basic Function/Use

This is a *read only* section of the TBD file used to store system critical scripts that without the system would not function correctly. E.g. nodes and scripts that you may find:

- Station Template
- Database Logging Scripts
- Pump Run Hours Calculations scripts
- Comms Fail Counters
- Paging Scripts
- DVG Scripts

CAUTION: THESE SHOULD NOT BE ALTERED!

10.7. DATRAN VI - Sites Nodes in the TBD File and Basic Function/Use

The 'Site' node in a TBD file is used to represent a site in the outside world and contains all configuration data for that specific site. A typical site will look as below

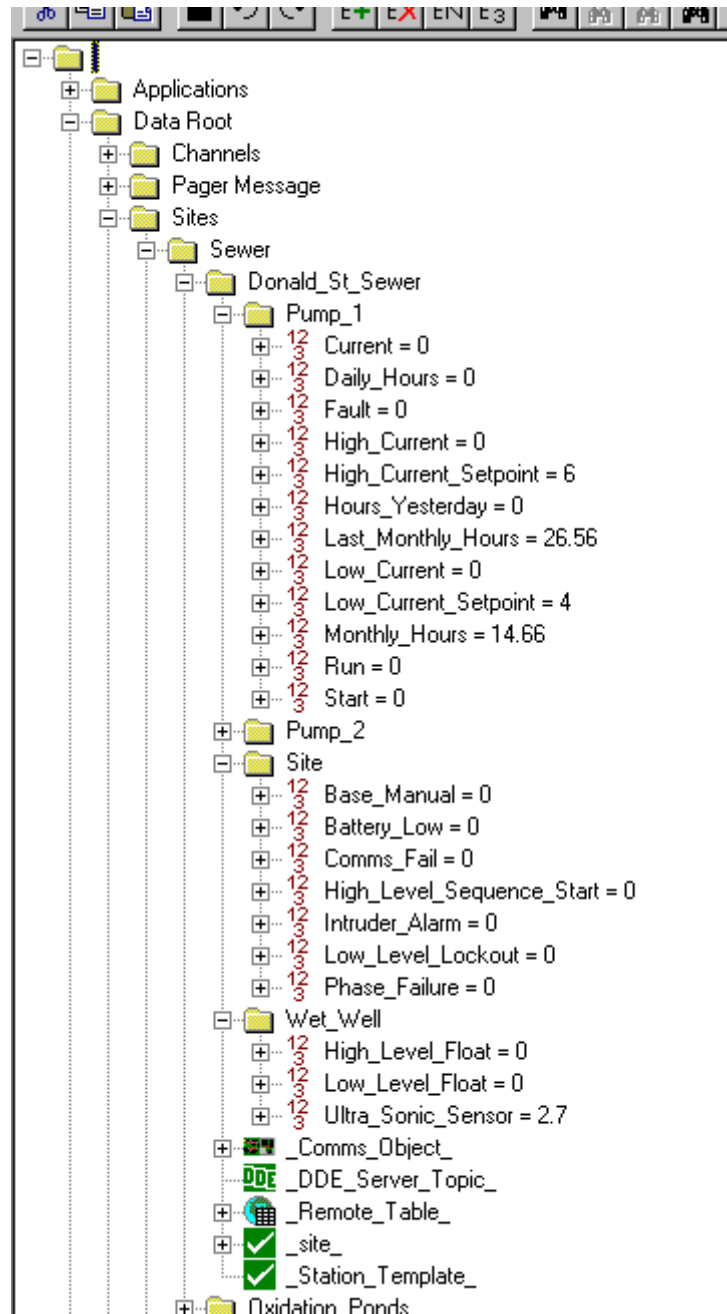


Figure 7: Typical Site Layout

Note: This site adheres to the standards previously mentioned and thus is utilizing the *Station Template* and *Site Templates*. This makes setting up a new site very simple and leaves less room for user errors (e.g. typo's and copying errors).

11. Comms_Object_nodes, Alarm_nodes, I/O, Formula and Copy_To_nodes

11.1. The Comms_Object

The Comms_Object is a special node that is present in every site in the TBD file. It contains information about the site including the address of the site, the comms channel that the site has been installed on and the DLP the site should be using. It also contains several controls for communicating with the site. The ForcePoll node, for example, causes the Base Station to immediately poll the site and retrieve its current I/O and status.

The force poll is triggered by either selecting it and pressing Alt-A, Right clicking it and choosing 'Transmit', or pressing enter on it and changing its value to a '1'.

Double-Clicking a Comms_Object node will cause a dialog box to appear with all the various properties grouped into categories to make configuration easier.

11.2. I/O

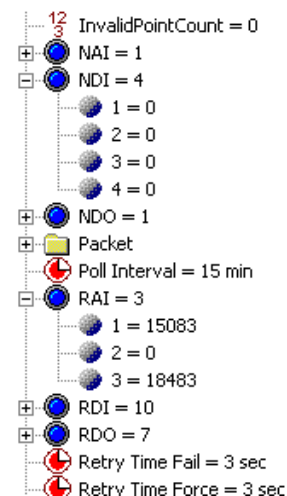
The Comms_Object is also the entry point for the I/O from each site into the DATRAN TBD file. There are 8 categories of I/O that can be passed between the base station and the RTU. These are Digital Input and Outputs and Analogue Inputs and Outputs of either "Real I/O" or "Notional I/O". An example can be seen in the following diagram:

This shows a section of the Comms_Object for a site that has 1x Notional Analogue Input, 4x Notional Digital Inputs, 1x Notional Digital Output, 3x Real Analogue Inputs, 10x Real Digital Inputs, 7x Real Digital Outputs.

The NDI's and RAI's have been expanded to show you the sub nodes that show the value of each individual I/O point the site has.

For example, the value of NDI-3 is '0' or OFF. The value of RAI-1 is 15083.

Note that the value of analogues is shown as their RAW value, which is an unsigned 16 bit number.



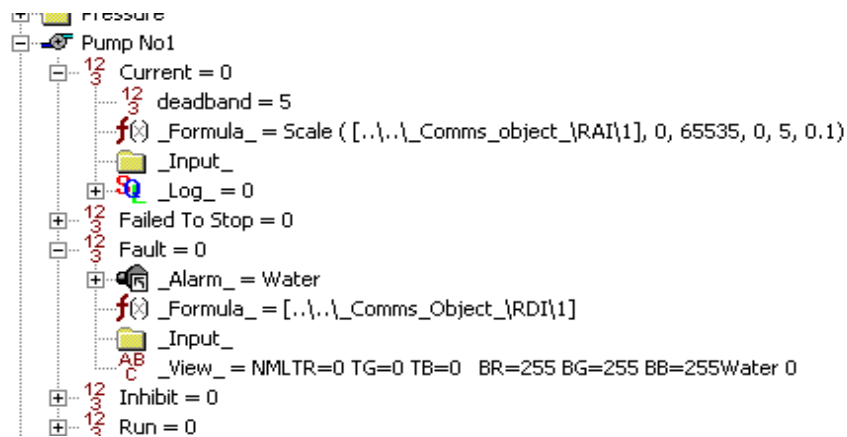
Consider when a 4-20 mA transducer is connected to an RTU that shows pressure with a range of 0-10 bar. When the pressure being measured is 10 bar, the current will be 20 mA, which will equate to a RAW value of 65535.

When the pressure is 3 bar, the current will be 8.8 mA, which equates to a RAW value of 19660.

11.3. Formula_Nodes

As discussed above, the I/O from a given site enters the TBD file as a sub node of the comms object, but this is not a very logical format for the I/O. How does a user know, for example, that RDI 1 is "Pump 1 Running" or that RAI 3 is "Wetwell Level"?

For this reason, a site in a TBD file is arranged in such a way as to publish the values from the `_Comms_Object_` to a location that describes what that I/O is associated with. The example below shows how we use a `_Formula_` node to achieve this.



The function of a `_formula_` node, in simple terms, is to evaluate an expression, and to push the result of that expression into the PARENT node of the `_formula_` node.

It can be seen in the example above that the `_formula_` under the “Pump No 1\Fault” node contains a reference to RDI\1 from the `_Comms_Object_`. In this particular case, the `_formula_` reads the value of RDI\1 and pushes the result into the Pump No 1\Fault node.

Compare that with the `_formula_` under the Pump No 1\Current node. This time, the `_formula_` reads the value from RAI\1 in the comms object but also scales it so that it shows the value in terms of the units that are appropriate. i.e. The raw value in the range 0 => 65535 gets scaled to a number in the range 0 => 5 with a rounding granularity of 0.1. This would correspond to a 0 to 5 Amp transducer being used in the field.

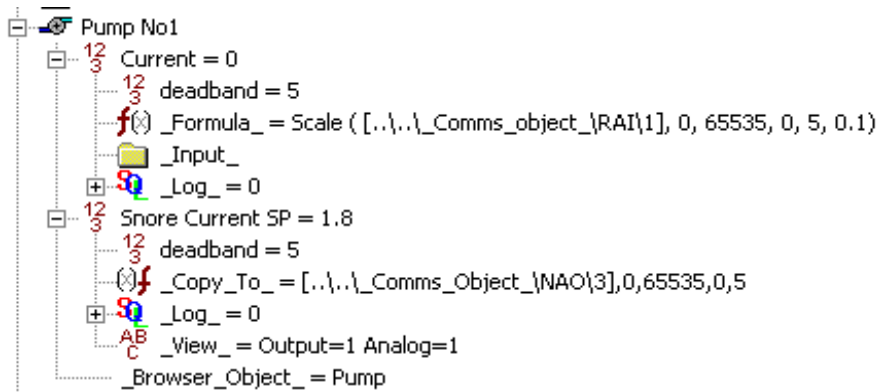
There are many different functions that can be used inside a `_formula_` node allowing the user to create some quite complex functionality. Details of these can be found in the DATRAN online help.

11.4. `_Copy_To_` Nodes

As described in the previous section, a `_formula_` node can be used to read a node somewhere else in the TBD file and to publish the result into its parent. In this situation, the `_Formula_` node can be thought of as “pulling” a value from elsewhere in the TBD file and presenting for the user.

In some situations, however, it makes more sense to “push” a value to another location in a TBD file. This is usually done with outputs such as set points or On-Off controls. This is where the `_Copy_To_` node comes in.

As with a `_formula_` node, the basic function of a `_Copy_To_` node can be described quite simply; A `_Copy_To_` node takes the value of its parent node and pushes it to the designated location in the TBD file. It can also reverse-scale the value at the same time, if desired.

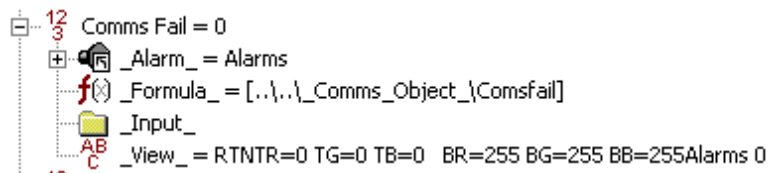


We can see in the picture above, a portion of a TBD file where “Snore Current” set point is being transmitted to the RTU. This particular set point will cause an alarm to be triggered when the pump current is less than the snore current set point.

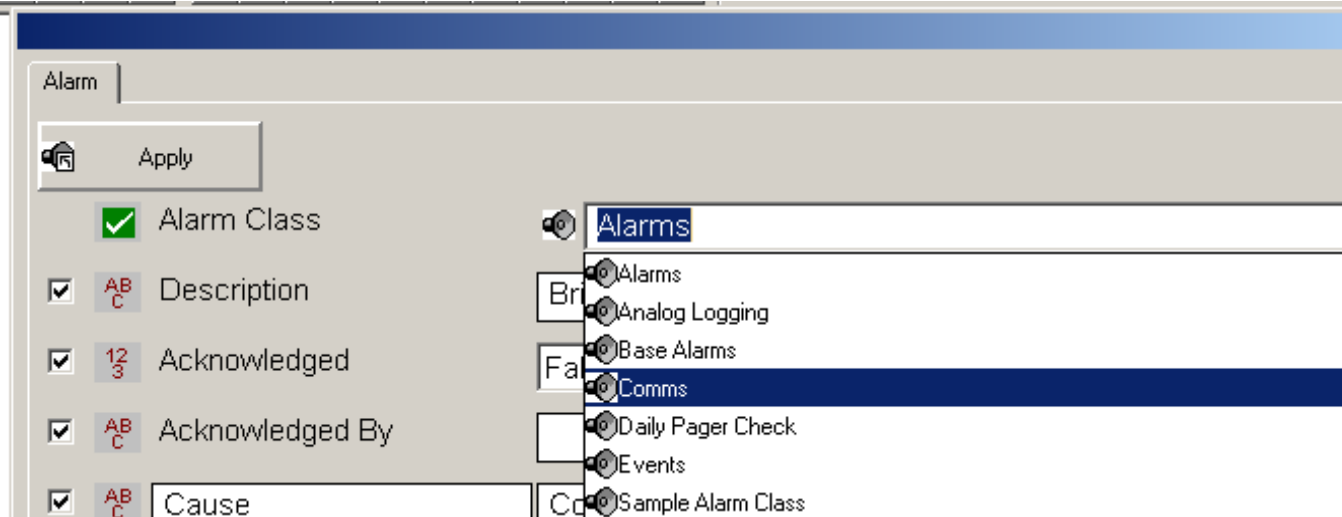
Notice the use of the `_Copy_To_` node under the snore current SP node. This takes the value of the parent node and publishes it to the specified location, in this case `NAO\3` in the comms object. Notice that the extra parameters that specify scaling are in the same order as the `_Formula_` node above it. This means that when the value 1.8 arrives at `NAO\3`, it will have already been rescaled to a RAW number for use in the RTU. The value will be 23593.

11.5. Alarm Nodes

Often, a specific action is required when an I/O point changes state. These actions can range from something as simple as logging the event to the DATRAN Database all the way through to something more complicated like sending an SMS message to an operator to alert them that a problem needs dealing with. DATRAN achieves this using `_Alarm_` nodes.



An `_Alarm_` node, as seen in the screenshot above, triggers whenever the parent node of the `_Alarm_` changes state. When the `_Alarm_` triggers, it calls the “Alarm Class” to which it has been assigned to execute.



The alarm class can be changed at any time by double clicking on the `_Alarm_` node. This will open a dialog box that allows the user to select from all the alarm classes that have been defined in the system. Once changes are made the user must click “APPLY” for the changes to be applied by the system.

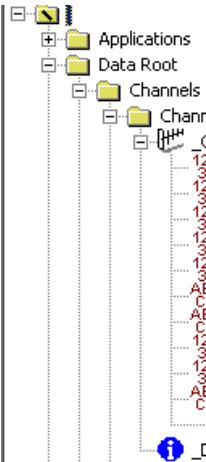
More information on Alarm Classes can be found in section 16

12. ADDING SITES AND THEIR ASSOCIATED CONFIGURATION ISSUES

12.1. Site Topics

- The Station Template
- Adding a new site to an existing system
- Adding a new discrete point to a system
- Adding a new analogue point to a system
- Removing a site

12.2. Editing a communications channel



In a standard TBD file, all communications channel configuration is contained in the Data Root\Channels section of the TBD file, as shown. Each of the sub nodes under the “Channels” node specifies the name of each channel.

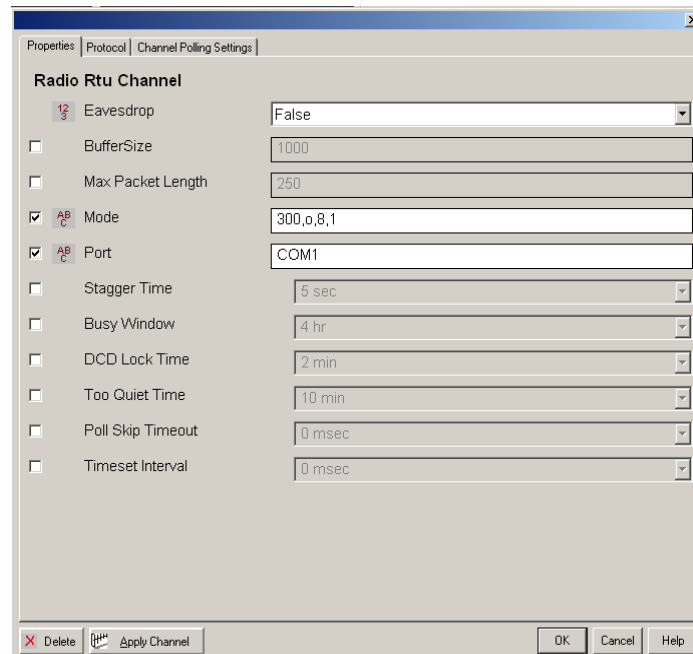
Each channel has a `_Comms_Object_` in a similar way to each site in the system also has a `_Comms_Object_`. Under the `_Comms_Object_` there are several nodes that show statistics on the comms channel.

The screenshot shows a tree structure with the following nodes under 'Channel A':

- `_Comms_Object_`
- `Busy (%) = 0`
- `Busy (%) 24hr = 0`
- `DCD = 0`
- `DCD Locked On Alarm = 0`
- `Eavesdrop = 0`
- `Mode = 300,o,8,1`
- `Port = COM1`
- `Site Count = 0`
- `Too Quiet Alarm = 0`
- `Type = RtuChannel`
- `_Browser_Object_ = Radio Rtu Channel`
- `_Documentation_ = Radio Channel`

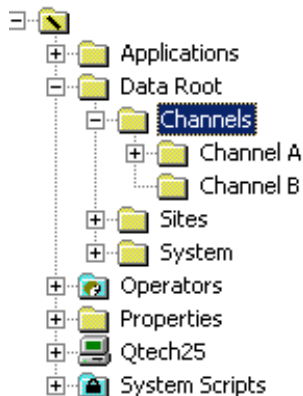
- *Busy (%) and Busy (%) 24hr* give an average view of how busy the comms channel has been over the specified period.
- *DCD Locked On Alarm* shows if the DCD signal coming from the modem has been on continuously for a set period. This may indicate a site has its transmitter locked on.
- *Too Quiet Alarm* this shows when there has been no activity on the DCD signal for a given period. This may indicate the radio at the base station has been turned off or unplugged, or that the repeater (if any) has failed.

Double-Clicking the _Comms_Object_ will open a dialog to allow the user to configure the comms channel. The screenshot below shows the dialog for a Radio Comms Channel. The properties on other media may be different, and the more commonly used items are explained below.

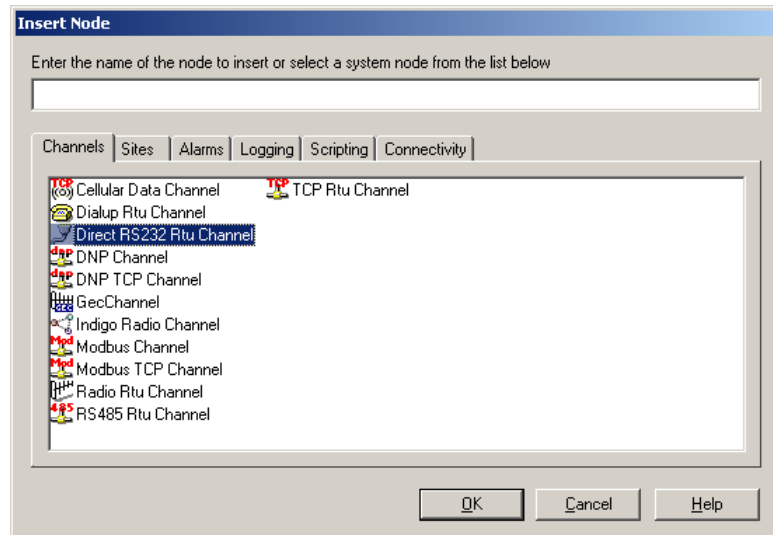


- Eavesdrop mode puts the comms channel into a mode where it will listen to a radio channel but prevents it from transmitting.
- Max Packet Length specifies the maximum packet length for packets being transmitted on this channel. For radio channels a maximum of 250 Bytes is normal, but on faster media (Ethernet, Data Radios) larger packets are desirable.
- Mode has four parameters; <baud rate>,<parity..(e)ven, (o)dd or (n)one>,<data bits>, <stop bits>. Baud rate for conventional radio channels is either 300 or 1200 baud. Parity is usually 'o' for Odd parity. The number of data bits and stop bits is almost never changed.
- Port specifies which serial port is used for this channel.
- Stagger Time specifies how long the base station should wait between getting a response from one station before transmitting the next packet in its queue.
- Busy Window specifies the averaging period for the Busy (%) statistic.
- DCD Lock time specifies how long DCD must be on before the DCD Locked On Alarm is raised.
- Too Quiet Time specifies how long DCD must be off before the Too Quiet Alarm is raised.
- Poll Skip Timeout allows the base to be configured to ignore any scheduled polls of the site if the base has recently received a COS message from the site.
- Timeset Interval specifies how often the base should automatically send Time Set commands to each RTU on this channel.

12.3. Adding a new communications channel – Worked Example

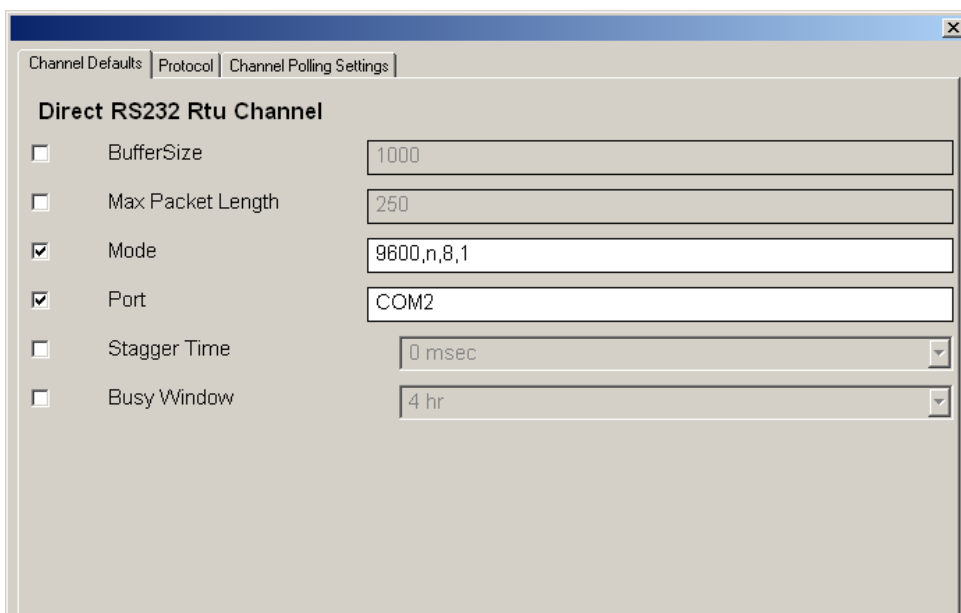


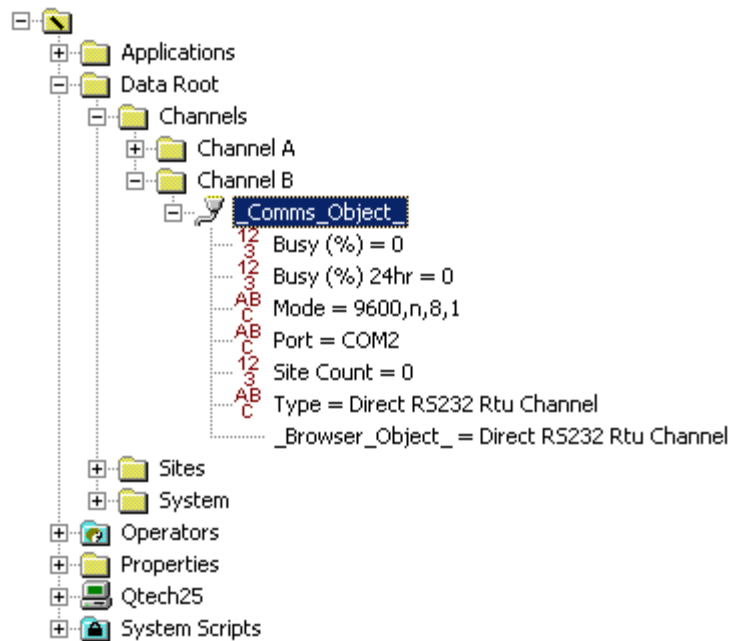
Select the channels node in the TBD file, right click and “Add Node”. The following dialog will appear and type the name of the new channel (in this case, Channel B) into the top field.



Select the type of channel required and click OK. This will open the channel configuration dialog for the channel type you have chosen. In this example we are using “Direct RS232 RTU Channel” which you would use to plug an RTU in directly to the base station serial port. It is also the channel type you would use when setting up a channel to use serial data radios, such as the Tait TM81xx.

Configure the serial port options for Mode and Port and then click “Apply Channel”. If you only click “OK” then the options you entered will be remembered, but the system will not use them to open the serial port.





12.4. Station Template

Most DATRAN TBD files have a built-in system script that is called the “Station Template”. This script automates several tasks involved with the creation of new sites or new points in existing sites. Depending on how a particular system is configured, the Station Template may carry out the following tasks:

- Filling out various sub nodes of an `_Alarm_` node that are required for its correct operation.
- Write settings for displaying points in DATRAN Supervisor as inputs and outputs.
- Write DDE Server settings
- Set up Analogue Logging for analogue points
- Insert Pump Run Hours accumulation scripts.

To run the station template, highlight it and press F2. The node will change to a yellow folder icon while it is running and then change back to a green tick when it is finished. Most of the time this will happen so fast the user will not see it happen.

An important point to note about the Station Template is that some of the tasks that it carries out the first time again require it to be run a second time to complete the job. For example, the first time it is run it may create some Pump Run Hours accumulators which require logging to the database. The script would then have to be run a second time to add the logging under these points. To be safe, the station template should always be run twice when it is used.

12.5. Adding a New Site to an Existing System

Here is a list of things to do when adding a new site:

Depending on the system there will be different requirements. The system we are assuming here is one that uses DATRAN Supervisor as the front end but does not have an HMI or DATRAN Trending. There are however some additional notes about what to do if these components are used on a system.

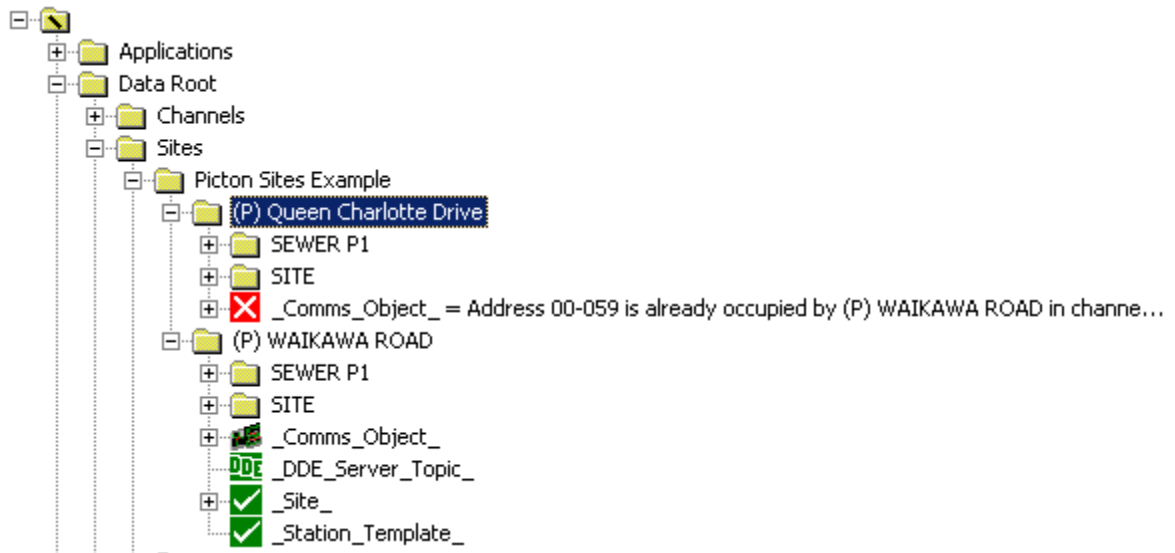
1. Add the *Site* node for the new site.
2. Copy parts of other sites that are similar including the *Comms Object*.
3. Change the comms address in the *Comms Object* and any other communications specific settings, such as I/O count and DLP file location.
4. Change any formulae or alarm settings. It is a good idea to delete any nodes under an '*_Alarm_*' node that has been copied from another site at this point, as this will all be information from the old site.
5. If used, insert the *Station Template*. This will carry out several tasks that are applied automatically, including re-building all the information under the '*_Alarm_*' nodes that was deleted in step 4. Select the station template node and press F2 to cause it to execute a second time. (The station template should always be run twice.
6. Add a '*_Site_*' node to the new site. This makes this site visible to DATRAN Console/Supervisor.

An Example:

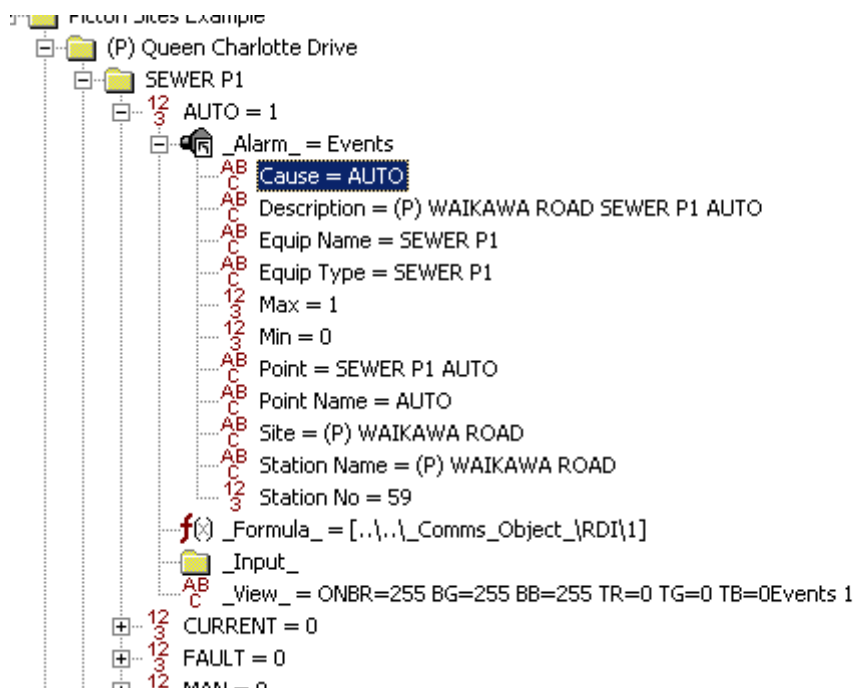
We want to add a new site to the Picton Telemetry System. The site is to be called Queen Charlotte Drive and will have the same points as the Waikawa Rd site.

What to do:

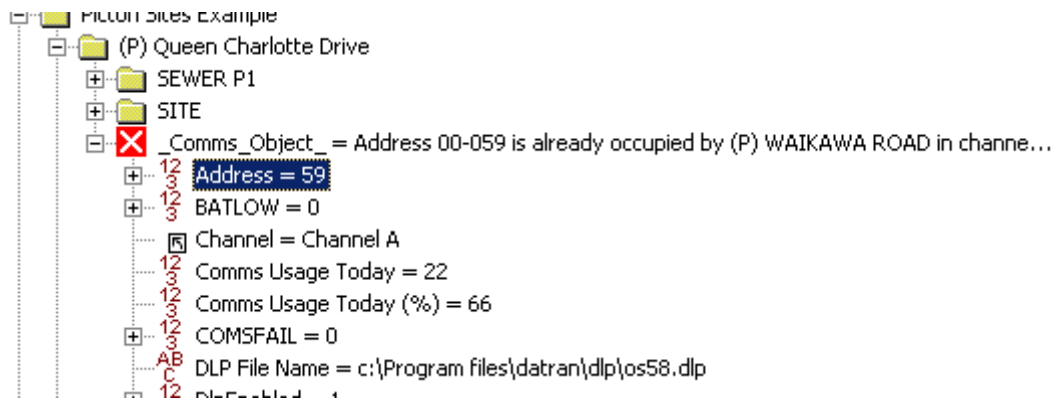
1. Insert a new node under *Sites* called Queen Charlotte Drive.
2. Copy the Comms Object from the Waikawa Rd site and paste it to Queen Charlotte Drive.
3. Copy the '*Pump No 1*' and '*Site*' node from the Waikawa Rd site and paste them to Queen Charlotte Drive.



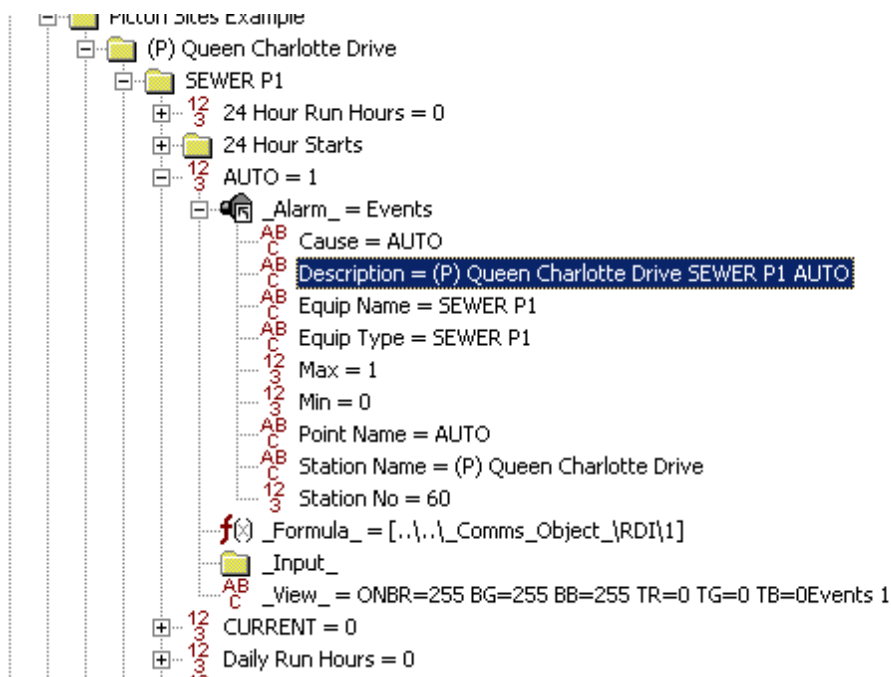
4. Delete all the *Sub Nodes* under any '_Alarm_' nodes you also copied across in the previous step. Note how the alarm description still contains the information for the original site?



5. Change the address of the Queen Charlotte Drive site in its _Comms_Object_.



6. Insert the *Station Template* node under Queen Charlotte Drive. Note how the alarm description has been regenerated, now with the correct information.



7. Add a *'_Site_'* node under Queen Charlotte Drive, then delete it. This is another script that will generate some additional nodes that control how the point is displayed in DATRAN Supervisor.

12.5.1. Third Party GUI Software Is Used

Add a *'_DDE_Server_Topic_'* node under the site node, if using DDE, or an *'_xml_datatopic_'* node if using OPC. This will create a topic on the DDE or OPC server for this site and allow the external application to access I/O. Remember you may need to restart your GUI application for it to recognise the new I/O points exist.

12.6. Adding a New Discrete Point to a Site

The system we are assuming here is one that uses DATRAN Supervisor as the front end but does not have a *HMI* or *DATRAN Trending*. There are however some additional notes about what to do if these other components are used on a system.

The following approach is not the only way but probably the quickest way to make a new point.

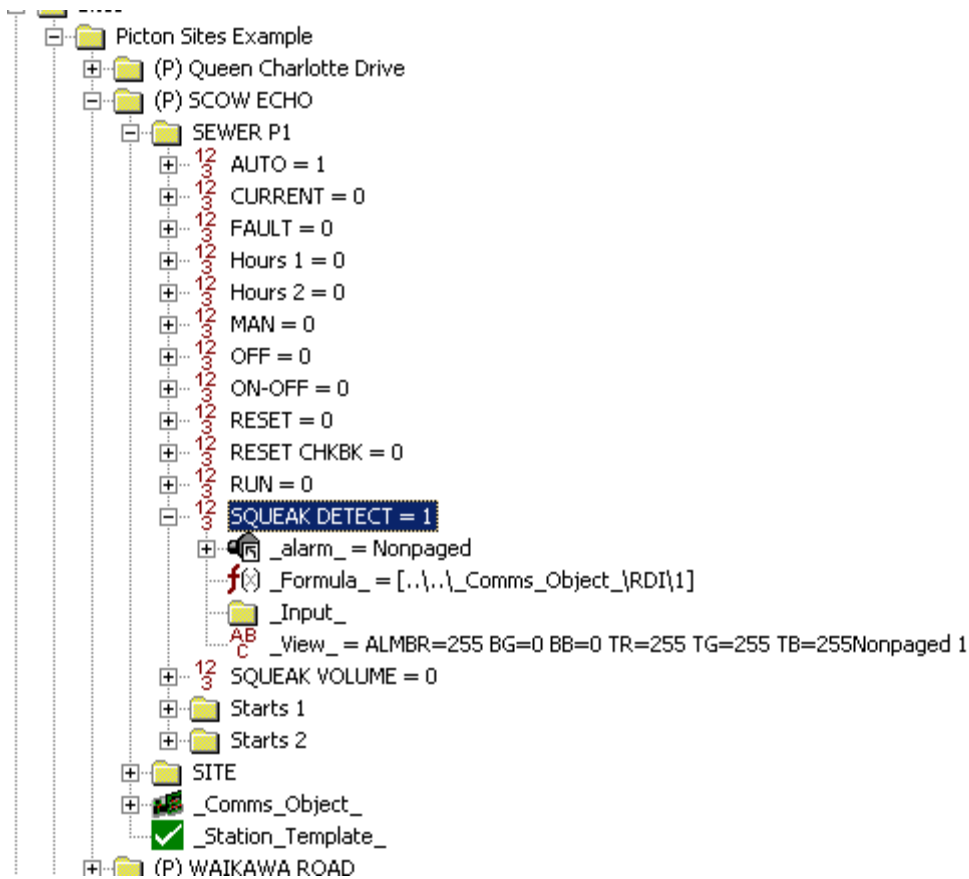
To Do:

- ❑ Copy a similar point from the same or a different site.
- ❑ Modify the alarm class and formula as appropriate.
- ❑ Apply the *Station Template* to rebuild the alarm description.

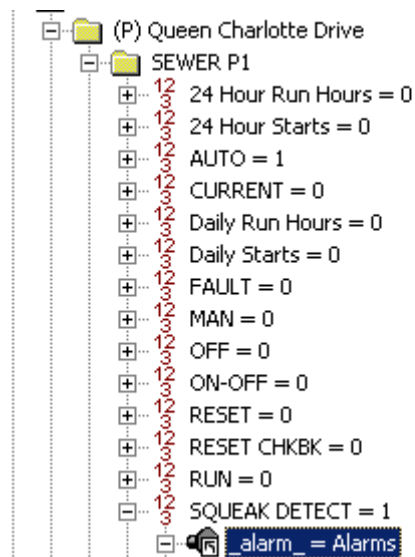
Here Is An Example

The pump at Queen Charlotte Drive has had a squeak detector fitted and wired to the digital input number 6 on the QRTU. The squeak alarm is to be processed by the *Squeak Alarm Class*. Here is what to do:

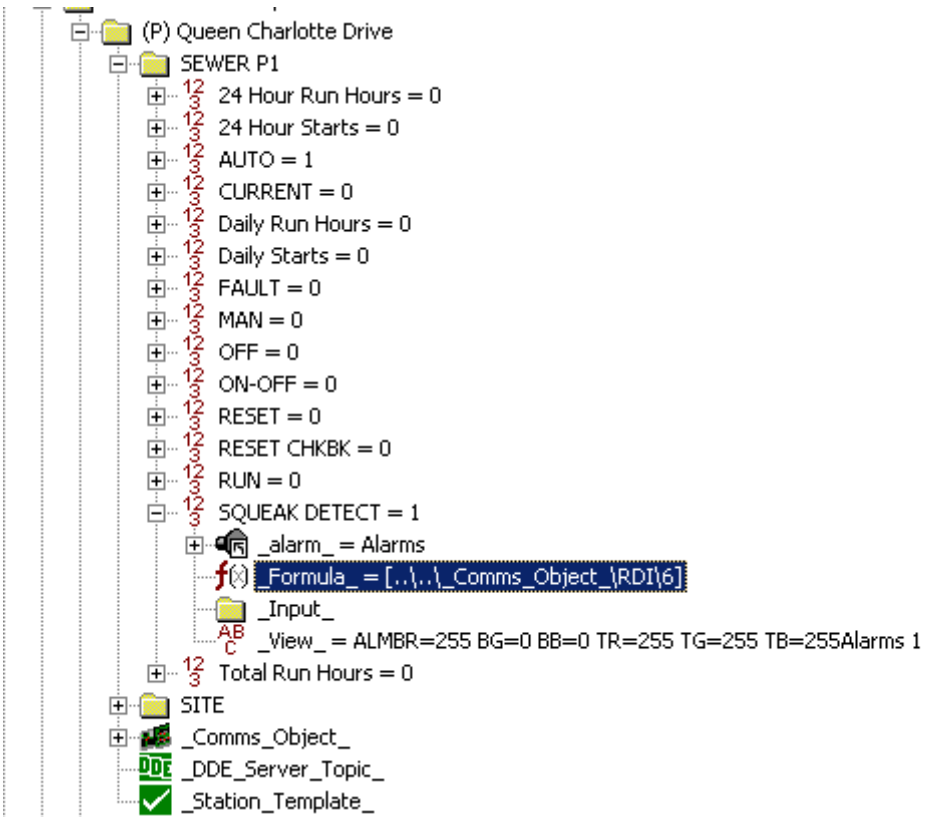
1. Copy the Squeak Detected point from the Scow Echo site to the *Pump* node on the new Queen Charlotte Drive site.



2. Change the *Alarm Class* to 'Alarms' (from 'Non Paged').



3. Change the formula under the node so that it connects to RDI6 instead of RDI1.

4. 
 - (P) Queen Charlotte Drive
 - SEWER P1
 - 24 Hour Run Hours = 0
 - 24 Hour Starts = 0
 - AUTO = 1
 - CURRENT = 0
 - Daily Run Hours = 0
 - Daily Starts = 0
 - FAULT = 0
 - MAN = 0
 - OFF = 0
 - ON-OFF = 0
 - RESET = 0
 - RESET CHKBK = 0
 - RUN = 0
 - SQUEAK DETECT = 1
 - _alarm_ = Alarms
 - Formula_ = [...]_Comms_Object_\RDI{6}
 - _Input_
 - AB
 - C
 - _View_ = ALMBR=255 BG=0 BB=0 TR=255 TG=255 TB=255Alarms 1
 - Total Run Hours = 0
 - SITE
 - _Comms_Object_
 - _DDE_Server_Topic_
 - _Station_Template_

5. Delete all the sub nodes under the _Alarm_ node for this point and then run the Station Template to re-create the correct ones.

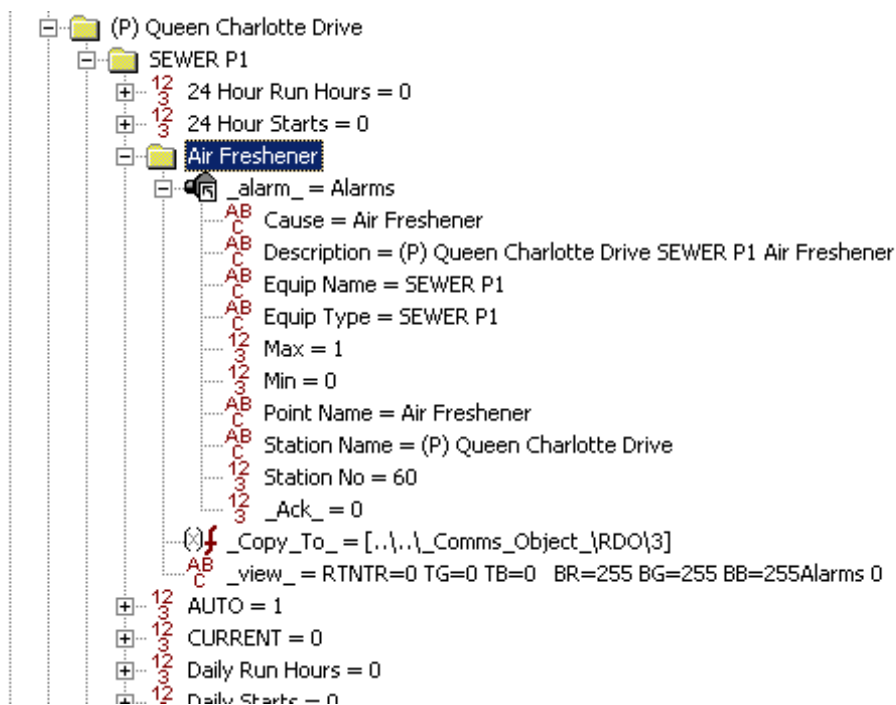
Example: Adding An Output Point

The pump at Queen Charlotte Drive has had an air freshener fitted. This has been wired to the QRTU on 'Digital Output 3'. The intention is that when staff happen to be visiting this site, they can ring the operator at the *SCADA Base Station* and request the air freshener to be operated. This is so that the site is pleasant to work in.

So that staff have the confidence that the air freshener has been applied, the operation of the air freshener will be treated as an Alarm and it will be paged to the staff in the field.

Here Is What To Do:

1. This time there are no other similar points so insert the air freshener point under the Pump node.
2. Insert a '_Copy_To_' node under the air freshener point and link it to 'RDO 3'.
3. Insert an '_Alarm_' node under the air freshener node and allocate it to the 'Alarms' Alarm Class.
4. Insert a '_View_' node under the air freshener node and in the properties box on the property sheet type 'Output=1, Digital=1'.
5. Reapply the Station Template.



12.7. Adding a New Analogue Point to a Site

The system we are assuming here is one that uses DATRAN Supervisor as the front end but does not have an HMI or RTU Data logging. There are however some additional notes about what to do if these other components are used on a system.

The following approach is not the only way but probably the quickest way to make a new point.

To Do:

- ❑ Copy a similar point from the same or a different site.
- ❑ Modify the *Alarm Class* and *formula* as appropriate.
- ❑ Apply the *Station Template* to rebuild the alarm description.

Here Is An Example

The pump at Queen Charlotte Drive has had a Squeak volume detector fitted and wired to the analogue input number 1 on the QRTU 'The squeak'. Here is what to do:

1. Copy the 'Squeak Volume' node from scow echo and paste it under the 'Pump 1' node of Queen Charlotte Drive.
2. Change the formula so that it connects to 'RAI 1' instead of 'RAI 2'. Also remember to change the scaling of the analogue if the detector has a different span at the new site.
3. Delete the '_log_' node from under the Squeak Volume node, and run the Station template, twice.
4. Under the '_log_' node that the station template creates, you will find an '_Alarm_' node. Double click this to open its configuration dialog and change the Min and Max values to whatever is appropriate. These values will be the defaults that are used when the pen is displayed in DATRAN Trending.

12.8. Removing a site from a system

Sometimes it becomes necessary to remove a site from a system if the site is going to be out of commission for a number of weeks for maintenance, or when the site is being removed altogether.

12.8.1. Permanent Removal

Permanent removal of a site from a DATRAN system is as easy as selecting the site in DATRAN Browser and deleting it. If the site has a lot of configuration under it then it may take a long time to delete. DATRAN Browser will warn the user if the delete operation is going to delete more than 1000 sub nodes. If this is the case, some

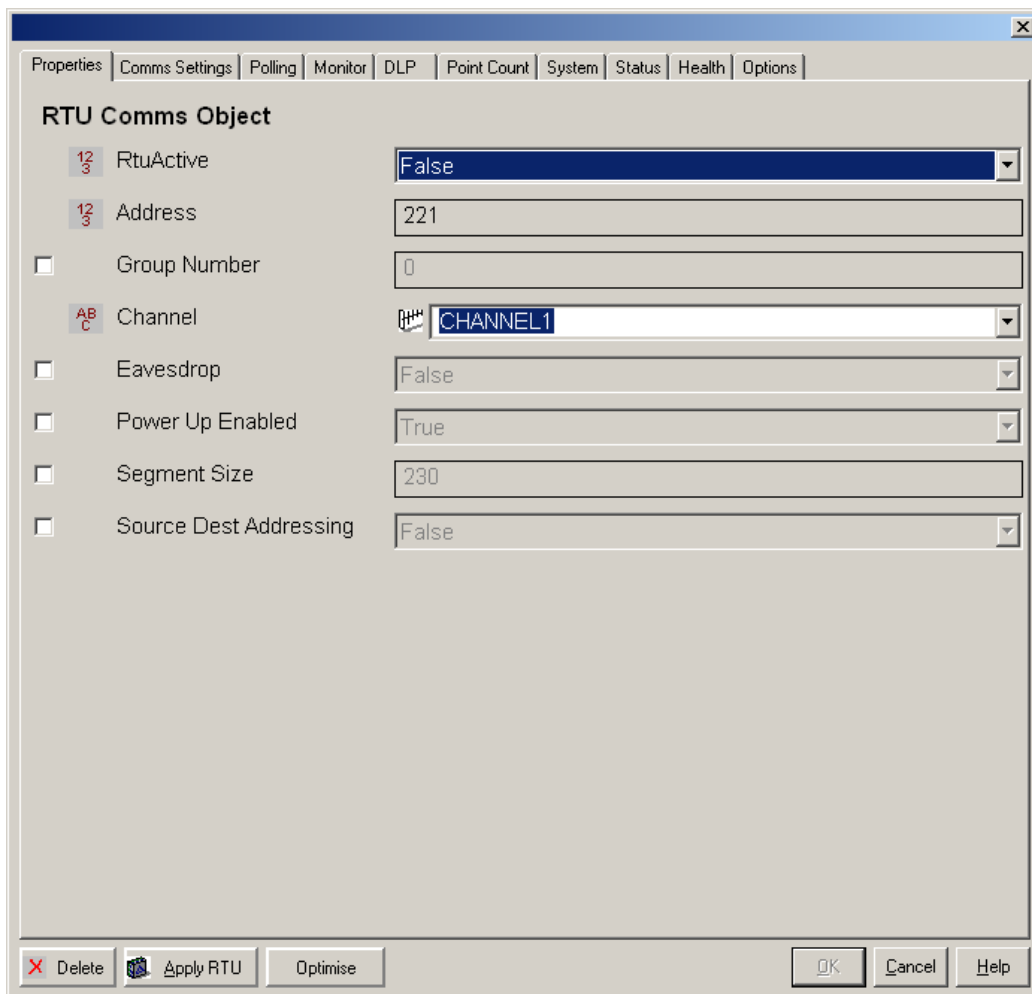
thought should be given to deleting the site a piece at a time, rather than all in one go.

After the site has been deleted, it is advisable to restart the system to clear any residual information that may still be present in I/O servers like DDE or DATRAN TCP server.

In most cases it would be prudent to make a back up of the TBD file prior to deleting a site, “Just in case”.

12.8.2. Temporary Removal

If a site needs to be disabled on a temporary basis, this can be achieved by using the RTU Active property in the `_Comms_Object_` for that RTU.



When RTU Active is set to “False” the base station will make no attempt to communicate with that site, and will ignore any incoming comms from that site. This way, Radio bandwidth is not wasted trying to communicate with a site that is out of commission.

Because some alarms that are “normally closed” may cause false alarms when stations are in this state, it is common that such alarms have an additional piece put in the formula node so that the alarm can only be active if RTU Active is set to true.

For example :

`[.._Comms_Object_ \Comsfail] And [.._Comms_Object_ \RtuActive]`

13. DATRAN VI – SCRIPTS, TEMPLATE'S AND MACROS IN THE TBD FILE AND BASIC FUNCTION/USE

13.1. The Script, Macro and Template Nodes

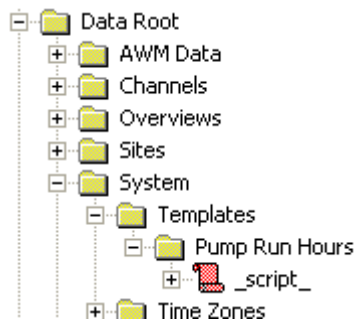
Scripts, Template's and Macro's are the tools which the advanced user has to perform small amounts of program logic on nodes and amongst nodes within the DATRAN TBD file. These three objects serve three distinct and different purposes.

- **Scripts** – A script is a collection of commands that have been put together to perform a task and are generally written to perform this task on several points throughout the TBD file. All scripts have properties, which can be added to the script for special functions.
- **Templates** – A template is the means by which a script is utilized throughout the TBD file. For example if you wrote a script that was to set a node at midnight, then you would write it in such a way that when you placed a template of the script under a node, it would activate at midnight. Depending on the way a script is written, the Template can normally be manually activated in three ways: Select it and press alt-a, double click it and then click 'Apply' or right click it and select 'Apply' from the pop-up menu.
- **Macros** – Macros are small portions of commands that can be read into a script. These nodes act as functions to a script. For example if you where to write a script that might read in a file and then scan the file for the number '1' the script might contain a list of Macro's as follows.
 1. Open File
 2. Read File
 3. Scan File
 4. Close File

It should be noted that a Macro is pasted into the script from where it is called so if variable (Name X) is altered in one Macro then it will remain modified in any subsequent uses of the variable name in other macros in the script.

13.2. The Script Editor

To create a script in DATRAN, create a node with the name of the script that you wish to create. This node is usually placed under the '*Data Root\System\Templates*' node. Underneath that node, create a '*_Script_*' node. This will automatically start the DATRAN script editor.



If you want to edit a pre-existing script, then simply double-click on the ‘*_script_*’ node, or right-click on it and select “Edit” from the pop-up menu.

Right clicking in the editor window will allow you to browse all the available functions and statements to insert them in your script. You can also manually type these in if you wish. There is an extensive online help available for most script commands, this can be accessed by clicking on the keyword you want help with and pressing **F1**.

You can also insert node references via the right-click menu by browsing the TBD file in the same manner as when you edit a ‘*_Formula_*’. It is important to note that when browsing for nodes, the interface will create a node reference that is relative to the position of the ‘*_script_*’ node in the TBD file. However when the script executes, DATRAN will treat the node reference as relative to the ‘*_template_*’ node that called the script to run. An easy way around this discrepancy is to place the ‘*_script_*’ node at the same position in the TBD file as you intend the ‘*_template_*’ node to be while you are editing it, and only move it to the *Data Root\Scripts\<script name>* node once you have finished.

There are three buttons that will let you exit the script editor; “*Apply Script*”, which will save the changes you have made, and then update the system with the new script. “*Exit*”, which will save your changes to the script, but will not update the system to use the changes until the script is applied, and “*Cancel*” which will abort any changes you have made and restore the script to the state it was in before you started the editor.

13.3. Script Triggers

There are several ways a script can be triggered to run in DATRAN, and a script can utilize one, or many of these to behave differently depending on how it was triggered.

- ❑ **OnStart:** Causes the statements that follow to be executed when the *Derived Value Generator* service starts up. This will usually only occur when a system is rebooted, but the service may also be manually restarted by itself if the user desires.
- ❑ **OnDo:** Causes the statements that follow to be executed when Template or Command nodes are created, or when the script is applied, but *not* then the DVG first starts up.
- ❑ **OnPoll:** Causes the statements that follow to be executed when the template is polled after a poll statement.
- ❑ **OnTrap:** Causes the statements that follow to be executed when the template responds to an event trap set up by [TrapSingle](#) or [TrapMultiple](#).
- ❑ **OnUndo:** Causes the statements that follow to be executed in *reverse order* when Template nodes are deleted, or when the script is redefined.

For Example:

OnStart OnDo:

```
TrapSingle( [_Comms_Object_]ForcePoll );
```

OnTrap:

```
If ( [_Comms_Object_]ForcePoll == 1 )  
{  
  [ForcePoll Status] := "Polling";  
}  
Else  
{  
  [ForcePoll Status] := "OK";  
}  
TrapSingle( [_Comms_Object_]ForcePoll );
```

Will cause the script to execute when either DVG starts up, or when the *'_template_'* is reapplied, or if the *'_script_'* is redefined.

When the script first executes, it will only execute the first **OnStart OnDo:** section. This section sets up a trap event on (in this case) the *ForcePoll* node under the *'_Comms_Object_'*. This means that when the *ForcePoll* Node changes state, the **OnTrap:** Section of the script will execute, and will change the status of the [ForcePoll Status] node will change depending on the value of the *ForcePoll* node in the comms object.

13.4. Template Keys

DATRAN does not allow two nodes under a given parent node to have the same name, this means that you can only have one *'_template_'* node under any given node in the TBD file. This could be problematic if you wanted to run several different scripts in the same location in the TBD file. One option might be to have each script write a different value in the template node as it finished, effectively calling the next script to run when it had finished, but this would not allow for things like polling or trapping to work very well. To get around this, every script has an optional parameter called the *"Template Key"*. This allows the user to call a script by creating a node with the same name as the template key, instead of using a template node.

For example, if we had created a script called "Pump Run Hours", up until now the only way we had of calling that script was to create a *'_template_'* node and specify that script to run. But if we were to add the *"Template Key"* parameter to the script and call it *"_Pump_Run_Hours_"*, we can now call the script from anywhere in the TBD file by creating a node called [_Pump_Run_Hours_]. This is the same way the Station Template script is triggered by creating the [_Station_Template_] node.

By convention, all Template Keys should utilize the underscore character in place of any spaces in the key, as well as having a leading and trailing underscore character. This way, a user can easily spot that a given node is actually a Template Key, rather than just a normal node.

One of the drawbacks of using a Template Key, is that the user will not be able to *'apply'* it in the same manner as was possible with the *_Template_* node. In fact, the only way to trigger an *"OnDo:"* with a Template Key is to either reapply the

appropriate script (which will cause ALL the instances of the Template Key to re-execute simultaneously), or to delete the node that is the Template Key, and to remake it. To get around this, it is good practice to have the script set the node that is the Template Key to a success type, and then to trap it. This way, we can trigger the script to execute on an *“OnPoll:”* by clicking the Template Key node and pressing F2, which will change it to a NULL type.

14. DATRAN VI – DISASTER AVOIDANCE AND RECOVERY

14.1. Disaster Recovery and Avoidance

The following will be covered

- ❑ Disaster Avoidance
- ❑ Backing up a DATRAN system to tape or other media
- ❑ Using DATRAN Server Manager to make automatic back-ups
- ❑ Loading a back-up TBD (Tree Base Data) file

14.2. Disaster Avoidance

There are a number of techniques that can be utilised in DATRAN VI to avoid disaster. By disaster we mean loss of a TBD file, failure of a DATRAN system without knowledge of operators or inability to rebuild a failed system quickly. Here is a list of things that can be done to avoid the disasters mentioned above:

- ❑ Set up the automatic back up by *Server Manager*.
- ❑ Make back-ups to tape or other media of the TBD files that are produced by *Server Manager*.
- ❑ Set up *Server Manager Mission Critical Checking*. If a DATRAN Service fails an operator could be paged.
- ❑ Using the windows service manager, set up each service to automatically restart when they fail.
- ❑ Set up a watch-dog system by sending the *Server Manager Heart Beat* to an RTU. The RTU would ring an operator using an auto-dialler or send an SMS message using a cellular SMS modem.
- ❑ Take regular drive images of the DATRAN machine and store them in a safe place. With these, any identical hardware can be re-imaged and up and running quickly.
- ❑ Set up system [ALARMS](#) that will be triggered when system parameters are getting near their limits. Examples are the amount of free space on C:\, the size of the TBD file, the size of the database etc.

14.3. Alarms

- ❑ Low hard disk space

- ❑ Too many nodes in DATRAN VI or the TBD file is over a certain size.
- ❑ Alarms related to SQL Server where this used to store historical data.
- ❑ Alarms alerting the operator to a failure of any of the DATRAN services.

14.4. Backing up a DATRAN System

The following files need to be backed up:

- ❑ The TBD files. The default place to find the main one is in C:\Program Files\Datran\Data. The *Server Manager* back ups are stored in C:\Program Files\Datran\Backup.
- ❑ Any SQL database data files.
- ❑ Any DLP/ASM files. Normally found in C:\Program Files\Datran\ASM and/or C:\Program Files\Datran\DLP
- ❑ Any HMI configuration files for GUI applications such as InTouch or DATRANVue.

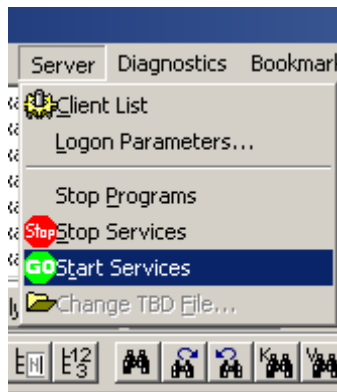
There is no need to back up application files as these can all be reloaded from the DATRAN VI CD.

14.5. Loading a backup TBD

If when viewing a system using *DATRAN Browser* it is noticed that data is missing it is likely that the data in the *DATRAN Server* has been corrupted. Sometimes an unplanned (forced) shutdown of the computer can also cause TBD file corruption if the file was being written at the time the power was removed. The only thing that can be done is to load a back up of the TBD file. Here is how to do it:

- ❑ Stop *DATRAN Services*, if they are currently running. This can be done from DATRAN Browser, or the Windows Services Control (services.msc) If using Windows Services Control, the easiest way is to stop Tree Base Server as all other services have a dependency on this and will also stop automatically.

- Once the services have stopped, rename the TBD file you suspect is corrupt to something obvious. The default location of the TBD file is *C:\Program Files\Datran\Data\<TBDName>.TBD*.
- Copy your back-up TBD file into the *Datran\Data* directory and rename it so that it has the file the corrupt one originally had BEFORE you renamed it.
- Select the *Start Services* from the *Server* menu in DATRAN Browser. Alternatively, *services.msc* can be used to start all the services one at a time.



- The back-up TBD File should be loaded and the system should begin to operate normally.

15. DATRAN VI – THE DATRAN PROPERTIES HIERARCHY

The DATRAN VI SCADA engine as mentioned before is a highly configurable application. And as such has many changeable properties and setup options. To deal with this there is a hierarchical configuration of the properties as discussed below.

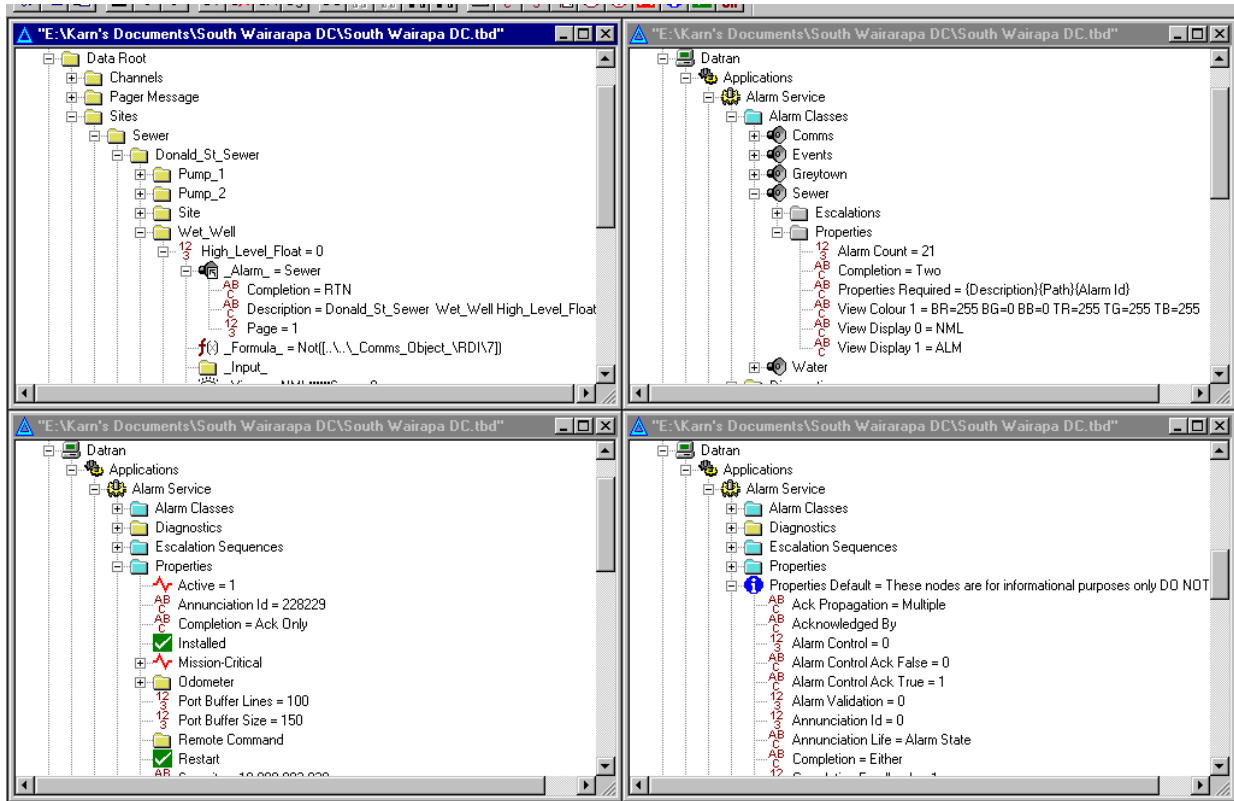


Figure 8: Levels of Precedence

In 'Figure 8' above you will notice that the screen has been setup with four windows. Each window represents one of the four different levels of the hierarchy: *Local*, *Grouped*, *Global* and *Default*. Each level up having precedence to the levels below. In this example we will take an alarm class completion property.

- **Local (Level 1)** As you can see in the top left hand corner we are looking at a point on a site within the Sites configuration area of the TBD. You will notice that there is a 'Completion = RTN' node. This is the local property and has precedence over all other levels. This property is used first if it exists, but does not have to exist.

- **Grouped (Level 2)** In the Top Right hand corner you will see the section of the TBD file which deals with each of the alarm classes. Here you will see the 'Sewer' alarm class and its properties node. It is at this level that we call the property a 'Grouped' variable. This is because the property is being used on a selected group of items (in this case all the 'Sewer' alarms). This property is used in precedence to the 'Global' and 'Default' properties, but again does not have to exist.
- **Global (Level 3)** At the bottom left hand corner of the diagram you will see that we are now looking at the 'Properties' node under the alarm service application. This is what is called the 'Global' property. This property will be used on every alarm class that does not contain a 'Grouped' or 'Local' property. Again this property does not always exist but if it does will take precedence to the 'Default' level property.
- **Default (Level 4)** This is the last rung in the hierarchical ladder and always exists. These values cannot be changed and should not be. Whenever the service in which these properties reside is restarted it will write over the top of any changes. These are the 'Factory Default' settings and are set in such a way as to not cause problems within the system. They have also been set up to the 'Basics', which in many and indeed most cases need to be taken out to another level to tailor to the solutions that are needed.

This is but one example of the application of the DATRAN VI hierarchical system that you will find throughout the different services and applications in slightly different forms. Another common example is the Operator fields as discussed above. The operators Global and Default properties can be found under the Paging application node as the operator information is used predominantly for paging purposes, and sending text messages.

Note: *Not all properties have a 'Default' level*

16. DATRAN VI – THE ALARMING SYSTEM

16.1. The Alarm System

The following topics are covered:

- General functionality
- Description of commonly changed alarm class properties such as alarm completion
- Creating alarm classes, adding escalations
- Allocating alarms to alarm classes
- Alarm Service Diagnostics

Each Alarm class is made up of two key components, the *Escalation Sequence* and the *Property* node. The property node holds the Grouped Properties as discussed previously. The *Escalation Sequence* holds the sequence of events, which are triggered when ever an instance of the alarm class is triggered.

Escalation Sequence – This holds a list of ‘Annunciators’ which perform certain tasks. These are arranged in the escalation sequence to gain the desired actions to be taken when an alarm is triggered.

For example in the ‘Sewer’ alarm class that can be found in ‘Figure 9’ overleaf. The sequence is as follows...

- **Standard** – this is a predetermined set of escalations which are common to several alarm classes. These usually contain such things as ‘Beep’ annunciations and logging to the DATRAN Supervisor and to the Database.
- **Wait** – This is a stock annunciator that waits a set period of time before continuing, in this case to stop intermittent faults being paged out.
- **Alarm Pager Messages** – This is a stock paging annunciator that deals with the paging of alarms out to the correct operator.
- **Wait** – Another wait statement to allow the operator to acknowledge or fix the problem.
- **Alarm Pager Messages** – This is to make sure that someone else gets the message if for any reason the first operator missed the page.

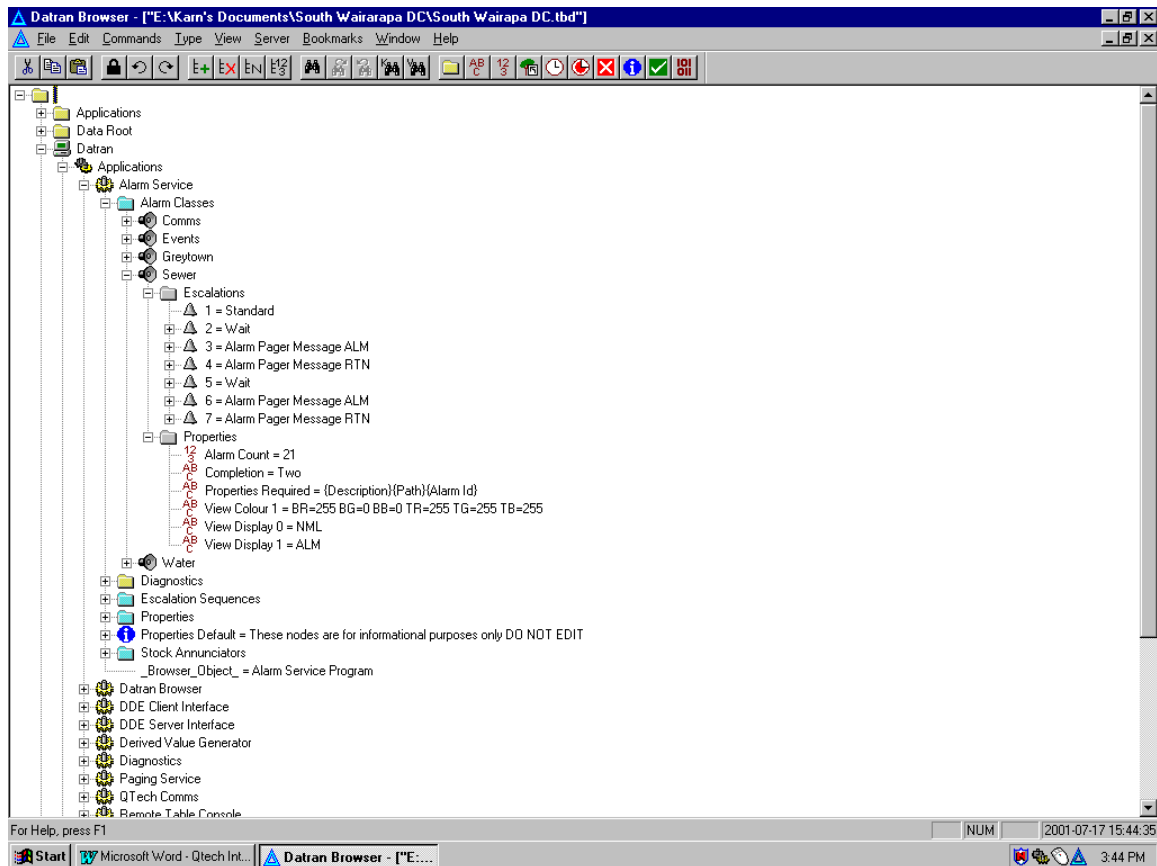


Figure 9: Escalation Sequence

16.2. DATRAN VI – The Annunciator

Each escalation in an *Alarm Class* is linked to an ‘*Annunciator*’. There are many different annunciators to perform many tasks, but again they all have a similar format. Take the standard ‘*Alarm Pager Message*’ annunciator shown below.

As you can see from Figure 10, below, the Annunciator is broken up into three major nodes, ‘*Annunciations*’, ‘*Formats*’ and ‘*Properties*’.

Annunciations – This displays a list of the current annunciations currently being annunciated or that have previously been annunciated and have not yet been deleted.

Formats – This is the most important node structure and most unique. It is separated up into five sub nodes, ALM, ALM Ack, RTN and sometimes Complete. Each sub node represents one of the possible states an Annunciation can be in, ALM for when the alarm is triggered, ALM Ack for when the alarm is acknowledged, RTN for when the alarm returns to normal and Complete for when the Annunciation is complete. Each field (Node) under each of these sub nodes indicates a piece of information that the Annunciator needs to function correctly. You will see in ‘Figure 9’ above that the ALM format has several fields under it. You will also notice that there

are several instances of sets of '{ }' brackets. These indicate that information is needed in this space and the values within them are either the value of nodes from where the Annunciation was triggered or certain system generated values. You will also notice that there is a successfully annunciated Annunciation under the 'Annunciations' node. This shows you how the Alarm service gathers its information. You will notice that all '{ }' fields have been replaced with the relevant information from the nodes surrounding the escalation sequence or 'Alarm_' node that triggered the Annunciator.

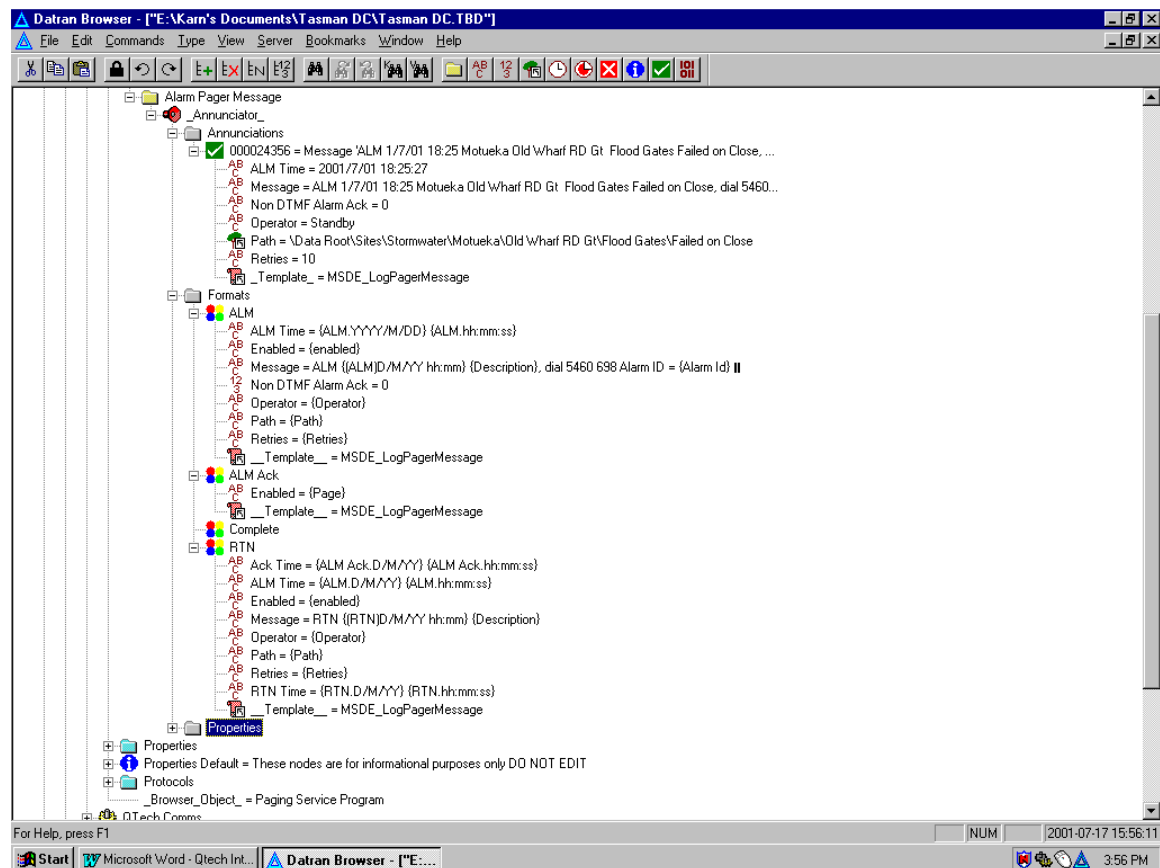


Figure 10: Standard Alarm Pager Message Annunciator

Properties – This as discussed earlier contains the 'Grouped' properties for this Annunciator.

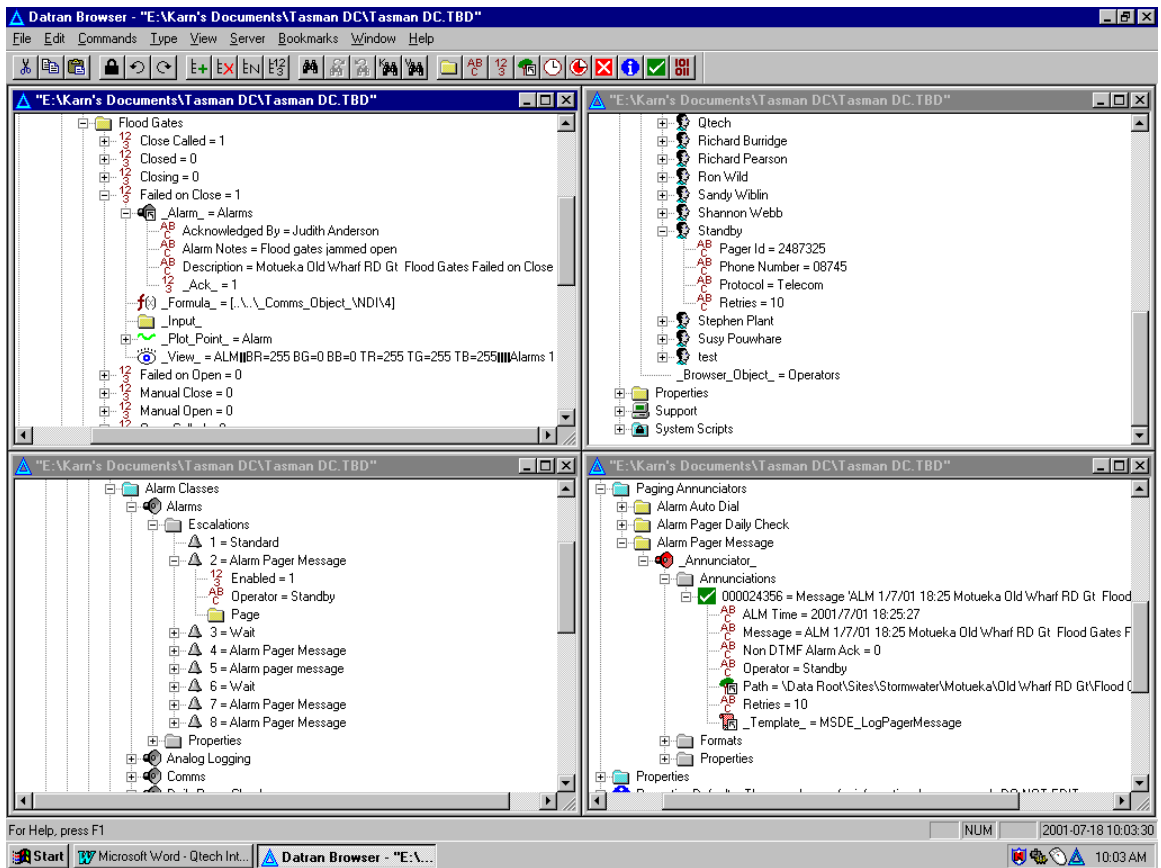


Figure 11: Alarm Pager Message - Annunciations

As you can see in 'Figure 11' above all the Fields that have been filled in the annunciated annunciation in the bottom right hand corner come from certain hierarchical properties throughout the TBD file.

17. ADO CONNECTIVITY MODULE

17.1. System Requirements:

The MSDE or MS SQL Server must be installed on, or has a network connection to the machine running the ADODCM Service (ADO Database Connectivity Module service). Without this installed the database object node will display a "Pending" message and still not get anywhere.

17.2. Overview:

The DATRAN ADO Database Connectivity module runs in the background as a Windows NT service. The module is responsible for logging data from DATRAN VI to a specified MS SQL Server database in a Client-Server fashion.

ADO is the communications interface capable of connecting to both MS SQL Server databases and MS Access databases. ADO provides a data storage layer/interface to Microsoft databases.

The ADO Database Connectivity module never communicates directly with DATRAN VI Browser or any of the other DATRAN services or modules.

Communication is achieved via RPC calls. DATRAN VI Browser has a node that gets updated; the Browser then sends the requested data to be logged to the TBServ (Tree Base Server) via RPC. TBServ then sends that data to the DATRAN ADO Database Connectivity service via RPC which then logs the data to the specified database on the specified server.

DATRAN VI catches a node named [_ADO_Execute_] when it polls (about every 5 seconds) and this node usually has an SQL statement assigned to it. Data logging scripts build an SQL statement and then create one of these [_ADO_Execute_] nodes and assign it with this statement. The statement may be raw SQL or the name of a stored procedure with the appropriate arguments. When DATRAN VI or more specifically the DVG (Derived Value Generator) polls and processes the [_ADO_Execute_] node the execute() method in the ADODCM service is called. This method begins a transaction with the database server, batches and executes the SQL statement. The Query() method creates/updates TBBrowser nodes for each column and row combination in a specified query result set.

17.3. Microsoft SQL Server and SQL Server 2005 Express

Microsoft's SQL Server is an enterprise level database server with many features that enables easy collection of data from a variety of sources in a number of different data repositories. Due to MS SQL Server's design it is always recommended to run the Database Server on an independent Windows 2000/2003/XP server. Independent meaning that MS SQL Server is the only major service application running on the computer. This is a common model for many Microsoft service applications. MS SQL Server 2005 Express Ed. is a cut down version of MS SQL Server; as far as applications are concerned it is the same although it does not support all of the advanced features of MS SQL Server. Also, MSDE 1.0 has a **2GB** limit to each of its databases. MS SQL Server 2005 Express Edition has a **4Gb** limit on each of its databases. MSDE / SQL Server 2005 Express is a data engine, which is replacing the MS Access Jet engine although it still uses MS Access 2000 for the data storage.

Datran Database Logging And Historical Trending Mechanism

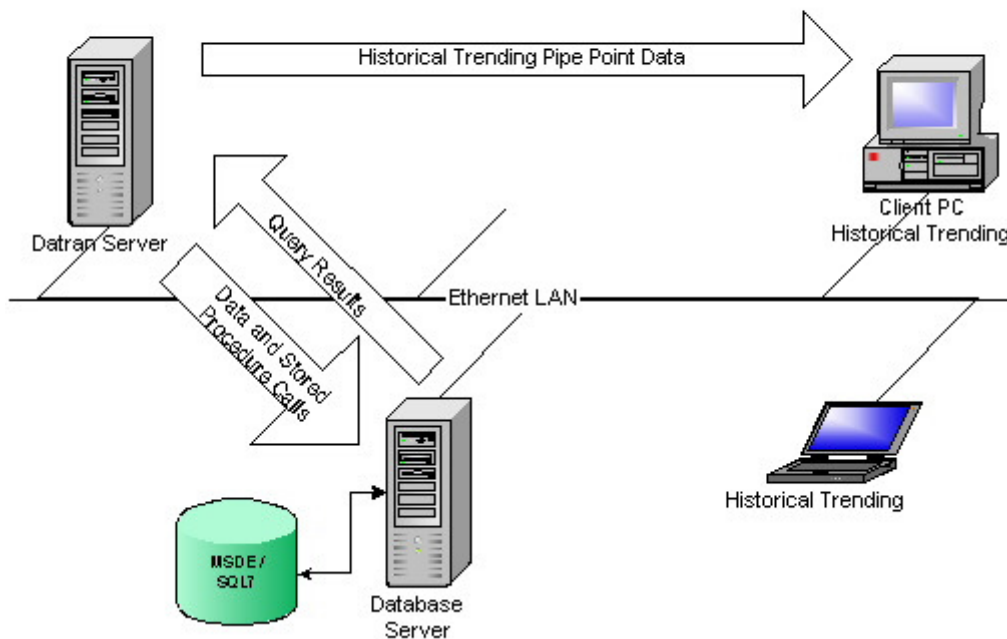


Figure 12 – Database Logging & DATRAN Trending Mechanism

17.4. Setting up DATRAN VI to log to a SQL Database

17.4.1 Installing MS SQL Server 2005 Express Ed. on the DATRAN Server

In order to install the SQL Server 2005 Express onto a Windows computer, Access 2000 must be installed. Do not install SQL Server 2005 Express on a computer with MS SQL Server 7 or above. MSDE is shipped with Access 2003 or Office 2003 Professional. The installer is located in the `\SQL\X86\SETUP` folder and the set-up executable is named *SetupSQL.exe*.

Initialise the installer and follow the prompts. (The default options are acceptable)

When asked which access method is wanted please choose 'SQL Server' rather than 'NT Authentication'. The installer will start and stop the SQL Server 2005 Express service and may require a computer reboot.

17.4.2 Adding a SQL Server 2005 Express database to DATRANVI

To use ADO Data Logging in DATRAN VI the DATRAN ADO Database Connectivity Module (ADODCM.exe) must be installed and running.

Under the *Computer | Applications | ADO Database Connectivity | Property* node check that 'Active = 1'. Add a new node (or modify if it already exists) called Auto Create and set its value to '1'.

Browse to *Computer | Applications | ADO Database Connectivity | Databases* node and insert a new node called DATRANdatabase. A properties dialog will be presented so enter in the relevant details.

- ❑ **Server Name** = <Name of the computer MSDE is installed on>
- ❑ **Login name** = sa or <database user name>.
- ❑ **Password** = <password> often left blank.
- ❑ **Database Name** = DATRANdatabase.

The ADODCM will now execute a series of built-in commands, which will create a database, and its tables, views and stored procedures. You will know it has finished when the database node icon (a yellow database symbol) is visible. After the database has been set-up remember to go back in and set the Auto Create node to 0.

17.4.3 DATRAN ADO Database Connectivity Properties

The main properties (*Computer / Applications / ADO Database Connectivity / Properties*) that are of interest are:

- ❑ **Active** – a value of 1 means the service is running, 0 means the service has stopped.
- ❑ **Auto Create Database** – a value of 1 means that a database will be created if it does not already exist, 0 means it will not automatically create a database.
- ❑ **Active Connection** – “Connected” means that the DATRAN VI has successfully connected to the database and can log data; “Disconnected” means it is not connected and/or DATRAN VI is experiencing problems when logging data.

17.4.4 System Scripts – SQL Database Logging Properties

The following nodes are located under the *System Scripts / SQL Database Logging / Properties* node.

These properties are used for performing advanced functions to achieve database maintenance, data integrity and general housekeeping.

- ❑ **Backup Archive Path** – The backup scripts will use this value as the path to store archives of a backup database.
- ❑ **Backup File Names** – This node holds a list of values which name the files when being backed up. It is unlikely that these values will need to be changed.
- ❑ **Backup Minimum HDD Space** – this value controls the minimum available space on the backup drive for a backup to be attempted. If the available space is less than this value then a backup will not be performed. This value is in Bytes.
- ❑ **Backup Path** – this value is a path to where the database backups will be stored.
- ❑ **Database Space Set point** – this value is used by the SQL Space alarm. If the available space in the database is less than this value then the alarm will be triggered.

- ❑ **DATRAN database Status** – this is no longer used.
- ❑ **FPath** – this value is the file path and pattern that is used to load SQL recovery files.
- ❑ **Keep For** – The length of time SQL error files are to be kept for.
- ❑ **Load Data To File Poll Rate** – length of time before log errors are written to disk. This will happen if the Database is disconnected.
- ❑ **Log Space Set point** – this value is used by the Log Space Alarm. If a query result is less than this value then the alarm will be triggered.
- ❑ **Purge** – how often the files are checked to see if they should be acted upon.
- ❑ **Purge Offset** – offset from the purge value.

17.4.5 Logging Data with DATRAN VI

SQL Data logging is achieved in DATRAN VI via some internal SQL Logging scripts. These scripts can be found under the System Scripts folder in DATRAN Browser. Unless you want to customise the database tables there is no need to make any changes to these scripts.

Digital Logging is performed through use of an alarm. When the point is set to '1' then the alarm is toggled, and the alarm service annunciates the alarm to the SQL Logging Annunciator. The SQL Logging Annunciator calls a script which performs the necessary actions to collate the alarm data passed to it and executes a stored procedure on the Database to log the data into the appropriate tables.

Analogue logging is more complicated. When the station template is executed the script looks for any formulas and 'copy-to's that link to analogue points in *comms objects*. It then creates a 'log' node. This log node is trapped by the Analogue Logging Template that fills in the *log* nodes properties and adds an *alarm* node. The *log params* node (a property of the log node) contains the following properties:

- ❑ **Deadband** – amount the value needs to change before any logging will occur, this value is set so that small fluctuations are logged. Its value is relative to the Full Scale value.
- ❑ **Full Scale** – scaling of the value to be logged.
- ❑ **Interval** – Interval of logging. If no changes in the value being logged are apparent then the value will be logged to the database at the specified interval.
- ❑ **Interval Offset** – offset to the interval to reduce pressure on the system resources. This value is set so that all interval logging occurs at slightly different times to avoid a bottleneck and potential failure.
- ❑ **Equiptype** – specifies the type of the value being logged.

A template traps the value to be logged and when it changes by an amount above the 'deadband' the log node is set to one. This then causes the alarm to be triggered which annunciates to the SQL Analogue Logging Annunciator which in turn executes a script collating the alarm data and performs the necessary actions to log to the database.

17.4.6 Database Alarms

DATRANVI can be configured to create alarms for certain database failures or warnings about potential problems such as, but certainly not limited to:

- ❑ **SQL Backup Failed** – this alarm is toggled by a formula which reads the Backup OK node under Backup. The Backup OK node is set by the backup scripts when they run so if a backup fails the Backup OK node will be set to 0 and the alarm will be generated.
- ❑ **SQL Log Space** – Under Scheduling is a script which calls a stored procedure on the database to report the space used. This alarm is toggled by a script which reads the results of the space used queries and determines whether the space used is greater than the Log Space Checkpoint under Properties.
- ❑ **SQL Disconnected** – This alarm is toggled by a formula reading the Is Connected node of the database. If the Is Connected is set to 0 then the alarm will be generated.

17.4.7 Database Backups

DATRAN VI can execute stored procedures to make the MSDE / MS SQL Server backup its databases. Existing scripts can be found under the *System Scripts / SQL Database Logging / Scheduling node*.

When creating backups MS SQL Server requires a backup device to backup to. This can be either a disk file or a backup tape device. DATRAN uses a disk file which can then be backed up manually either by a scheduled backup task by the system administrator, or by using any other tool to copy the file to another location.

When changing backup locations modify the properties and then apply the Change Backup Location script. The Copy Backup Files node contains a script which will copy the backup files to the Backup Archive path. This effectively maintains two copies of the backed up database. I.e. this week's backup and last week's backup.

17.4.8 Manually querying the MSDE / MS SQL Database

The DATRAN ADO Database Connectivity module is essentially a SQL query pass through agent. The scripts for SQL Logging create SQL queries which get passed to the database via the ADODCM and executed by the server. This means that the DATRAN VI user has all the power of a commercial database query tool at their fingertips.

There are three main node types that DATRAN uses to communicate with the database.

- _ADO_Execute_
- _ADO_Query_

These nodes execute in a very similar fashion. The ADO Database Execute command is used to execute statements and stored procedures that do not require any returned values. Should a return value be needed then use the ADO Query node. After inserting these nodes a dialog will appear allowing the user to enter in their SQL Statements, queries or Stored Procedure calls. Any SQL statement that is allowable on the particular database server being used (MSDE, MS SQL Server 7 or MS SQL Server 2000 or MS SQL Server 2005) is acceptable in a DATRAN SQL node.

So DATRAN VI's SQL connectivity is extremely powerful, even more powerful when coupled with DATRAN VI's internal scripting engine.

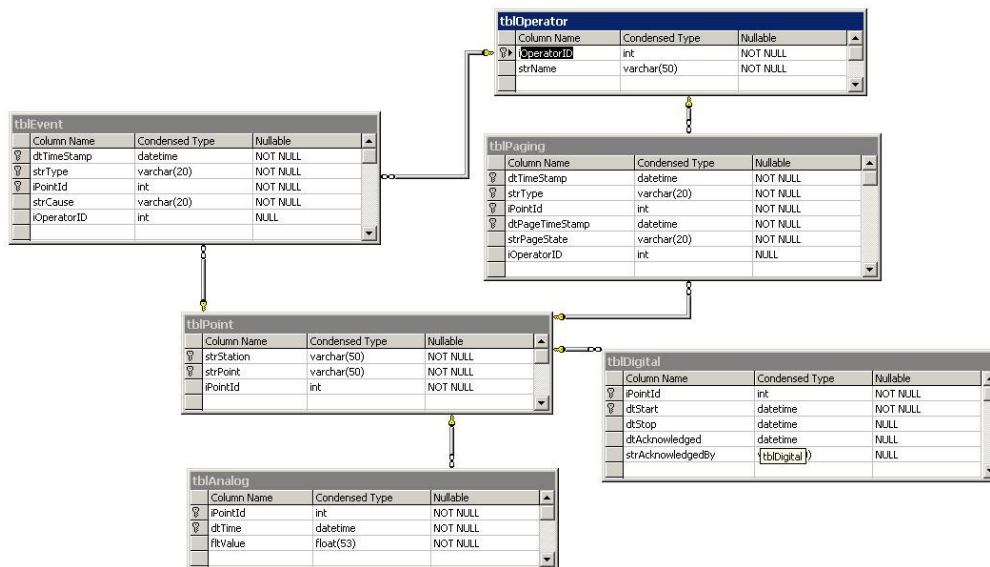


Figure 13: Default DATRAN Database Data Structure

18. InSQL DATABASE CONNECTIVITY MODULE

18.1. System Requirements:

The Wonderware IndustrialSQL/InSQL Server must be installed on, or has a network connection to the machine running the InSqlICM Service (InSQL Database Connectivity Module service). Without this installed the database object node will display an error message and still not get anywhere.

18.2. Overview:

The DATRAN InSQL Database Connectivity Module (InSqlICM) service runs in the background as a Windows NT service. The module is responsible for logging data from DATRAN VI to a specified Wonderware InSQL Server database in a client-server fashion.

A Wonderware InSQL database is equivalent to DATRAN VI's DATRAN database. An InSQL database provides data storage in a different format.

The InSQL Database Connectivity module never communicates directly with the DATRAN Browser or any of the other DATRAN VI services or modules.

Communication is achieved via RPC calls. DATRAN Browser has a node that gets updated. The DATRAN Browser then sends the requested data to be logged to the TBServ (Tree Base Server) via RPC. TBServ then sends that data to the DATRAN InSQL Database Connectivity Module service via RPC which then logs the data to the specified database on the specified server.

DATRAN VI catches nodes named [_InSQL_Log_Analogue_] or [_InSQL_Log_Digital_] when it polls (about every 0.5 seconds) and this node usually has information such as the tag name. InSqlCM builds an InSQL statement and places it in the action queue. Once the action queue is old enough or if it has exceeded its maximum size, InSqlCM creates a FastLoad file. InSqlCM writes all the InSQL statements in the action queue to the FastLoad file and moves it into the FastLoad directory of the InSQL Server e.g. C:\InSQL\Data\DataImport\FastLoad\. The InSQL Server detects the FastLoad file and imports the data within the FastLoad file into the RunTime database.

18.3. Wonderware IndustrialSQL/InSQL Server

Wonderware's InSQL Server is an enterprise level database server with many features that enables easy collection of data from a variety of sources in a number of different data repositories. Due to Wonderware InSQL Server's design it is always recommended to run the InSQL Server on an independent Windows 2000/2003/XP server. Independent meaning that Wonderware InSQL Server is the only major service application running on the computer. This is a common model for many Wonderware service applications.

18.4. Setting up DATRAN VI to log to an InSQL Database

18.4.1 Installing Wonderware IndustrialSQL Server on the DATRAN VI server

In order to install the IndustrialSQL Server onto a Windows computer the installation CD will be required.

Initialise the installer and follow the prompts (the default options are acceptable).

18.4.2 Adding an InSQL database to DATRANVI

To use InSQL data logging in DATRAN VI the DATRAN InSQL Database Connectivity Module (InSqlCM.exe) must be installed and running.

Under the *Computer | Applications | InSQL Database Connectivity | Property* node check that 'Active = 1'.

Browse to *Computer | Applications | InSQL Database Connectivity | IndustrialSQL Server Group* node and insert a new node called “RunTime”. A properties dialog will be presented so enter in the relevant details.

- ❑ **Server Name** = <Name of the computer InSQL Server is installed on>
- ❑ **Login name** = aaPower or <database user name>
- ❑ **Password** = aaPower or <password>
- ❑ **Database Name** = RunTime
- ❑ **FastLoad Path** = C:\InSQL\Data\DataImport\FastLoad\

18.4.3 DATRAN InSQL Database Connectivity Properties

The main properties (*Computer | Applications | InSQL Database Connectivity | Properties*) that are of interest are:

- ❑ **Active** – a value of 1 means the service is running, 0 means the service has stopped.
- ❑ **Active Connection** – “Connected” means that the DATRAN VI has successfully connected to the database and can log data; “Disconnected” means it is not connected and/or DATRAN VI is experiencing problems when logging data.

18.4.4 Logging Data with DATRAN VI

InSQL data logging is achieved in DATRAN VI via some internal InSqlCM logging nodes. These nodes can be found in the Insert Node dialogue of DATRAN Browser. Look under the Logging tab.

Digital logging is performed through the use of the `_InSQL_Log_Digital_` node. When the point is set to ‘1’ or ‘0’, the `_InSQL_Log_Digital_` node is toggled.

Analogue logging is performed through the use of the `_InSQL_Log_Analogue` node. When the point analogue value changes beyond the set deadband, the `_InSQL_Log_Analogue_` node is notified of this change.

Whether digital or analogue, the InSqlCM service creates a FastLoad statement that gets put into the action queue. Once the statements in the action queue are old enough (defined by the Oldest Action property) or if the action queue has exceeded its maximum size (defined by the Max Action Queue Length), they are then written into a FastLoad file. The FastLoad file is

then moved into the FastLoad directory of the InSQL Server (defined by the FastLoad Path property). The InSQL Server then imports the data inside the FastLoad file into its own RunTime database.

18.4.5 Database Alarms

DATRANVI can be configured to create alarms for certain database failures or warnings about potential problems such as, but certainly not limited to:

- **InSQL Disconnected** – This alarm is toggled by a formula reading the Active Connection node of the database. If the Active Connection is set to Disconnected then the alarm will be generated.